

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
ОЦІНКИ ДІЯЛЬНОСТІ ОПЕРАТОРА АВТОМАТИЗОВАНОЇ СИСТЕМИ
УПРАВЛІННЯ ТЕХНОЛОГІЧНИМ ПРОЦЕСОМ
RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS FOR
ASSESSING THE ACTIVITY OF THE OPERATOR OF PROCESS
AUTOMATION SYSTEM

Виконав: студент(ка) 2 курсу, групи КНТ-119м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Мохаммед Х.С.

(прізвище та ініціали)

Керівник Сердюк С.М.

(прізвище та ініціали)

Рецензент Казурова А.Є.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти магістр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2020 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Мохаммеда Халіля Саїда

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів оцінки діяльності оператора автоматизованої системи управління технологічним процесом. Research and Software Implementation of Methods for Assessing the Activity of the Operator of Process Automation System

керівник проєкту (роботи) Сердюк Сергій Микитович, к.т.н., доцент,
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “03” листопада 2020 року № 301

2. Строк подання студентом проєкту (роботи) 5 грудня 2020 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Розробка імітаційної моделі діяльності оператора автоматизованої системи управління. 4. Експлуатація, тестування та експериментальне дослідження програми 5. Економіко-організаційна частина. 6. Охорона праці та безпека в надзвичайних ситуаціях Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Сердюк С.М., доцент		
Економіко-організаційна частина	Пожуєва Т.О. проф.		
Охорона праці та безпека в надзвичайних ситуаціях	Коробко О. В., ст., викладач		
Нормоконтроль	Калініна М.В., асистент		

7. Дата видачі завдання “04” вересня 2020 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області	2-3 тижні	Розділ 1
3	Розробка та удосконалення методів, моделей й алгоритмів вирішення задачі.	4-5 тижні	Розділ 2
4	Розробка архітектури програми.	6 тиждень	Розділи 2, 3
5	Розробка програми.	7-8 тижні	Розділ 3
6	Тестування та експериментальне дослідження програмного забезпечення.	9 тиждень	Розділ 4
7	Розробка організаційно-економічної частини.	10 тиждень	Розділ 5
8	Розробка заходів з охорони праці та безпеки в надзвичайних ситуаціях.	11 тиждень	Розділ 6
9	Оформлення пояснювальної записки та документів до неї.	12-13 тижні	
10	Нормоконтроль та рецензування.	14-15 тижні	
11	Захист роботи.	16 тиждень	

Студент(ка)

(підпис)

Мохаммед Х.С.
(прізвище та ініціали)

Керівник проєкту (роботи)

Сердюк С.М.

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
134 с., 11 табл., 29 рис., 2 дод., 42 джерел.

ІМІТАЦІЙНА МОДЕЛЬ, СИСТЕМА ДОЗУВАННЯ, SIMC, СИСТЕМА
МАСОВОГО ОБСЛУГОВУВАННЯ, ОПЕРАТОР, ТЕХНОЛОГІЧНИЙ
ПРОЦЕС.

Об'єкт дослідження – процес управління складними технологічними процесами (ТП).

Предмет дослідження – моделі, методи і засоби оцінки діяльності оператора автоматизованої системи управління технологічним процесом (АСУТП).

Мета роботи – підвищення ефективності та якості діяльності оператора АСУТП шляхом розроблення методу кількісної оцінки діяльності оператора та імітаційного моделювання ТП.

Матеріали, методи та технічні засоби: методи імітаційного моделювання; функціонально-структурна теорія ерготехнічних систем; крос-платформовий інструментарій Qt 5.13; система імітаційного моделювання SIMC; операційна система Windows 7 або вище.

Результати. Розроблено метод і програмне забезпечення, що дозволяють кількісно оцінити діяльність оператора АСУТП газобетонного виробництва.

Висновки. Використання розробленого методу дозволяє вирішення завдань кількісної оцінки способів управління ТП з метою їх оптимізації.

Галузь використання – проектування діяльності оператора АСУТП.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 134 pages, 11 tables, 29 figures, 2 appendixes, 42 sources.

SIMULATION MODEL, DOSING SYSTEMS, SIMC, QUEUEING THEORY, OPERATOR, TECHNOLOGICAL PROCESS.

Object of study is process of managing complex technological processes (TP).

Subject of study are models, methods and means of assessing the activities of the operator of Process Automation System (PAS).

The purpose of the work is to increase the efficiency and quality of the activities of the operator of PAS by developing a method for quantitative assessment of the operator's activities and simulation modeling of TP.

Materials, methods, and tools: artificial intelligence methods; simulation methods; functional and structural theory of ergotechnical systems; cross-platform tools Qt 5.13; simulation system SIMC; operating system Windows 7 or higher.

Results. A method and software have been developed to quantify the activity of the aerated concrete production Process Automation System operator.

Conclusions. The implementation of the developed method allows solving the problems of quantitative evaluation of TP management methods in order to optimize them.

The field of use is a design of the activity of the operators of Process Automation System.

ЗМІСТ

	С.
Перелік скорочень та умовних познач 8	8
Вступ 9	9
1 Аналіз предметної області 12	12
1.1 Дослідження системи керування лінії приготування газобетону на ЗАТ «Мотор Січ».....12	12
1.2 Аналіз методів та систем імітаційного моделювання.....16	16
1.3 Висновки до першого розділу 23	23
2 Матеріали і методи 25	25
2.1 Аналіз стану досліджень у області кількісної оцінки діяльності оператора 25	25
2.2 Формальний опис процесу дозування 40	40
2.3 Висновки до другого розділу 46	46
3 Розробка імітаційної моделі діяльності оператора автоматизованої системи управління 48	48
3.1 Моделювання в середовищі SIMC.....48	48
3.2 Розробка алгоритму роботи імітаційної моделі процесу дозування компонентів суміші газобетону 51	51
3.3 Розробка алгоритму роботи імітаційної моделі діяльності оператора 54	54
3.4 Розробка імітаційної моделі процесу виготовлення газобетону.....56	56
3.5 Висновки до третього розділу 60	60
4 Експлуатація, тестування та експериментальне дослідження програми.....61	61
4.1 Вихідні дані для імітаційного експерименту 61	61
4.2 Результати імітаційного експерименту 63	63
4.3 Висновки до четвертого розділу 65	65
5 Економіко-організаційна частина.....67	67
5.1 Планування розробки програмного продукту.....67	67
5.2 Визначення витрат на розробку програми 68	68

5.3 Розрахунок економічної ефективності програмного продукту.....	75
6 Охорона праці та безпека в надзвичайних ситуаціях	78
6.1 Аналіз потенційних небезпек.....	78
6.2 Заходи із забезпечення безпеки	79
6.3 Заходи з виробничої санітарії та гігієни праці.....	81
6.4 Заходи безпеки у надзвичайних ситуаціях.....	89
Висновки	95
Перелік джерел посилання.....	96
Додаток А Текст програми	101
Додаток Б Копії слайдів презентації	125

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

PAS	– Process Automation System;
TP	– technological process;
АСУТП	– автоматизована система управління технологічним процесом;
ВКВ	– вапняно-кремнеземисте, що в'яже;
Гб	– гігабайт;
ІМ	– імітаційна модель;
ЛМІ	– людино-машинний інтерфейс;
Мб	– мегабайт;
ОКК	– об'єкти контролю й керування;
ОП	– операційна панель;
ПУ	– пульт управління;
СІМ	– система імітаційного моделювання;
СМО	– система масового обслуговування;
ТП	– технологічний процес;
ТФС	– типова функціональна структура.

ВСТУП

Досвід експлуатації АСУТП розроблених навіть за допомогою сучасних SCADA-систем показав, що кількість аварій об'єктів контролю і управління істотно не скоротилася. Якщо в 60-х роках помилка людини була первісною причиною тільки 20% аварій (80%, відповідно, за технологічними несправностями і відмовами), то в 90-х роках частка людського фактору виросла до 80% [1]. Причому, в зв'язку з постійним вдосконаленням технологій і підвищенням надійності електронного обладнання та машин, ця частка може ще зрости.

Таким чином, підвищення ефективності, якості і надійності систем диспетчерського управління шляхом оптимізації інформаційного завантаження людини-оператора АСУТП є, на сьогоднішній день, актуальною проблемою.

На сучасному етапі одним з найбільш важливих і корисних знарядь аналізу структури складних процесів і систем є імітаційне моделювання. Воно дає можливість користувачеві експериментувати з системами (існуючими або пропонованими) в тих випадках, коли робити це на реальному об'єкті практично неможливо або недоцільно, дозволяє дізнатися про стан системи або окремих її елементів в певні моменти часу.

Імітаційне моделювання дозволяє вирішувати багато питань аналізу і організації діяльності оператора. До них відносяться: визначення необхідної кількості операторів; вимоги до рівня підготовленості оператора (кваліфікації, швидкості реакцій, об'єму пам'яті і т.п.); допустимій щільності потоків сигналів, що поступають до оператора; рішення деяких задач організації взаємодії операторів. Існує можливість обчислення ймовірності різних станів системи людина-машина, у тому числі і різних небажаних станів [2].

У даній випускній роботі ставиться задача підвищення ефективності та якості діяльності операторів процесу управління дозуванням матеріалів газобетонної суміші на ЗАТ «Мотор Січ».

Впровадження нових науково обґрунтованих технічних рішень у виробництво газобетонних виробів та у процеси керування ТП з урахуванням діяльності оператора, приведе до підвищення ефективності роботи технологічних ліній і якості продукції, що випускається, а також зниженню потреб в електроенергії на одиницю продукції.

Для вирішення поставленої задачі необхідно розробити модель, що формалізує діяльність людини-оператора і фізичні процеси, що відбуваються в ході ТП, з урахуванням характеристик виконавчих механізмів і режимів їх управління, а також її практична реалізація у вигляді імітаційної моделі (ІМ).

Технологічним аспектам приготування газобетонів присвячено досить велику кількість робіт, наприклад [3], [4]. При цьому питання, присвячені побудові та функціонуванню автоматизованих систем управління виробництва газобетону висвітлені в загальних рисах і носять описовий характер [5].

Система «оператор-АСУТП» є людино-машинною системою (ЛМС). Застосування математичних методів для розробки ЛМС, зважаючи на складність операторської діяльності, часто дозволяє оцінювати її лише наближено [6]. В цьому випадку, ефективне використання імітаційного моделювання ЛМС і математичного апарату теорії масового обслуговування [2], [6].

Тому тема випускної роботи є актуальною і затребувана практикою будівельної галузі промисловості України.

Об'єкт дослідження даної випускної кваліфікаційної роботи магістра – процес управління складними ТП.

Предмет дослідження – моделі, методи і засоби оцінки діяльності оператора АСУТП.

Мета роботи – підвищення ефективності та якості діяльності оператора АСУТП шляхом розроблення методу кількісної оцінки діяльності оператора та імітаційного моделювання ТП.

Для досягнення поставленої мети у роботі необхідно вирішити наступні завдання:

- дослідити процеси управління та виготовлення газобетону, та моделі, що їх описують;
- дослідити структуру і функціонування систем імітаційного моделювання (СІМ);
- провести аналіз відомих СІМ;
- розглянути стан досліджень у області кількісної оцінки діяльності оператора;
- розробити комплексну математичну модель, що формалізує діяльність як людини-оператора, так і фізичні процеси, що відбуваються в ході ТП;
- практична реалізація комплексної математичної моделі у вигляді ІМ процесу управління дозуванням матеріалів газобетону на ЗАТ «Мотор Січ»;
- провести імітаційний експеримент та дослідити кількісні характеристики діяльності оператора.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження системи керування лінії приготування газобетону на ЗАТ «Мотор Січ»

Діяльність оператора пов'язана з об'єктом автоматизованого управління у вигляді технологічної лінії виробництва газобетону. Типова автоматизована технологічна лінія приготування газобетону представляє собою складну технологічну систему, яка складається з множини взаємопов'язаних підсистем управління та обладнання (технологічного та електротехнічного). Структурна схема взаємозв'язків технологічних підсистем і програмно-логічних блоків такої лінії показана на рис. 1.1 [5].

Як видно з структурної схеми, кожна підсистема виконує функції управління відповідними технологічними об'єктами та ІМ на основі автоматизованого аналізу поточної інформації, яка надходить від датчиків контролю, що здійснюється при оперативному управлінні відповідною ділянкою технологічної лінії та фазами ТП, а також ТП в цілому.

Аналіз структурної схеми показує, що ТП приготування газобетону укрупнено може бути представлений наступною послідовністю операцій: помел; дозування; змішування; сушка; порізка; термічна обробка і повернення відходів на дозування після змішування і порізки (рис.1.2).

АСУТП дозування матеріалів для виготовлення газобетону на ЗАТ «Мотор Січ» призначена для виконання перших двох операцій (рис.1.2) . Тому, необхідне використання відомих математичних моделей процесів помелу, дозування (багатокомпонентного і однокомпонентного) компонентів газобетонної суміші (рідких і сухих) та їх змішування.

Газобетонна суміш має наступний склад:

- рідкі компоненти (гаряча і холодна вода, піщаний і поворотний шлам, суспензія алюмінієвої пудри);
- сухі компоненти (вапняно-кремнеземистого, що в'яже (ВКВ), що складається з помелених спільно вапна, піску і цементу).

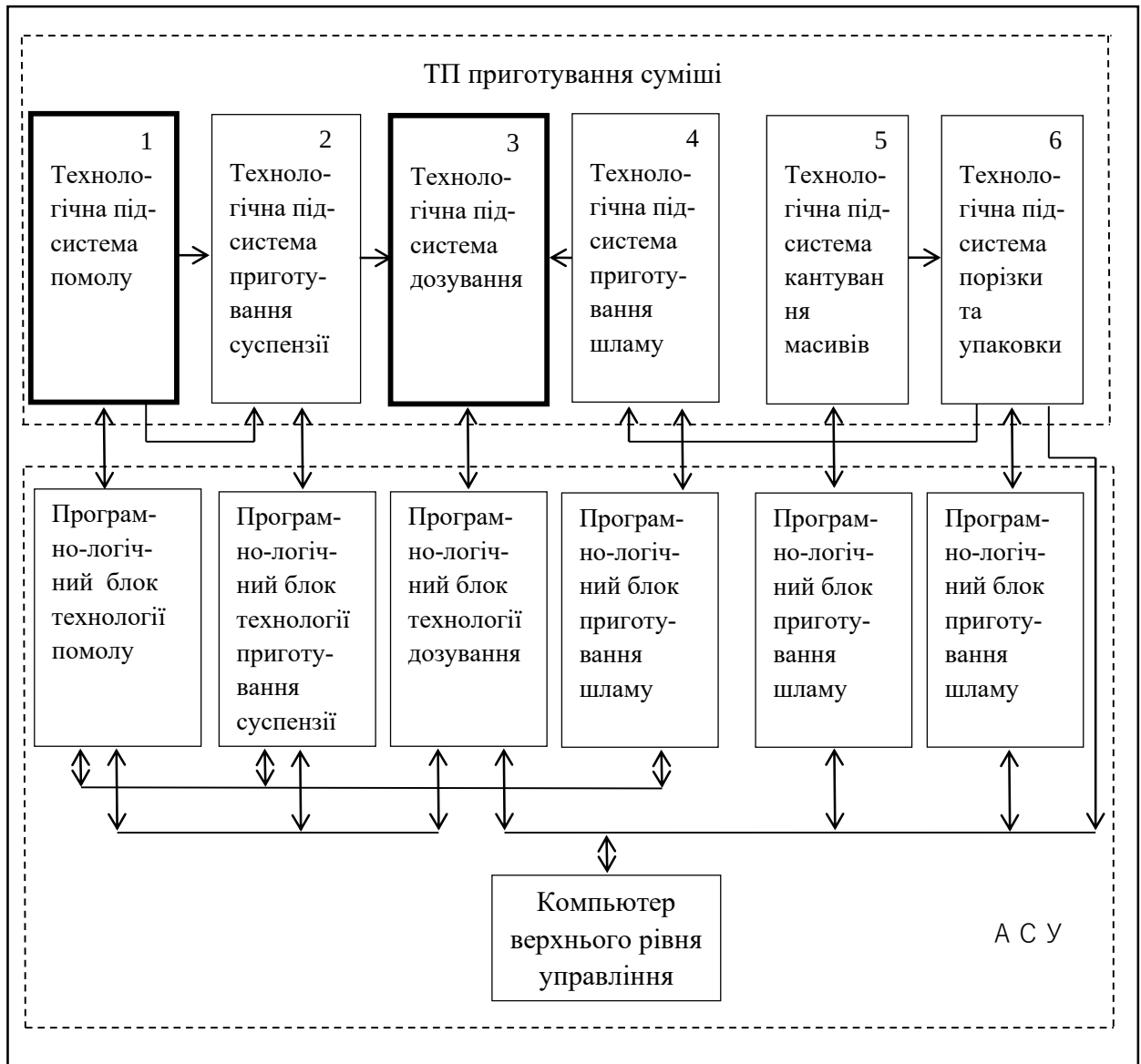


Рисунок 1.1 – Структурна схема технологічних підсистем

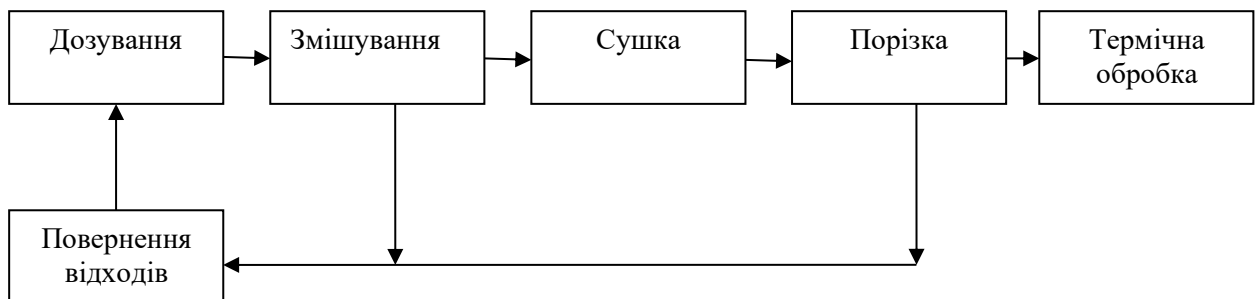


Рисунок 1.2 – Укрупнені операції технологічного процесу приготування газобетону

Дозування відповідних компонентів здійснюється в певних пропорціях і послідовності, які потім змішуються в відповідних технологічних ємностях згідно директивного графіка (циклограммі). Після технологічної фази «спучування» і досягнення певного ступеня щільності (застигання) газобетонна суміш калібрується на відповідні блоки і подається в автоклави для термічної обробки в середовищі з директивною вологістю.

1.1.1 Підсистема помелу вихідних компонентів

Підсистема помелу вихідних компонентів служить для приготування компонентів суміші газобетону (помелу піщаного шламу і ВКВ), які використовуються безпосередньо на наступному етапі ТП в підсистемі дозування.

Управління технологічною підсистемою помелу з боку оператора має забезпечити [5]:

- отримання піщаного шламу заданої щільності ($1600 - 1800 \text{ кг/м}^3$);
- коригування подачі певної кількості води для підтримки відповідної вологості піску;
- поточне коригування та стабілізація кількісного співвідношення компонентів по щільності одержуваного піщаного шламу згідно з показаннями щільноміру (поз. 12 на рисунку 1.2);
- забезпечення і підтримання технологічного запасу піщаного шламу в шламбасейні;
- задану активність ВКВ шляхом регулювання потоків вапна і піску, що подаються у відповідну технологічну ємність;
- забезпечення і підтримання заданого запасу ВКВ в технологічних ємностях у міру їх витрати;
- безперервну роботу та поточну продуктивність лінії у відповідність до директивної програми.

Таким чином, оператор має:

- стежити за роботою обладнання і регулювати технологічний режим за показаннями контрольно-вимірювальних приладів (або елементів інтерфейсу) з пульта управління;
- контролювати дотримання параметрів технологічного процесу (установок), виявляти, аналізувати допущені відхилення від заданих режимів і приймати рішення для їх своєчасної ліквідації.

1.1.2 Підсистема дозування і приготування газобетону

Підсистема дозування компонентів газобетонної суміші необхідна для дозування компонентів суміші і подальшого їх змішування з метою отримання гомогенної рідкої суміші газобетону.

Задачі управління технологічною підсистемою дозування наступні [5]:

- забезпечення одночасного дозування наступних компонентів: сухих, шламів, холодної і гарячої води, суспензії алюмінієвої пудри в умовах коливання їх потоків подачі;
- багатокomпонентне дозування сухих (ВКВ і цементу) з похибкою не більше $\pm 0,5\%$;
- багатокomпонентне дозування шламів і холодної води з похибкою не більше $\pm 1\%$;
- дозування гарячої води з похибкою не більше $\pm 1\%$;
- дозування суспензії алюмінієвої пудри з похибкою не більше $\pm 0,5\%$;
- пошук і забезпечення оптимального співвідношення точність дозування – швидкість дозування шляхом адаптивного зміни попередження закінчення дозування і переходу режим точного дозування;
- забезпечення безперервності ТП, включаючи виготовлення масивів газобетону і технологічних відходів із заданою періодичністю.

З наведеного аналізу технологічних режимів і особливостей роботи окремих підсистем можна сформулювати загальні завдання управління АСУ ТП виробництва газобетону, яка здійснює:

- забезпечення ефективного програмного управління із заданою періодичністю виготовлення сирих еталонних газобетонних масивів і повного завантаження автоклавів;

- синтез управління підсистемами технологічної лінії із заданою періодичністю окремих фаз ТП і циклів між ними, а також зниження непродуктивних простоїв виконавчих механізмів та обладнання;

- забезпечення необхідного резерву компонентів газобетонної суміші;

- коригування фізико-технічних параметрів газобетонної суміші при змінах (тобто параметричних збурень) характеристик окремих компонентів, зовнішніх умов навколишнього середовища (вологість і температура) і зворотного шламу;

- коригування та підтримання оптимальних режимів роботи виконавчих механізмів (регуляторів, заслінок, приводів) технологічної лінії.

Процес дозування, його формалізація і опис діяльності оператора будуть розглянуті в розділі 2.

1.2 Аналіз методів та систем імітаційного моделювання

1.2.1 Аналіз структури і функціонування систем моделювання

Формально, ТП виробництва газобетону може бути представлений у вигляді системи масового обслуговування (СМО) дискретного типу. В цьому випадку, предмети праці (вихідні компоненти) можуть бути описані у вигляді вхідних потоків заявок, що надходять на обслуговування в СМО.

При дискретній імітації стан системи може змінюватися тільки в моменти здійснення подій. Так як стан системи не змінюється між цими моментами, повний динамічний портрет станів системи може бути отриманий шляхом просування імітаційного часу від однієї події до іншої.

Для визначення вимог до системи імітаційного моделювання в цілому, доцільно провести аналіз структури, призначення і принципу функціонування узагальненої системи імітаційного моделювання (СІМ) СМО дискретного типу.

Нехай об'єктом імітації є послідовна обробка i -ї деталі на декількох станках. Кожен j -й верстат забезпечує виконання свого набору операцій обробки деталі i , реалізуючи, таким чином, функціональну дію ΦD_{ij} . Обробку множини деталей можна розглядати як складну систему, в якій кожна j -а деталь є компонентою K_i .

При реалізації ІМ зазвичай використовують три форми подання часу: t_R - реальний час системи; t_0 - модельний час, за яким організовується синхронізація часу в системі та t_e - машинний час.

В ІМ відбувається представлення кожної ΦD_{ij} парою дій [7]: спрощеним функціональним дією $\Phi D'_{ij}$ і модифікацією тимчасової координати i -ї компоненти моделі t_i . Ця пара дій називається активністю AK_{ij} . Будь-яка AK_{ij} складається з опису алгоритму AL_{ij} і оператора модифікації тимчасової координати компоненти моделі Mt_{ij} . Для синхронізації виконання компонент ІМ на ЕОМ виконується глобальна змінна – модельний час t_0 . Управління активностями здійснює планувальник подій (управляюча програма моделювання – УПМ). Крім модифікації модельного часу t_0 і виконання операторів Mt_{ij} в його функції входить:

- запуск на виконання алгоритмів AL_{ij} ;
- організація взаємодії AL_{ij} один з одним в ході імітації;
- перевірка умов закінчення імітації.

Для задання вхідних умов, початкових значень параметрів і запуску ІМ до складу моделі включається підпрограма початку імітації. Так як ІМ створюється для вивчення на основі статистики, що характеризує поведінку компонент K_i реальної системи в різних режимах її функціонування, то для збору статистики моделювання в ІМ вводиться відповідна підсистема. По досягненню умов закінчення імітації планувальник подій передає управління на підпрограму закінчення імітації, яка часто і забезпечує розрахунок характеристик поведінки моделі системи і здійснює виведення результатів моделювання. Всі ці підпрограми взаємодіють тільки з планувальником подій.

На рис. 1.3 представлена схема переведення реального об'єкта моделювання в його ІМ.

З даного рисунка видно, що будь-яка ΦD_{ij} в ІМ описується відповідною активністю AK_{ij} . Кожна AK_{ij} є попарним поєднанням опису алгоритму AL_{ij} і оператора модифікації тимчасової координати компоненти моделі Mt_{ij} . УПМ передає управління на початок програми, що реалізує алгоритм AL_{ij} . Повернення до УПМ здійснюється при виконанні операторів Mt_{ij} .

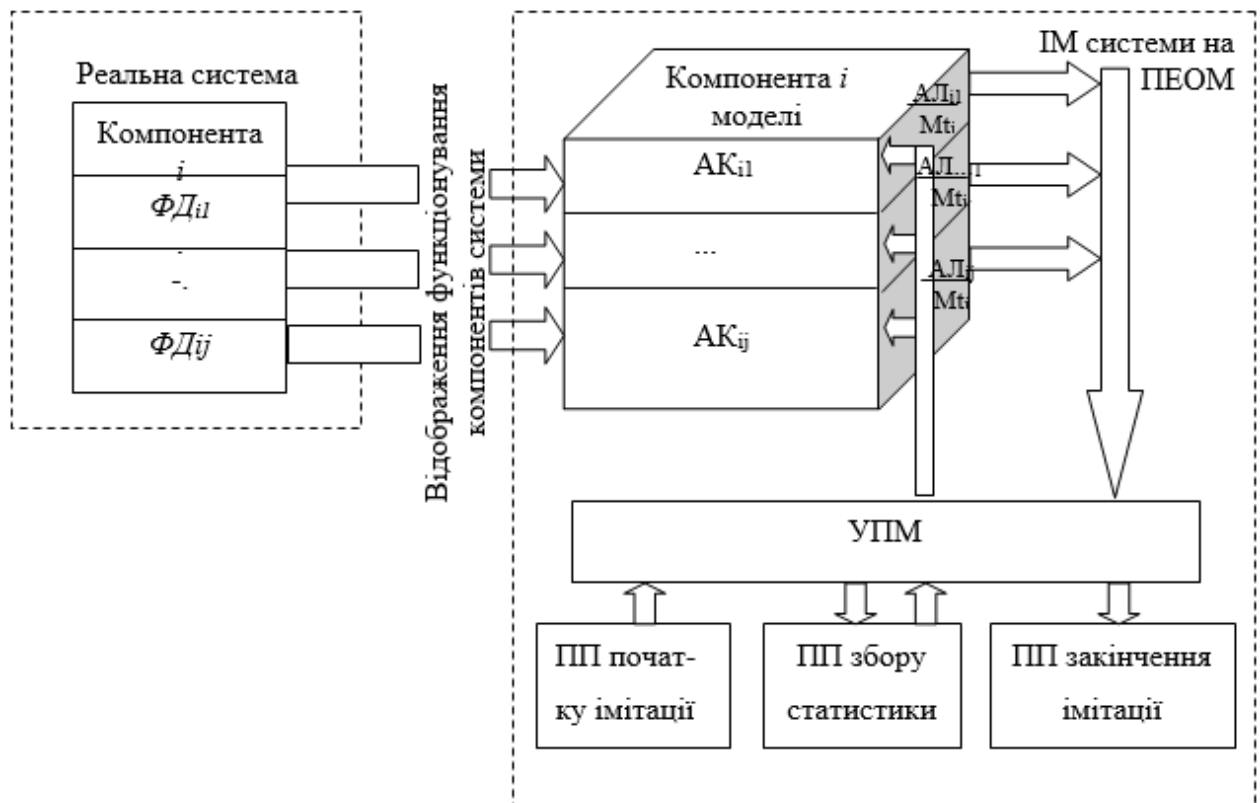


Рисунок 1.3 – Схема переходу від реального об'єкта до його ІМ

Залежно від складу алгоритмів AL_{ij} , наявності зв'язків між компонентами, цілей і задач моделювання вибирають той чи інший оптимальний спосіб представлення K_i і реалізації активностей в ІМ. Найбільшого поширення набули наступні п'ять способів опису ІМ [7]-[10]: безпосередньо активностями, апаратами подій, транзактами, агрегатами та процесами.

Одну і ту ж систему що досліджується можна представити будь-яким із зазначених способів формалізації, однак ІМ, побудовані на їх основі, будуть

відрізнятися розмірами і кількістю ресурсів, що витрачаються на їх створення, випробування і використання.

1.2.2 Аналіз парадігм та систем імітаційного моделювання

Світова практика прийняття складних управлінських рішень в різних системах перейшла на новий рівень, коли ті чи інші варіанти можливих рішень повинні бути попередньо випробувані на моделях. У зв'язку з цим прийняття рішення або інновації вимагають попередніх оцінок можливих результатів за допомогою системного аналізу та імітаційного моделювання.

Судячи з обсягів продаж SIM AnyLogic, що стає все більш популярною, найактивніше імітаційним моделюванням цікавляться в металургійній та нафтогазовій галузях, виробництві будматеріалів та харчових продуктів [11].

У імітаційному моделюванні виділяють наступні основні парадігми: системна динаміка, дискретне моделювання та агентне моделювання. Якщо дискретно-подієве моделювання та системна динаміка, які були розроблені ще в 60-х роках ХХ століття, прийнято відносити до традиційних підходів, то агентне моделювання вважається новою парадігмою. Практично всі програмні інструменти імітаційного моделювання розроблені для підтримки одного певного підходу [12].

Крім того, у роботах різних авторів серед підходів імітаційного моделювання відзначаються також динамічне моделювання, об'єднаний підхід, мережеві парадигми, мережі кусочно-лінійних агрегатів та ін. [13].

Системна динаміка як методологія була запропонована в 1961 році Дж. В. Форрестером для дослідження динамічних процесів, вивчення складних систем зі зворотним зв'язком. Системно-динамічна модель описує поведінку системи та її структуру як множину взаємодіючих зворотних зв'язків і затримок. Математично така модель виглядає як система диференціальних рівнянь. Результатом моделювання в системній динаміці є виявлення глобальних залежностей та причинно-наслідкових зв'язків у системі, що досліджується [14].

Парадигма «Системна динаміка» реалізована в програмних продуктах: VenSim, PowerSim, iThink, ModelMaker та ін.

Основи дискретно-подієвого підходу були закладені Дж. Гордоном зі створенням CIM GPSS на початку 60-х років XX століття. Це найбільш поширений підхід імітаційного моделювання, який застосовується в багатьох сферах та системах моделювання. В його основі лежить концепція систем масового обслуговування, а у якості способу опису ІМ – транзактний спосіб організації квазіпаралелізму.

Переміщаючись по моделі, транзакти стають в черги до одноканальних та багатоканальних обслуговуючих пристроїв, захоплюють і звільняють їх, розщеплюються, знищуються і т.п. Дискретно-подієве моделювання застосовується, якщо можливо припустити, що змінні в системі змінюються миттєво в певні проміжки часу [13]. Цей вид моделювання найбільш підходить для моделювання виробничих процесів.

Дискретно-подієве моделювання реалізується за допомогою програмних продуктів: Arena, GPSS, SIMC, Extend, SimProcess, AutoMod, PROMODEL, Enterprise Dynamics, FlexSim, eMPlant і ін.

Агентне моделювання бере свій початок в 90-х роках XX століття. В основі цього підходу лежить поняття "агент" – сутність, що володіє активністю, автономною поведінкою, може приймати рішення відповідно до деякого набору правил, взаємодіяти з оточенням, а також самостійно змінюватися [15]. Даний підхід застосовується для імітації децентралізованих, інтелектуальних і розподілених систем з метою вивчення впливу на систему взаємодії елементів.

На відміну від системної динаміки або дискретно-подієвих моделей, тут немає конкретного місця, де централізовано визначається поведінка (динаміка) системи в цілому. Замість цього аналітик визначає поведінку на індивідуальному рівні, а глобальна поведінка виникає як результат діяльності багатьох агентів, кожен з яких за своїми власними правилами, живе в загальному середовищі і взаємодіє з цим середовищем та з іншими агентами.

Найбільш потужним з інструментів, що підтримують агентне моделювання, є інструментарій AnyLogic одноіменної компанії (XJ Technologies у минулому), що довів у нещодавній час свою міць і зручність за рахунок застосування об'єктно-орієнтованого підходу, візуального проектування, дружнього графічного інтерфейсу, платформонезалежної мови Java, агентних технологій, технології гібридних систем. Крім того, він є єдиним інструментарієм, який реалізує об'єднаний підхід за рахунок надання можливості створення гібридних моделей на основі моделей системної динаміки, дискретно-подієвих моделей і агентного підходу [16].

Окрім вищезгаданого AnyLogic, агентне моделювання реалізується за допомогою програмних продуктів: Swarm, RePast та ін.

Базуючись на рекомендаціях [13] та [16], виберемо наступні критерії оцінки вищезгаданих систем і проведемо порівняльний аналіз найбільш популярних систем моделювання (табл. 1.1):

- розробник;
- області використання;
- мова програмування, або спеціалізована мова;
- можливість одночасного використання всіх парадигм моделювання в одній моделі;
- 3D-анімація;
- доступ до "ядра" керуючої програми моделювання;
- кількість стандартних теоретичних розподілів (для моделювання випадкових процесів);
- наявність об'єктно-орієнтованого підходу;
- документування результатів моделювання;
- наявність постійної підтримки середовища розробниками;
- умови ліцензії.

Таблиця 1.1 – Порівняльний аналіз середовищ імітаційного моделювання

Характеристика	AnyLogic	Arena	Simulink	GPSS World	SIMC
Розробник	The AnyLogic Company	Rockwell Automation Inc	MathWorks Inc	Minute man Software	НУ «ЗП»
Області використання	Бізнес процеси. Соціальна динаміка. Управління проектами і персоналом. Виробництво. Логістика. Стратегічне прогнозування.	Бізнес процеси. Виробництво. Логістика. Склад. Центр обробки викликів.	Медичні дослідження. Системи управління, фінансові розрахунки. Обробка сигналів і зображень.	Виробництво. Системи масового обслуговування.	Виробництво. Системи масового обслуговування.
Мова програмування, спеціалізована мова	Java	SIMAN	MATLAB	GPSS	SIMC
Одночасне використання всіх парадигм моделювання	+	–	–	–	–
3D-анимація	+	+	+	–	–
Доступ до "ядра"	–	–	–	–	+
Стандартні теоретичні розподіли	29	13	>30	>20	10
Об'єктно-орієнтований підхід	+	–	–	–	+
Документування результатів моделювання	+	+	+	+	+
Підтримка середовища розробниками	+	+	+	-	+
Умови ліцензії	Безкоштовна тільки версія Personal Learning Edition	759303 руб.	MATLAB and Simulink Student Suite \$55	+	+

Слід зазначити, що в будь-якій СІМ можна знайти якісь недоліки з погляду її застосування у кожному конкретному випадку. Крім того, сфера інформаційних технологій бурхливо розвивається, з'являються нові підходи і методи вирішення актуальних задач, змінюються механізми взаємодії користувачів з інформаційними системами. Все це призводить до необхідності постійного вдосконалення СІМ і рішень, що в них використовуються.

Таким чином, із зазначеного в табл. 1.1 порівняння, а також з робіт [16] та [17] випливає висновок про беззаперечні переваги AnyLogic. Але з урахуванням причин, перелічених нижче у якості СІМ було обрано систему SIMC [18]:

- систему SIMC розроблено на кафедрі програмних засобів національного університету «Запорізька політехніка», тому автор має практичний досвід її використання;
- можливості системи SIMC достатні для моделювання ТП виготовлення газобетону на основі використання дискретно-подієвої парадигми;
- доступ до "ядра" керуючої програми моделювання забезпечується тим, що модель представляє собою програму на мові високого рівня C++, що відлагоджується та виконується з використанням крос-платформового інструментарію Qt 5.13.
- необхідність у 3D-анімації не мала вирішальної ролі;
- система SIMC є безкоштовною.

Обрана система SIMC базується на транзактному способі опису ІМ.

1.3 Висновки до першого розділу

Дослідження системи керування лінії приготування газобетону на ЗАТ «Мотор Січ» показало, що вона призначена для виконання операцій помелу та дозування. Тому, при імітаційному моделюванні, необхідно використання математичних моделей процесів як помелу так і дозування

(багатокомпонентного і однокомпонентного) компонентів газобетонної суміші (рідких і сухих) та їх змішування.

Проведений аналіз структури і функціонування систем моделювання показав, що найбільш поширеними є наступні п'ять способів опису ІМ: безпосередньо активностями, апаратами подій, транзактами, агрегатами та процесами.

В результаті аналізу парадігм та систем імітаційного моделювання у якості СІМ для моделювання діяльності оператора газобетонного виробництва було обрано SIMC, що базується на транзактному способі організації квазіпаралелізму у моделі.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Аналіз стану досліджень у області кількісної оцінки діяльності оператора

2.1.1 Види операторської діяльності

Співвідношення між розумовими і практичними діями в діяльності оператора в значній мірі залежить від ступеня автоматизації процесу управління. Чим більше ступінь автоматизації, тим більше питома вага розумових дій і менше – практичних. Залежно від ступеня ієрархії, на якій знаходиться людина-оператор, і виду автоматизованої системи, розрізняють наступні класи операторської діяльності [2], [8], [9] .

Оператор-технолог. Оператор безпосередньо включений в технологічний процес і працює в основному в режимі негайного обслуговування. В його діяльності переважають виконавчі дії. До цього класу відносяться оператори технологічних процесів, автоматичних ліній, оператори по прийому і переробці інформації.

Оператор-маніпулятор. Для його діяльності велике значення має сенсомоторна координація (наприклад, безперервне стеження за об'єктом, що рухається) і моторні (рухові) навички, а також, хоч і у меншій мірі, механізми образного і понятійного мислення.

Оператор-спостерігач (контролер). Для його діяльності важливе значення мають інформаційні і концептуальні моделі, а також процеси прийняття рішень. Такий тип діяльності є масовим для систем, що працюють в реальному масштабі часу (оператори радіолокаційних станцій, диспетчери на різних видах транспорту і т. ін.).

Оператор-дослідник. Для діяльності такого оператора важливе значення мають процеси оцінки інформації. Базуючись на своєму досвіді, закладеним в концептуальну модель (урахування усіх чинників, що можуть вплинути на результат дослідження) об'єкту, що досліджується, використовуючи апарат понятійного мислення, дослідник має набір алгоритмів, вибір одного з яких

залежить від ідентифікації отриманої інформації. До таких операторів відносяться дослідники будь-якого профілю – користувачі обчислювальних систем, дешифрувальники об'єктів (зображень) і т. ін.

Оператор-керівник. На відміну від інших видів операторів, оператор-керівник здійснює безпосереднє або опосередковане (через технічні засоби і канали зв'язку) управління підлеглими йому людьми. Основний режим діяльності оператора-керівника – оперативне та стратегічне мислення.

Таким чином, за розглянутою класифікацією, оператор АСУТП дозувально-змішувального відділення ЗАТ «Мотор Січ» є оператором-технологом.

2.1.2 Формалізація діяльності оператора

Для оцінки кількості інформації, яка міститься у різноманітних стимулах, що надходять до оператора АСУТП від людино-машинного інтерфейсу (ЛМІ), використовують математичний апарат з теорії інформації. Кількість інформації традиційно визначається величиною зменшення ентропії (тобто, у даному випадку, невизначеності технологічної ситуації) після обробки оператором поточних даних. Ентропія і кількість інформації вимірюються в двійкових одиницях інформації, або бітах. Біт – це ентропія системи, яка має два рівноімовірні стани. Величина ентропії фізичної системи X , що приймає стани $x_i (i=1, 2, \dots, n)$ розраховується за формулою (2.1) [6]:

$$H = - \sum_{i=1}^n P_i \log_2 P_i, \quad (2.1)$$

де P_i – імовірність надходження i -го сигналу (знаходження системи або об'єкту контролю й керування (ОКК) в i -му стані);

n - загальне число різних сигналів (станів системи).

Якщо поява різних станів $P_i = P = 1/n$ рівноімовірна, то у даному випадку ентропія системи буде максимальна і, виходячи з формули (1.1), розраховується за формулою (2.2) [6]:

$$H(X) = -\log_2 P = \log_2 2n. \quad (2.2)$$

Чим більше об'єм отриманих даних і чим вони змістовніші, тим більша інформація є у оператора про систему. Тому природно кількість інформації вимірювати зменшенням ентропії цієї системи, для уточнення станів якої призначені ці дані. Якщо в результаті отримання даних стан системи став повністю визначеним, то кількість отриманої інформації чисельно дорівнює апіорній ентропії системи.

Кількість інформації, переробленої людиною-оператором, може істотно відрізнятися від кількості інформації (ентропії джерела повідомлень), що надійшла, – частина інформації може бути втрачена за рахунок впливу перешкод, а деяка кількість інформації може збільшитися за рахунок використання додаткової інформації.

Загальна кількість інформації, котра переробляється оператором, розраховується за формулою (2.3) [2]:

$$I = H(X) + H_{\text{дод}} - H_{\text{пер}}, \quad (2.3)$$

де $H(X)$ – ентропія джерела повідомлень (кількість інформації, що отримується оператором від ЛМІ);

$H_{\text{дод}}$ – додаткова кількість інформації, яка використовується оператором при вирішенні задачі (наприклад, дані зі своєї пам'яті);

$H_{\text{пер}}$ – ентропія джерела перешкод, або кількість втраченої при передачі інформації.

Можливі також втрати інформації в процесі її передачі по технічних каналах АСУТП або ж самим оператором (наприклад, внаслідок своїх помилок або обмеженості можливостей аналізаторів людини) [10].

Кількість інформації, що переробляється оператором необхідна для вирішення наступних ергономічних задач [2]:

- визначення ступеня складності розв'язуваної задачі;
- визначення ступеня узгодженості швидкості надходження інформації з пропускнуою спроможністю оператора (його психофізіологічними можливостями);
- визначення часу обробки інформації.

Час на рішення задач оператором є пропорційним кількості інформації, що їм переробляється. Оператор переробляє різні види інформації з різною швидкістю, розраховується за формулою (2.4) [19]:

$$\tau = \alpha + \sum_{i=1}^k \frac{H_i}{v_i}, \quad (2.4)$$

де α – латентний період реакції оператора на сигнал (близько 0,2 с);

H_i – кількість інформації i -го виду;

v_i – швидкість переробки інформації i -го виду.

Практично, оператор АСУТП газобетонного виробництва приймає інформацію, закодовану за допомогою алфавітів двох модальностей - зорової та слухової. Однак, застосування тільки теорії інформації, при вирішенні перерахованих вище ергономічних завдань, не дозволяє врахувати наступні особливості [19]:

- у процесі переробки закодованої інформації, що надходить, оператор використовує свій внутрішній алфавіт, а його суб'єктивні ймовірності сигналів можуть відрізнятися від об'єктивних;
- об'єктом дослідження теорії інформації є стаціонарні процеси, в той час як характеристики оператора змінюються в часі (втома, емоції і т.п.);

– теорія інформації не враховує семантику інформації, яка може значно вплинути на результати діяльності оператора.

Зазначені недоліки вирішуються в процесі побудови комбінованої моделі, яка є аналітико-імітаційною моделлю. Тому в ній, в порівнянні з аналітичними моделями [29], простіше врахувати:

- зміну станів технологічної системи при випадкових впливах;
- нелінійні характеристики елементів системи;
- дискретний і безперервний характер елементів та ін.

Для вирішення завдань управління і контролю оператор здійснює інформаційний пошук необхідних елементів інформаційної моделі (мнемосхеми, екран людино-машинного інтерфейсу (ЛМІ)). Наприклад, на початковому етапі робочої зміни оператор здійснює пошук рецепту із списку, або задає компоненти власноруч (рис. 2.1).

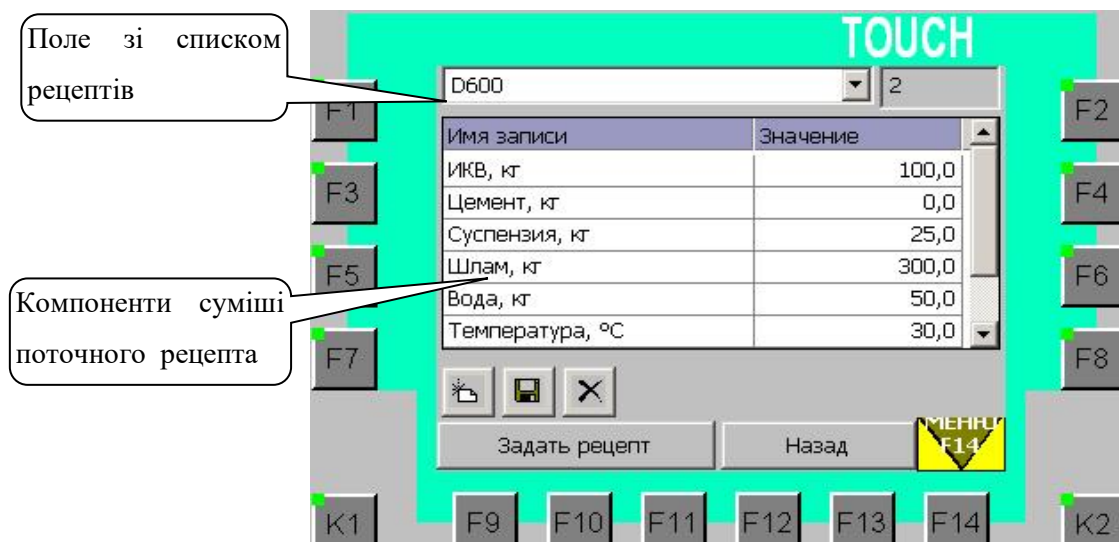


Рисунок 2.1 – Екран «Завдання рецепта»

Загальний час пошуку визначається за формулою (2.5) [10]:

$$t = \sum_{i=1}^n (t_{Fi} + t_{Si}), \quad (2.5)$$

де t_{Fi} – час зчитування i -го слова (рядка);

t_{Si} – час i -го переміщення погляду (саккади ока).

На деяких етапах свого алгоритму функціонування, оператор виконує операції зчитування текстової інформації. У цьому випадку, замість часу i -ї фіксації погляду t_{Fi} в формулу (2.5), необхідно підставити значення часу зчитування i -го слова (рядка) t_{Ri} . Для його розрахунку скористаємося результатами моделювання лексичної ідентифікації з використанням стратегії зміни уваги [21]. Середній час лексичної ідентифікації слова (зчитування), с центральною фіксацією погляду і $\log_{10}$ частоти зустрічаємості слова, розраховується за формулою (2.6) [10]:

$$t_{Ri} = base5 + 15(length - 5) + 40(1 - freq), \quad (2.6)$$

де $base5$ – середній час (близько 150 мс), необхідний для ідентифікації п'ятибуквеного слова з $\log_{10}$ частоти 1,0;

$length$ – довжина слова;

$freq$ – значення $\log_{10}$ частоти зустрічаємості слова.

Розрахунок часу читання оператором t_{Ri} за формулою (2.6) в моделі, що розробляється був зроблений виходячи з наступних положень [10]:

- оператор знаходиться на відстані 60-70 см від операторної панелі і меню екранної форми цілком знаходиться в полі його зору;
- довжина кожного рядка (меню або інформаційного повідомлення) не більше 20 символів, що не перевищує області сприйняття тексту (функціонального поля зору оператора) під час фіксації погляду. Функціональне поле зору обмежене 3-4 буквами зліва від точки фіксації погляду та 14-15 буквами праворуч від точки фіксації [22] або до 20 згідно [21];
- погляд фіксувався на центрі слова;
- можливість рефіксації погляду не враховувалася.

Стандартне відхилення S для часу читання t_{Ri} визначається за формулою (2.7) [21]:

$$S = 0,1 \cdot t_{Ri}. \quad (2.7)$$

Психофізіологічні характеристики оператора накладають обмеження на його можливості з прийому, переробки інформації та прийому рішень по управлінню ТП. Для урахування цього, введемо в модель номенклатуру показників [19], [23], що характеризують гранично допустимі норми діяльності оператора АСУТП.

Інформаційне навантаження оператора характеризується коефіцієнтом завантаженості визначається за формулою (2.8):

$$z = 1 - \frac{t_{\text{роб}}}{T}, \quad (2.8)$$

де $t_{\text{роб}}$ – загальний час, впродовж якого оператор не зайнятий обробкою інформації, що надходить в ЛМІ;

T – загальний час знаходження оператора на робочому місці.

Його максимальне граничне значення Z_{max} не повинно бути більше ніж 0,75 [19], [23].

Обмеження на безперервну (без пауз) роботу оператора, визначається показником «максимальний період зайнятості» $T_{\text{зан}}$, який не повинен перевищувати 15 хвилин [19].

Інформаційне навантаження оператора характеризується, також, таким показником, як коефіцієнт черги ρ , що показує кількість сигналів, оброблених оператором в умовах наявності черги на обслуговування, розраховується формулою (2.9):

$$\rho = \frac{N_{\text{ч.}}}{n}, \quad (2.9)$$

де $N_{\text{ч}}$ – кількість сигналів, оброблених оператором в умовах черги на обслуговування;

n – загальна кількість сигналів, що надійшли до оператора.

Експериментальним шляхом визначено, що величина ρ не повинна перевищувати 0,4 [19], [23]. Максимально допустима довжина черги обмежується обсягом оперативної пам'яті людини і не повинна перевищувати 5-9 сигналів. Однак рекомендується [19], [23], щоб середнє число сигналів в черзі $K_{\text{доп}}$, що одночасно утримуються в пам'яті оператора, не перевищувало трьох.

У моделі, що розробляється, введені алгоритми функціонування оператора (АФО) щодо усунення таких видів неполадок в ТП або помилок оператора:

- відсутність матеріалів в бункерах суспензії, сухих матеріалів (ВКВ і цементу);
- перегрів електроприводу шнека;
- попадання в шнек стороннього тіла;
- попадання на дозатори сторонніх предметів.

Для формального опису процесу функціонування оператора було використано узагальнений структурний метод А.І. Губінського [9]. Наведемо АФО для випадку відсутності матеріалу (ВКВ) в одному з бункерів сухих, із зазначенням типових функціональних одиниць (ТФО), що моделюють реальні операції, які виконуються оператором лінії приготування газобетонної суміші ЗАТ «Мотор Січ» за допомогою пульта управління (рис. 2.2) і набору екранних форм, що відображаються на операторній панелі ОР 177В. У цьому випадку на ньому повинна блімати кнопка «Аварія». Стартовий екран дозування представлений на рис.2.3 [10].

Для усунення даної несправності оператор сумісно з обладнанням повинен виконати наступні операції.

Крок 1. Початок алгоритму (композиціонер «Стартер»).

Крок 2. Перевірка миготіння кнопки «Аварія» (рис. 2.2) (контрольна операція K1). У разі миготіння, перехід на крок 3, інакше крок 33 «Кінець».

Крок 3. Зчитування повідомлення про помилку №13 «Помилка набору ваги дозатора сухих!» у полі аварійних повідомлень-помілок (рис. 2.3) (робоча операція P1).

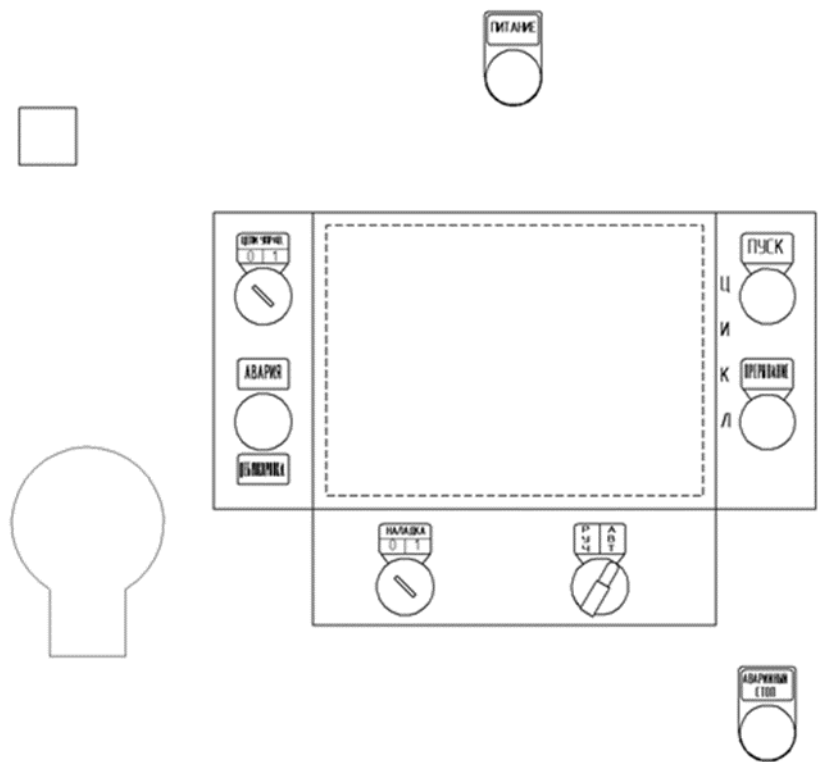


Рисунок 2.2 – Пульт управління оператора

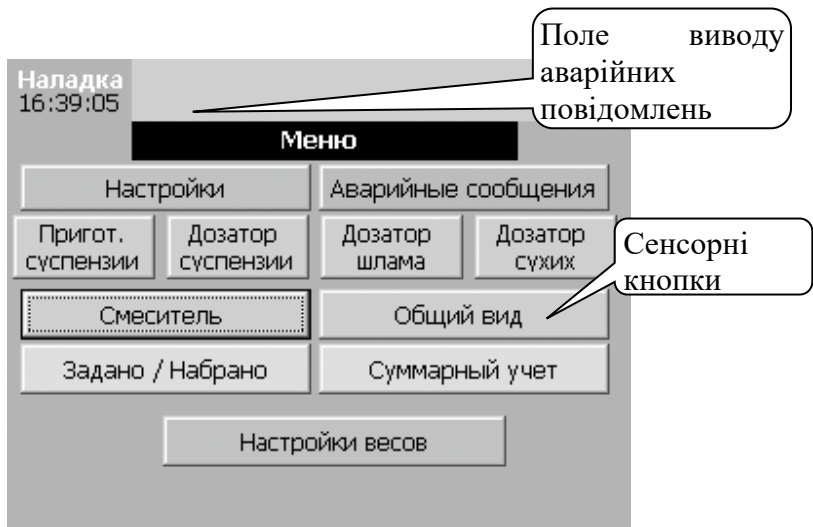


Рисунок 2.3 – Стартовий екран дозування

Крок 4. Натискання кнопки F14 на операторній панелі, щоб повернутися до головного меню (робоча операція P2).

Крок 5. Затримка на час відкриття форми «Стартовий екран дозування» (рис. 2.3) (операція затримка 31).

Крок 6. Вибір дозатора сухих за допомогою натискання відповідної сенсорної кнопки (рис. 2.3) (робоча операція Р3).

Крок 7. Затримка на час відкриття форми «Дозатор сухих» (операція затримка 32).

Крок 8. Зчитування з фіксацією в пам'яті/папері показань поточної ваги компонентів (робоча операція Р4).

Крок 9. Натискання кнопки «ДЕБЛОКИРОВАНИЕ» (рис. 2.2) (робоча операція Р5).

Крок 10. Натискання кнопки «ПУСК» (рис. 2.2) для повтору набору компонентів (робоча операція Р6).

Крок 11. Перевірка роботи (візуально по мнемосхемі форми «Дозатор сухих») в автоматичному режимі заслінок на виході дозатора сухих, шнекового живильника або появи інших помилок в полі аварійних повідомлень (контрольна операція К2). Якщо заслінка відкрилася, шнек спрацював і немає інших помилок, то перехід на крок 31.

Крок 12. Перевірка тиску стисненого повітря (контрольна операція К3). Якщо воно в нормі, то перехід на крок 13. Інакше зателефонувати в компресорну, дочекатися поки тиск не збільшиться (операція затримка 33), перехід на крок 9.

Крок 13. Переключити тумблер установки режиму управління на пульті в положення «РУЧ» (рис. 2.2) (робоча операція Р7).

Крок 14. Відкрити заслінку на вивантаження за допомогою натискання кнопки F6 на операторній панелі (робоча операція Р8).

Крок 15. Затримка на час відкриття заслінки (операція затримки 34).

Крок 16. Якщо заслінка відкрилася, то перехід на крок 18, інакше натиснути кнопку F8 на операторній панелі (робоча операція Р9) для закриття заслінки.

Крок 17. Виклик механіків для огляду/ремонту механізму (операція затримки 35). Після усунення поломки перейти на крок 14.

Крок 18. Включення приводу шнека, натискаючи та утримуючи кнопку F2 на операторній панелі (робоча операція P10).

Крок 19. Перевірка роботи шнека (візуально по мнемосхемі форми «Дозатор сухих») (контрольна операція K4). Якщо шнек працює перейти на п.20, інакше натиснути кнопки F4 (робоча операція P11) і F8 (робоча операція P12) на операторній панелі, для зупинки шнека і закриття заслінок відповідно, перейти до крок 17.

Крок 20. Зчитування показань поточної ваги компонентів (операція затримки 36). Якщо вага не зростає, то крок 21, інакше крок 27.

Крок 21. Включення пневмообрушувачів натисканням сенсорної кнопки «Обрушение» на екранній формі «Дозатор сухих» (робоча операція P13).

Крок 22. Зчитування показань поточної ваги компонентів (операція затримки 37). Якщо вага не зростає, то крок 23, інакше крок 27.

Крок 23. Натискання кнопок F4 (робоча операція P14) і F8 (робоча операція P15) на операторній панелі для зупинки шнека і закриття заслінок.

Крок 24. Дзвінок обслуговуючому персоналу для візуальної перевірки наявності матеріалу в бункері. Очікування відповіді (операція затримки 38). Якщо матеріалу немає, то крок 26.

Крок 25. Ліквідація обслуговуючим персоналом зависання матеріалу в бункері сухих механічним способом (операція затримки 39), перехід на крок 14.

Крок 26. Заповнення обслуговуючим персоналом бункера матеріалом (операція затримки 310). Після підтвердження наявності матеріалу перехід на крок 14.

Крок 27. Натискання кнопок F4 (робоча операція P16) і F8 (робоча операція P17) на операторній панелі для зупинки шнека і закриття заслінок.

Крок 28. Переключити тумблер установки режиму управління на пульті в положення «АВТ» (рис. 2.2) (робоча операція P18).

Крок 29. Натискання кнопки «ДЕБЛОКИРОВКА» на пульті (рис. 2.2) для скидання аварійного повідомлення (робоча операція P19).

Крок 30. Натискання кнопки «ПУСК» на пульті (рис. 2.2) для повтору набору компонентів (робоча операція P20).

Крок 31. Перевірка динаміки зміни ваги компонента в дозаторі за показаннями у відповідному полі мнемосхеми форми «Дозатор сухих» (операція затримки 311). Якщо вага змінюється задовільним темпом, то крок 33 «Кінець».

Крок 32. Натискання кнопки «ПРЕРЫВАНИЕ» на пульті (рис. 2.2) (робоча операція P21) і перехід на крок 12.

Крок 33. Кінець (композиціонер «Фінішер»).

Модель даного алгоритму діяльності у вигляді функціональної мережі, з використанням нотації мови узагальненого структурного методу надана на рис. (рис. 2.4).

Показники, що необхідні для розрахунку (згортки) функціональної мережі (рис. 2.4) були отримані з бази ергономічних даних алгоритмів діяльності [9] і проведення експериментів на робочому місці (для контрольних та альтернативних операцій), а також частково отримані в ході виробничого експерименту і враховують конструктивні особливості ЛМІ, що використовується оператором.

Величини затримок 31, 32, 34 були наведені в [24]:

- для затримок 31, 32 математичне сподівання $M=2,0$ с, середнє квадратичне відхилення $S=0,2$ с;
- для затримки 34 математичне сподівання $M=1,95$ с, середнє квадратичне відхилення $S=0,05$ с.

Час затримок 33, 35, 310, розподілений по нормальному закону з M і S відповідно: 900 с і 300 с; 4920 з і 4020 с; 1200 с і 60 с. Час затримки 36 - 39, 311 розподілений рівномірно з M і S відповідно: 10 ± 2 с; 15 ± 5 с; 240 ± 60 с; 105 ± 75 с і 15 ± 5 с.

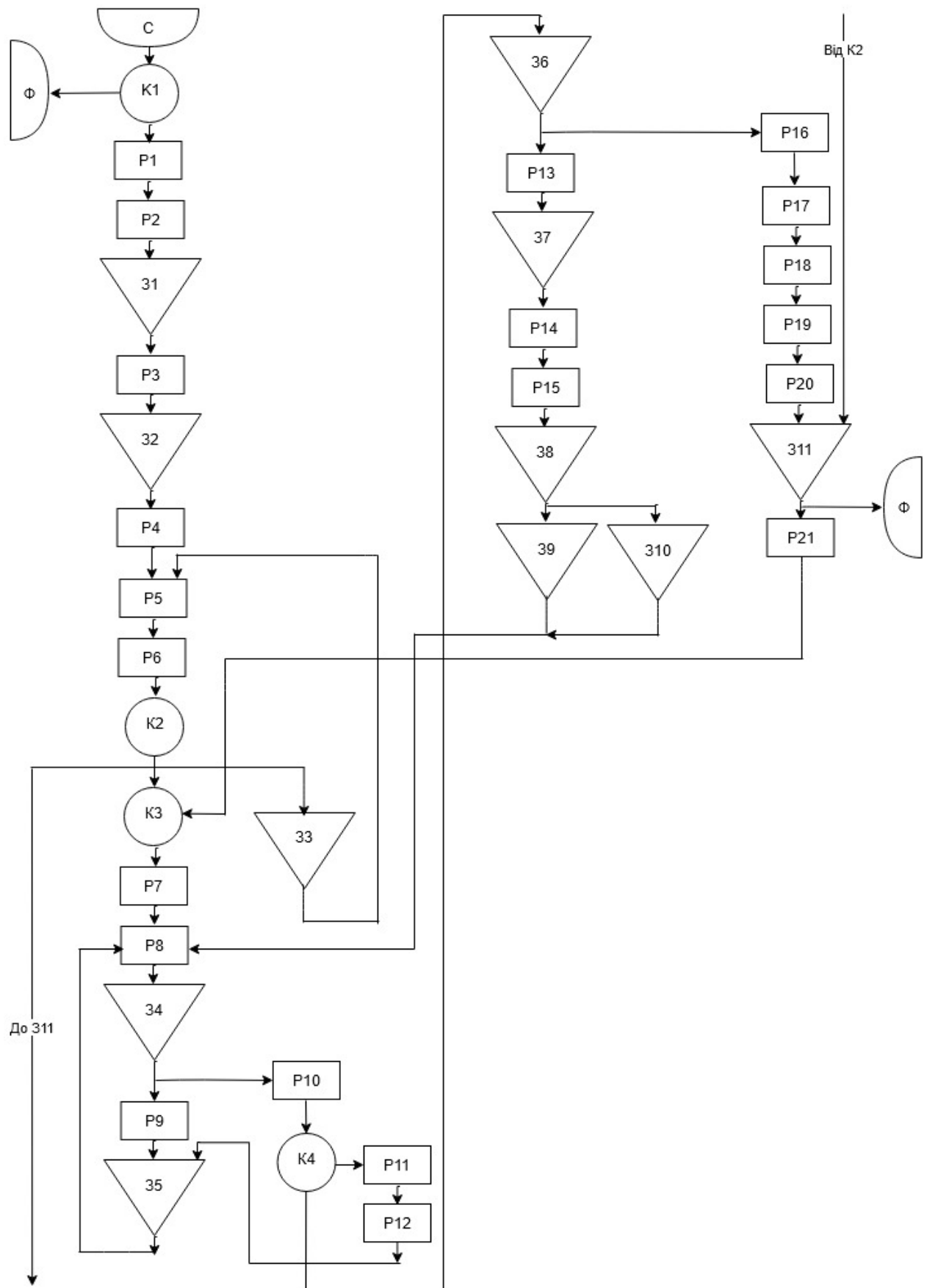


Рисунок 2.4 – Функціональна мережа діяльності для випадку «Відсутність матеріалу в бункері сухих»

Прийmemo наступні позначення:

B^1 - ймовірність безпомилкового виконання;

K^{00} - ймовірність виявлення помилки при її наявності;

K^{01} - ймовірність виявлення помилки при її відсутності ($K^{00} + K^{01} = 1$);

K^{11} - ймовірність виявлення відсутності помилки при її відсутності ($K^{10} + K^{11} = 1$);

K^{10} - ймовірність виявлення відсутності помилки при її наявності ($K^{00} + K^{10} = 1$);

M - математичне сподівання часу виконання;

S - середнє квадратичне відхилення виконання операції

В таблиці 2.1 наведено показники безпомилкового виконання робочих операцій.

Таблиця 2.1 – Показники безпомилкового виконання робочих операцій

Показник		Операція				
		P1	P2, P4-P6, P8, P9, P11-P17, P19-P21	P4	P7, P18	P10
Ймовірність B^1		0,9998	0,9951	0,9979	0,9986	0,9986
Часовий	M, c	1,1	0,48	0,25	0,18	1,1
	S, c	0,11	0,13	0,05	0,02	0,11

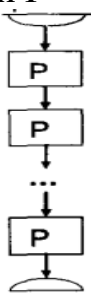
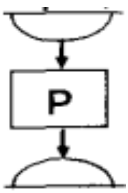
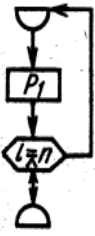
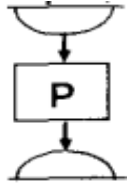
В таблиці 2.2 наведено показники безпомилкового виконання контрольних операцій.

Таблиця 2.2 – Показники безпомилкового виконання контрольних операцій

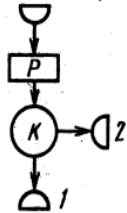
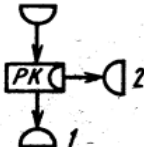
Показник		Операція	
		K1	K2, K3, K4
Ймовірність	K^{11}	0,999	0,994
	K^{00}	0,985	0,970
Часовий	M, c	0,28	1,4
	S, c	0,03	0,5

Далі, згідно з [9] проводиться аналіз функціональної мережі для виявлення в ній типових функціональних структур (ТФС), що представляють деякі типові сукупності типових функціональних одиниць (ТФО), що повторюються. Структура ТФС і вміст типових функціональних одиниць, що в неї входять визначають надійнісні та часові характеристики цієї ТФС. В табл. 2.3 наведена бібліотека ТФС з розрахунковими формулами для визначення основних надійнісних і часових характеристик ТФС [9], які мають місце в розробленій функціональній мережі (рис. 2.4).

Таблиця 2.3 – Бібліотека ТФС з розрахунковими формулами показників

Номер, зміст і схема ТФС	Еквіва- лентна ТФО	Показник	Розрахункова формула
1	2	3	4
1.Послідовне виконання робочих операцій Р 	Робоча операція 	Ймовірність безпомилкового виконання	$B = \prod_{i=1}^n B_i^1$, де i – номер ТФО
		Математичне очікування часу виконання (T)	$M(T) = \sum_{i=1}^n M(T_i)$
		Дисперсія T	$D(T) = \sum_{i=1}^n D(T_i)$
		Операція згортки	$P_{cs}(T_0) = (F_{p1} * \dots * F_{pk})(T_0)$ символ “*” – означає операцію згортки
2. n -кратне повторення робочої операції одного успішного результату 	Робоча операція 	Ймовірність безпомилкового виконання	$B = 1 - (1 - B_1^1)^n$
		Математичне очікування T	$M(T) = nM(T_1)$
		Дисперсія T	$D(T) = nD(T_1)$
		Операція згортки	$P_{cs}(T_0) = (F_p * F_k)(T_0)$

Продовження табл.2.3

1	2	3	4
3. Послідовне виконання робочої операції P і контролю K функціонування 	Робоча операція 	Ймовірність безпомилкового виконання	$B = B^1 K^{11}$
		Математичне очікування T	$M(T) = M(T_p) + M(T_k)$
		Дисперсія T	$D(T) = D(T_p) + D(T_k)$
		Операція згортки	$P_{cv}(T_0) = (F_{p1} * \dots * F_{pk})(T_0)$

Підсумкові показники якості алгоритму функціонування оператора наступні: ймовірність безпомилкового виконання $B^1 = 0,9411$; математичне сподівання $M(T) = 96,354$ с і середнє квадратичне відхилення часу виконання $S(T) = 12,43$ с.

Розроблена математична модель формалізує діяльність оператора і сумісно з математичною моделлю, що формалізує процеси дозування та змішування, є основою для побудови ІМ процесу функціонування АСУТП дозування матеріалів для виготовлення газобетону на ЗАТ «Мотор Січ».

2.2 Формальний опис процесу дозування

Математична модель процесу двокомпонентного дозування сухих компонентів була розроблена в [5]. Її модифікація у вигляді комбінованої (аналітико-імітаційної) моделі [24], дозволила додатково враховувати в процесі дозування вплив динамічної складової ваги набору компонентів в бункери дозаторів і швидкість витікання компонентів з витратних ємностей.

Кінцевим параметром, що характеризує продуктивність дозатора технологічної лінії і системи управління, є ваговий показник i -го компонента газобетону, що визначається за формулою (2.10):

$$G_i(t) = \int_{t_1}^{t_2} F_i(t, q) \cdot \rho_i(t) \cdot g dt, \quad (2.10)$$

де G_i - вага i -го компонента газобетону;

$F_i(t, q)$ - потік i -го компонента з видаткової ємності;

ρ_i - щільність i -го компонента, що дозується;

g - прискорення вільного падіння.

Продуктивність шнекового живильника $P_i(t, q)$ модулює потік компонента. У свою чергу, продуктивність $P_i(t, q)$ залежить від максимальної продуктивності шнекового живильника $P_{maxi}(q)$ та частоти на виході «перетворювач частоти – асинхронний двигун – редуктор» $f_{вих}(t)$ і управління контактором приводу шнека $Q_{упрi}(t)$, що розраховується формулою (2.11):

$$P_i(t, q) = P_{maxi}(q) \cdot f_{вих}(t) Q_{упрi}(t). \quad (2.11)$$

Кожен потік i -го компонента $F(t)$, описується вектором наступних параметрів, які зазначені за формулою (2.12):

$$Fi(t) = \{F_i(t, q), \rho_i(t)\}. \quad (2.12)$$

Потік $F_i(t, q)$ компонентів з видаткової ємності залежить від стану вихідної заслінки $Q_{заслi}(t)$ і знаходиться за формулою (2.13):

$$F_i(t, q) = S_i(t, q, \alpha_3) \cdot v_{flo}(t), \quad (2.13)$$

де $S_i(t, q, \alpha_3)$ – площа відкриття заслінки, яка визначає її пропускну здатність;

$\alpha_3 = \pi/2$ – кут відкриття заслінки для ВКВ та цементу;

$v_{flo}(t)$ – швидкість витікання.

Площу відкриття заслінки обчислюємо за формулою (2.14):

$$S_i(t, q, \alpha_3) = \frac{\pi \cdot D_3^2}{4} \cdot (1 - \cos \alpha_3(t)) \quad (2.14)$$

Потік компонентів у дозатор обчислюємо за формулою (2.15):

$$F_{iA}(t) = \begin{cases} F_i(t, q) = S_i(t, q, \alpha_3) \cdot v_{flo}(t), & \text{якщо } P_i(t, q) \geq F_i(t, q); \\ P_i(t, q) = P_{imax}(q) \cdot f_{вих}(t) \cdot Q_{гупр}(t), & \text{якщо } F_i(t, q) > P_i(t, q). \end{cases} \quad (2.15)$$

Маса компонентів у дозаторі m та маса порожнього дозатора $m_{доз}$ створюють статичне зусилля $P_{стат}$, пропорційне вазі дозатора з компонентами газобетонної суміші $G(t)$. Практичний досвід дозування свідчить, що після закриття заслінки вага в дозаторі ще протягом близько 0,5 с збільшується, а потім стабілізується на меншому значенні. Це явище обумовлено інерційністю виконавчих механізмів і часом стабілізації падіння частинок компонентів від вихідного патрубку заслінки живильника (точка А) до точки падіння (точка В), а також динамічним впливом $P_{дин}$ падаючих компонентів на ємність дозатора протягом часу дозування.

Тобто, у даному випадку необхідно знайти динамічний вплив $P_{дин}(t)$. З цією метою, швидкість падіння частинок компонента v_B , що набирається визначається по висоті падіння частинок компонентів $h(t)$ і часу падіння частинок компонентів τ .

Обсяг дозатора складається з обсягу його циліндричної $V_{цил}$ і конічної $V_{кон}$ (усічений конус) частини, та розраховується формулою (2.16):

$$V_{\text{доз}} = V_{\text{цпл}} + V_{\text{кон}} = \pi \cdot R^2 \cdot h_{\text{цпл}} + \frac{1}{3} \cdot \pi \cdot h_{\text{у.кон}} \cdot (R^2 + R \cdot r + r^2). \quad (2.16)$$

де π – константа 3,14;

R – радіус циліндричної частини дозатора;

r – радіус вихідного отвору дозатора.

Потік компонентів, що дозується в точці В визначається за формулою (2.17):

$$F_{iB}(t) = F_{iA}(t - \tau). \quad (2.17)$$

Висота падіння частинок компонентів $h(t)$ динамічно змінюється під час заповнення дозатору та визначається за формулою (2.18):

$$h(t) = \begin{cases} h_{\text{доз}} - \sqrt[3]{\frac{3 \cdot V(t)}{\pi \cdot t g^2 \alpha}}, & \text{якщо } V(t) < V_{\text{кон}}; \\ \frac{V_{\text{доз}} - V(t)}{\pi \cdot R^2}, & \text{якщо } V(t) > V_{\text{кон}}; \end{cases}, \quad (2.18)$$

де α – кут конуса дозатора;

$h_{\text{доз}}$ – висота бункера дозатора;

$V_{\text{доз}}$ – обсяг бункера дозатора;

$V_{\text{кон}}$ – обсяг конусної частини бункера дозатора.

Обсяг набраних в дозатор компонентів визначається за формулою (2.19):

$$V(t) = \sum_{i=1}^2 \int_{t_1}^{t_2} F_{iB}(t) dt \quad (2.19)$$

Швидкість падаючого компонента поблизу області торкання з поверхнею дозатору вже надійшовших у дозатор компонентів визначається за формулою (2.20):

$$v_B = \tau g = \sqrt{2g h(t)}, \quad (2.20)$$

де g – прискорення вільного падіння.

При абсолютно непружному зіткненні (швидкість після взаємодії дорівнює нулю) динамічна сила впливу падаючого матеріалу визначається [25] за формулою (2.21):

$$P_{\text{дин}}(t) = F_{iB}(t) \cdot \rho_i \cdot v_B = \sqrt{2g h(t)} \cdot F_{iB}(t) \cdot \rho_i(t) \quad (2.21)$$

Сила впливу на тензометричний датчик ваги складається з ваги порожнього дозатора $G_{\text{доз}}$, статичної та динамічної компонент і визначається за формулою (2.22):

$$P_{\Sigma}(t) = G_{\text{доз}} + g \cdot \sum_{i=1}^2 \int_{t_1}^{t_2} F_{iB}(t) \cdot \rho_i dt + v_B \cdot F_{iB}(t) \cdot \rho_i, \quad (2.22)$$

При наборі рідких компонентів, у відповідність з рецептом, використовується трипозиційна кругла поворотна заслінка. Її пропускна здатність залежить від площі відкриття, яка регулюється двома сигналами управління на відкриття заслінки на 90 градусів ($Q_i(t)$) і відкриття заслінки на 30 градусів ($Q_{ig}(t)$), відповідно грубого і точного набору i -го компоненту суміші [25].

Потік i -го компонента на вході дозатора в точці A визначається параметром потоку $F_i(t, q)$ компонентів з видаткової ємності, розраховується за формулою (2.23):

$$F_{iA} = F_i(t, q) = \mu S_i(t, q, \alpha_3) \cdot v_{ifloA}(t), \quad (2.23)$$

де $\mu_i = 0,95$ – коефіцієнт витрат;

$v_{iflo}(t)$ – швидкість витікання компонентів з видаткової ємності.

Швидкість витікання компонентів з видаткової ємності в початковому наближенні прийнята постійною, та визначається формулою (2.24) [26]:

$$v_{ifloA}(t) = \sqrt{\Delta p \cdot \frac{2g}{\rho_i}}, \quad (2.24)$$

де Δp – перепад тиску на виході заслінки (змінюється в залежності від рівня компонента у видатковій ємності);

ρ_i – щільність компонента.

В процесі дозування потрібна ідентифікація швидкості витікання потоку компонентів і, з урахуванням цього, зміна моменту перемикання набору (грубо / точно) і стабілізація потоку компонентів, що дозуються. Вода подається в дозатор не з накопичувальної ємності, а по трубопроводу з тиском в 5 бар. Тому, для розрахунку її швидкості витікання у формулі (2.24) необхідно прийняти Δp постійною і рівною 500000 Па.

При падінні потоку компонента, виходячи з умови збереження енергії, його швидкість в момент торкання поверхні (точка В) визначається за формулою (2.25):

$$v_{iB}(t) = \sqrt{v_{ifloA}^2(t) + 2gh(t)}. \quad (2.25)$$

Потік в цій точці визначається відповідно до формули (2.17), а висота падіння частинок і обсяг набраної води розраховуються за формулами (2.18) та (2.19) відповідно. Сумарна вага дозатора розраховується послідовним перетворенням потоку, що змінюється рідких компонентів газобетонної суміші в вагові показники в дозаторі. Потоки на виході відповідних дозаторів розраховуються за формулами (2.23) і (2.24), а сила впливу (маса компонентів) на тензометричний датчик ваги дозаторів за формулою (2.22).

Зоровий аналізатор людини-оператора не в змозі відслідковувати динаміку набору маси компонентів в дозатори з цифрового поля виведення на екрані ЛМІ. Але йому важлива тільки фіксація моменту кінцевої набраної маси компонента. Тому, можна прийняти потік в точці В рівним потоку на вході в дозатор (точка А), тобто $F_{iB}(t) = F_{iA}(t)$ і у формулі (2.22) враховувати тільки статичну складову:

$$P_{\Sigma}(t) = \sum_{i=1}^2 \int_{t_1}^{t_2} F_{iB}(t) \cdot \rho_i dt. \quad (2.26)$$

Таким чином, моделі що розглянуто вище, формально описують як обладнання, виконавчі механізми АСУТП, датчики технологічного процесу, так і взаємодію оператора з системою управління технологічним процесом. Подальше імітаційне моделювання ТП на базі цих моделей дозволить кількісно оцінити результати діяльності оператора.

2.3 Висновки до другого розділу

Проведений огляд видів операторської діяльності показав, що діяльність оператора АСУТП дозувально-змішувального відділення ЗАТ «Мотор Січ», що досліджується у даній роботі, є діяльністю оператора-технолога (залежно від виду автоматизованої системи та ступеня ієрархії, на якій знаходиться людина-оператор у цій системі).

Для формалізації потоку інформації, що надходить до оператора, використовують математичний апарат з теорії інформації, а саме ентропією системи. Загальна кількість інформації, котра переробляється оператором залежить як от ентропії джерела повідомлень, кількості додаткової інформації, яка використовується оператором при вирішенні задачі, так і від кількості втраченої при передачі інформації.

Для усунення недоліків підходів, що використовуються в теорії інформації доцільно, в рамках даного дослідження, розробити аналітико-імітаційну модель.

Для проведення ергономічної експертизи діяльності оператора в модель діяльності введена батарея наступних показників: коефіцієнт завантаженості z ; максимальний період зайнятості $T_{\text{зан}}$; коефіцієнт черги ρ ; середнє число сигналів в черзі $K_{\text{доп}}$.

Для формального опису процесу функціонування оператора лінії приготування газобетонної суміші ЗАТ «Мотор Січ» було використано узагальнений структурний метод А.І. Губінського та розроблена функціональна мережа діяльності оператора. Таким чином було отримано модель діяльності оператора, яка сумісно з відомими моделями, що формалізують процеси дозування та змішування, є основою для побудови ІМ процесу функціонування АСУТП дозування матеріалів для виготовлення газобетону на ЗАТ «Мотор Січ».

3 РОЗРОБКА ІМІТАЦІЙНОЇ МОДЕЛІ ДІЯЛЬНОСТІ ОПЕРАТОРА АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ

3.1 Моделювання в середовищі SIMC

В результаті проведеного у розділі 1 аналізу парадигм та систем імітаційного моделювання, у якості СІМ для моделювання діяльності оператора газобетонного виробництва було обрано систему SIMC [18].

Система імітаційного моделювання загально-цільового призначення SIMC призначена для моделювання систем масового обслуговування дискретного типу. SIMC є закінченою системою, тобто включає в себе: засоби розробки, налагодження і виконання імітаційних моделей, збору і обробки статистики.

Для створення імітаційних моделей та їх формального опису використовуються дискретно-подієвий підхід та транзактний спосіб організації квазіпаралелізму відповідно.

Імітаційні моделі оформлюються у вигляді типових процедур на мові C++ та процедур, функцій розробленої спеціалізованої бібліотеки (у вигляді файлу `simc.cpp`). Для реалізації функціональності даної бібліотеки і забезпечення працездатності імітаційної моделі, що виконується розроблена компонента доповнення `simc.h` для середовища розробки Qt Creator.

Структура системи SIMC представлена на рис. 3.1

Підсистема початку моделювання забезпечує створення модельного середовища шляхом ініціалізації системних списків, необхідних для забезпечення коректного старту імітаційної моделі, введення, зберігання та маніпуляції з початковими даними моделювання.

Підсистема "Планувальник подій" складається з процедур роботи зі списками, а також планування і переміщення транзактів (абстрактних динамічних об'єктів), за допомогою яких моделюються заявки, що надходять до системи масового обслуговування.

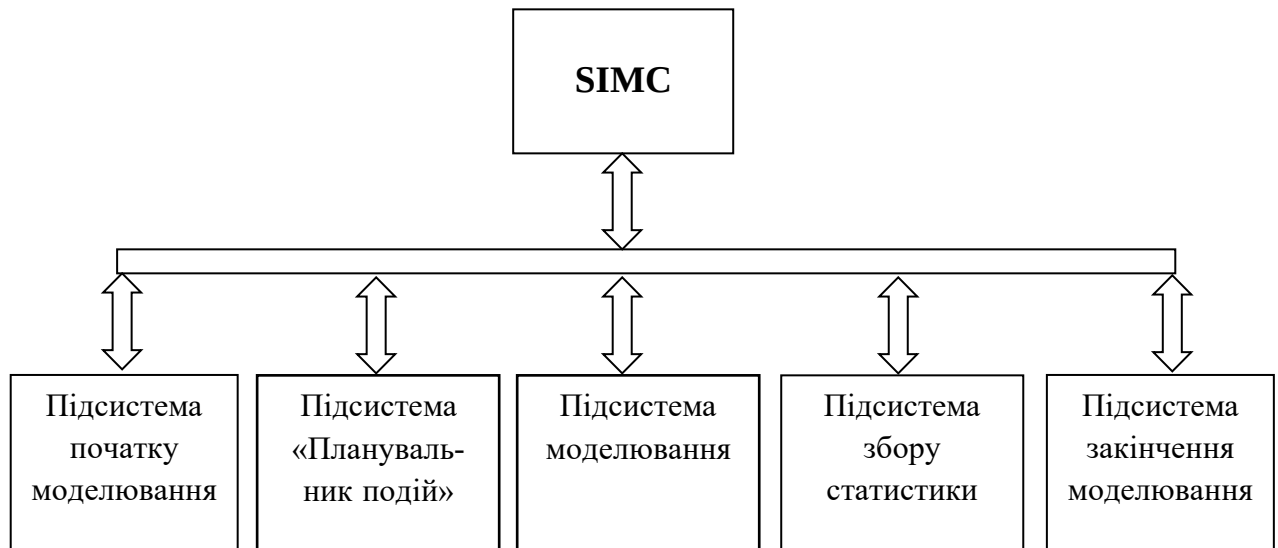


Рисунок 3.1 – Загальна структура системи SIMC

Підсистема моделювання складається з бібліотеки процедур `simc.h`, необхідної для створення імітаційної моделі, а також модуля, що складається з процедур-поглиначів і процедур, що дозволяють здійснювати налагодження і діагностику логічних помилок в ІМ.

Підсистема збору статистики забезпечує: генерацію 10 відомих теоретичних законів розподілу випадкових величин; збір статистики про скупчення транзактів скидання статистики та очищення системного та модельного середовища.

Підсистема закінчення моделювання забезпечує: взаємодію з підсистемою планування подій; виконання процедур друку списків, які використовуються в імітаційній моделі, гістограм і статистики про стан системи або окремих об'єктів моделювання; формування файлу результатів моделювання в форматі HTML.

Реалізація системи моделювання SIMC з використанням середовища програмування Qt Creator забезпечує запуск системи на більшості сучасних операційних систем, шляхом компіляції тексту програми для кожної операційної системи без зміни сирцевого коду.

Програмний комплекс працює на всіх IBM-сумісних комп'ютерах з сокетом 775 і більш пізніми моделями, а також, при наявності монітора SVGA, маніпулятора типу "миша" і середовища програмування Qt Creator. Рекомендована кількість оперативної пам'яті - 2 Гб і вище, а також вільне місце на жорсткому диску 20 Мб (допустимий мінімум необхідний для зберігання файлу результатів моделювання).

Для коректного читання результатів моделювання з файлу результатів на персональному комп'ютері повинен бути встановлений Internet-браузер типу Internet Explorer, Mozilla Firefox і т.п.

Так як імітаційна модель виконується у вигляді процедур на мові C ++, то для запуску системи моделювання SIMC необхідно відкрити розробленої в середовищі програмування Qt Creator проєкт імітаційної моделі і запустити на виконання (меню Run) файл `simc.cpp`. Відповідно, список файлів з вихідним кодом (Sources), які використовуються при складанні проєкту повинен включати файли `simc.cpp` і `simc.h`.

Компонента `simc.h` підключається до виконуваного модулю у вигляді стандартних директив препроцесора `include` на етапі компіляції. На етапі розробки імітаційної моделі необхідно слідувати правилам розробки програм прийнятих в середовищі програмування Qt Creator.

Перед початком моделювання необхідно за допомогою процедур та функцій бібліотеки створити системне середовище: системні списки; необхідну кількість транзактів та надати початкові значення системним змінним.

Після запуску імітаційної моделі, в вікні, слід ввести ім'я файлу, в якому в форматі `.html` будуть збережені результати моделювання.

Процес моделювання полягає в багаторазовому повторенні в циклі `while` управляючої програми моделювання у вигляді функції `void plan()` і блоку вибору події `switch(sysevent)`. У середині блоку вибору події оператор `switch(sysevent)` вибирає подію, номер якої збігається зі значенням `sysevent`.

Після закінчення моделювання система SIMC здійснює обробку, зберігання і документування зібраної статистики.

3.2 Розробка алгоритму роботи імітаційної моделі процесу дозування компонентів суміші газобетону

Алгоритм роботи імітаційної моделі заповнення двохкомпонентного дозатора компонентами суміші газобетону наступний.

Крок 1. Завантаження сухих компонентів.

Крок 1.1 Задаємо дискретність моделювання процесу, що дорівнює одній секунді та розраховуємо початкові дані:

- частоту обертання шнеків. Частота обертання в моделі задається в % від номінальної, обмеження в 100%, швидкість зростання 100% за одну секунду);
- площу відкриття заслінки. Діаметр заслінки ВКВ і цементу $D_3=250$ мм, час її відкриття 1с;
- швидкість витікання компонентів. Швидкість витікання $v_{flo}(t)$ для ВКВ - 0,5 дм³/с, цементу - 100 дм³/с.

Крок 1.2 Обчислюємо продуктивність шнекового живильника та сам потік компонентів. Для ВКВ прийнята продуктивність шнека $P_{шн}=20/1.5$ м³/годину при частоті обертання 100% . Частота змінюється на 20% у випадку точного дозування, та на 100% для грубого.

Потік компонентів з видаткової ємності обчислюємо за формулою (2.13). Площу відкриття заслінки обчислюємо за формулою (2.14).

Крок 1.3 Розраховуємо потік компонентів на виході з шнекового живильника (потік у дозатор) за формулою (2.15).

Крок 1.4 Розраховуємо дані під час засипання в дозатор:

- час падіння частинок розраховуємо за формулою (3.1):

$$\tau = \sqrt{\frac{2h(t)}{g}} \quad (3.1)$$

Вектор параметрів дозатора (у метрах) для ВКВ та цементу наступний: Height_Vector=[0, 0.075, 1], Radius_Vector=[0.1, 0.95, 0.95]. Параметром дозатора є три точки фігури обертання. Наприклад, Height_Vector=0 – це самий низ дозатора, при цьому Radius_Vector=0.1;

- потік компонентів, що дозуються в точці падіння В обчислюємо за формулою (2.17);
- обсяг набраних компонентів (ВКВ та цементу) обчислюємо за формулою (2.19);
- висоту падіння частинок h обчислюємо за формулою (2.18). Висота падіння по мірі заповнення дозатору змінюється від 1 м до 0 м;
- швидкість падаючого компонента, що вже надійшов у дозатор, поблизу області торкання з поверхнею дозатору, обчислюємо за формулою (2.20).

Крок 1.5 Обчислюємо силу впливу (масу компонентів) на тензометричний датчик ваги за формулою (2.22).

Крок 1.6 Отримуємо сигнал зворотного зв'язку від тензометричного датчика, який інформує систему автоматизованого управління заповнений дозатор чи ні.

Крок 2. Завантаження рідких компонентів (суспензія, гаряча вода, шлам).

Крок 2.1 Розраховуємо початкові дані: площа відкриття заслінки (по формулі 2.14); швидкість витікання води; кінцеві директивні значення ваги кожного компонента. При цьому враховуємо, що при завантаженні вхідний діаметр заслінки для гарячої води та суспензії дорівнює 50 мм, а шламу – 150 мм. Заслінка може мати проміжні положення 40% від куту 90 градусів.

Вихідний діаметр заслінки для гарячої води та суспензії дорівнює 80 мм, а шламу – 250 мм. Швидкість витікання гарячої води визначається

продуктивністю насосу (1000 м³/годину), а суспензії та шламу дорівнює 20 та 10 дм³/с відповідно.

Крок 2.2 Обчислюємо потік компонентів на вході дозатора в точці А (вихід з заслінки) за формулою (2.23).

Щільність шламу та суспензії відповідно складає 2,65 та 2,7 кг/дм³. Для підвищення точності моделювання, перепад тиску розраховується для пяти рівнів компонентів з шагом 20% за формулою (3.2):

$$\Delta p = P_{\text{комп}} - P_{\text{пов}} = (\rho_i \cdot g \cdot h) - P_{\text{пов}}, \quad (3.2)$$

де $P_{\text{комп}}$ – тиск компоненту;

$P_{\text{пов}}$ – повітряний тиск;

$h=1$ м – висота видаткових ємностей.

Крок 2.3 Розраховуємо дані під час надходження компонентів в дозатори: обсяг набраної води та висоту падіння частинок.

Параметри дозаторів гарячої води, суспензії та шламу наведені в табл. 3.1.

Таблиця 3.1 – Параметри дозаторів гарячої води, суспензії та шламу

Параметри	Гаряча вода	Суспензія	Шлам
Height Vector, м	[0, 0.075, 1]	[0, 0.075, 1]	[0, 1.2, 2.6]
Diameter Vector, м	[0.1, 0.95, 0.95]	[0.1, 0.95, 0.95]	[0.264, 1.2, 1.3]

Крок 2.4 Розраховуємо дані під час надходження компонентів в дозатори – обсяг набраної води, висоту падіння частинок згідно формулам (2.26) та (2.18) відповідно.

Крок 2.5 Розраховуємо потоки на виході відповідних дозаторів згідно формулі (3.3) та вихідними даними, наведеними на кроці 2.1:

$$F_{idoz}(t, q) = F_i(t, q) \cdot P_{ин}. \quad (3.3)$$

Крок 2.6 Обчислюємо силу впливу (масу компонентів у дозаторі) на тензOMETричний датчик ваги дозаторів по формулі (2.22).

3.3 Розробка алгоритму роботи імітаційної моделі діяльності оператора

Укрупнений алгоритм роботи оператора на початку робочої зміни (перший цикл виробництва) наступний (в дужках вказані елементи ЛМІ, що використовуються при виконанні відповідної дії).

Крок 1. Включити рубильник живлення і дочекатися завантаження системи (пульт управління (ПУ)).

Крок 2. Перевірити наявність системних помилок (ці помилки досить рідкісні і в моделі не враховувалися). Перевірка помилок моделювалася затримкою 0,4 с на час фіксації погляду на поле виведення аварійних повідомлень (рис.2.3) (операційна панель (ОП)).

Крок 3. Перехід на форму «Общий вид» (ОП).

Крок 4. Ввімкнути систему автоматики (ПУ).

Крок 5. Перевірка нульових значень в трьох дозаторах - «Суспензия», «Шлам», «Сухие» (ОП). Імовірність налипання матеріалу на дозаторах, що приводить до ненульових значень ваги у дозаторі становить 0,99. При відсутності налипань, перехід на крок 7, інакше крок 6.

Крок 6. Усунення налипань (ОП). Час виконання операції обдування (включення вібраторів в дозаторі сухих) рівномірно розподілений і дорівнює 50 ± 10 с.

Крок 7. Вибір рецепта (ОП). Для моделювання обрано рецепт приготування газобетону марки D600, який є шостим з 11 в списку меню рецептів. Імовірність правильного виконання операцій знаходження і зчитування рецепта оператором становить 0,9919, а середній час $M(T)$ складає 1,128 с зі стандартним відхиленням $S = 0,09$ с. Ймовірність виконання операцій розрахована відповідно до формул ОСМ і на основі довідкових даних [9] про можливості правильного виконання і середньому часу виконання таких операцій як перемикання тумблера, натискання кнопки і т.п.

Крок 8. Запуск лінії (ОП) - початок технологічного процесу.

Крок 9. Набір компонентів суміші (ОП). Через рівномірно розподілений інтервал часу 35 ± 5 с після початку набору, оператор в екрані «Общий вид» контролює набір ваги компонентів протягом інтервалу 4 ± 1 с.

Крок 10. Вивантаження шламу і сухих компонентів в змішувач (ОП).

Крок 11. Перемішування компонентів протягом рівномірно розподіленого інтервалу часу 60 ± 3 с (ОП).

Крок 12. Вивантаження суспензії в змішувач (ОП).

Крок 13. Перемішування компонентів протягом рівномірно розподіленого інтервалу часу 220 ± 20 с (ОП).

Крок 14. Вивантаження суміші із змішувача (ОП). Кінець першого цикл виробництва.

Крок 15. Початок другого циклу виробництва (використовується тільки ОП на усіх кроках) - аналіз ваг набраних і вивантажених матеріалів. Імовірність того, що буде дисбаланс компонентів становить 0,75. Дисбаланс води, суспензії і шламу усувається автоматично. Якщо дисбаланс відсутній, то виконати кроки 8-14, інакше крок 16.

Крок 16. Усунення дисбалансу. Налаштування дозатора сухих.

Крок 17. Виконати кроки 8-14. Кінець другого циклу виробництва.

На всіх наступних циклах протягом робочого дня оператором повторюються кроки 8-14.

3.4 Розробка імітаційної моделі процесу виготовлення газобетону

Імітувалась діяльність оператора АСУТП дозування матеріалів лінії приготування газобетонної суміші ЗАТ «Мотор Січ» у Запоріжжі. Ділянка працює по дві восьми годинні зміни в день, п'ять робочих днів в тиждень.

Кожному потоку заявок i -го типу (компонентів суміші, несправності тощо), що надходять на обробку відповідає свій потік транзактів. Обслуговування транзактів моделюється за допомогою двох блоків - `seize` (обслуговуючий прилад) і `delayt` (затримка). Просування транзактів по моделі здійснює керуюча програма моделювання (процедура `plan`). Переміщення транзактів тягне за собою виконання певних подій, що відображають динаміку реального ТП.

Для імітації тривалості зміни введено змінну.

```
int time8 = 8*60*60, // 8-годинний робочий день (с).
```

Для імітації оператора, та технологічного обладнання, в ІМ введено відповідні обслуговуючі прилади:

```
newfac(oper, "Operator"); // прилад-оператор
```

```
newfac(f_shlam_water, "Sludge and water dispenser"); // прилад -дозатор  
шламу та води
```

```
newfac(f_syxie, "Dry component dispenser"); // прилад -дозатор сухих
```

```
newfac(f_sysp, "Suspension dispenser"); // прилад -дозатор суспензії
```

```
newfac(f_smes, "Mixer"); // прилад -змішувач
```

Для забезпечення збору статистики щодо зайнятості оператора у нормальному режимі роботи та при виникненні неполадок у ТП, в ІМ введено два відповідних об'єкти типу «Черга»:

```
newqueue(q_work, "Zan_Oper");
```

```
newqueue(q_error, "Zan_Oper_Nepoladki");
```

Для забезпечення відображення статистики щодо зайнятості оператора впродовж зміни (діапазон від 0 до 480 хвилин), та протягом одної години (діапазон від 0 до 60 хвилин) в ІМ введено два об'єкти «Гістограма»:

```
newhist(h, 0, 480, 21, 1,"Zanyatosti operatora"); // створення гістограми
зайнятості оператора
```

```
newhist(h, 0, 60, 30, 1,"Operator Employment per hour"); // створення
гістограми зайнятості оператора впродовж години
```

Цикл while обмежує час моделювання 30 днями, двома змінами з 8-годинним робочим днем. Надходження потоків неполадок у модель описується потоками транзактів, що надходять у модель з початком моделювання (початком робочої зміни у реальній системі).

Для забезпечення моделювання відповідних кроків виконання ТП приготування газобетону, використовується конструкція у вигляді блоку:

```
switch(sysevent) {
    case 1: .....break;
    .....
    case n: .....break;
}
```

що буде повторюватися у циклі while.

У якості приклада, розглянемо реалізацію перших 5 кроків розробленого у п.3.3 укрупненого алгоритму роботи оператора на початку робочої зміни з зазначенням ймовірності (P) та часу виконання операцій (T).

```
case 2: // Включити рубильник і дочекатися завантаження системи. T = 30
± 2 с
```

```
    delayt(randab(28, 32, v1)); break;
```

```
case 3: // Перевірка, чи немає системних помилок. Помилки немає P = 1;
T = 0,4 с
```

```
    delayt (0.4); break;
```

```
case 4: // Натиснути кнопку «Общий вид» P=0,9973; T=0,18±0,03 с. При
помилці оператора - повторити T=0,18±0,03 с
```

```
    if(randab(0, 10000, v3) >= 9973)
```

```

k = 2;
delayt(randab(k*0.15, k*0.21, v2));
k = 1; break;
case 5: // Перехід на форму «Общий вид»  $T=2,0\pm0,2$  с
delayt(randab(1.8, 2.2, v4));
break;
case 6: // Ввімкнути тумблер «ЦЕПИ УПРАВЛЕНИЯ»  $P=0,9979$ ;  $T=0,64\pm0,02$ 
с. При помилці оператора - повторити  $T=0,64\pm0,02$  с
if(randab(0, 10000, v3) >= 9979)
k = 2;
delayt(randab(k*0.62, k*0.66, v5));
k = 1; break;
// Перевірка нульових значень в 3-х дозаторах «Суспензия», «Шлам»,
«Сухие»
case 7: // Зчитування першого значення:  $T=0,150 + 0,028=0,178\pm0,018$  с;
 $P=0,998$ . При помилці оператора - повторити  $T=0,178\pm0,018$  с.
if(randab(0, 10000, v3) >= 9980)
k = 2;
delayt(randab(k*0.16, k*0.196, v6));
k = 1; break;
case 8: // Перевірка логічної умови (тобто що значення не 0)  $T=0,29\pm0,11$ 
с;  $P=0,9966$  При помилці оператора - повторити  $T=0,29\pm0,11$  с
if(randab(0, 10000, v3) >= 9966)
k = 2;
delayt(randab(k*0.18, k*0.4, v6));
k = 1; break;
case 9: // Саккада + Зчитування другого значення:  $T=0,328\pm 0,068$  с;
 $P=0,998$ ; При помилці оператора - повторити  $T=0,178\pm0,018$  с
if(randab(0, 10000, v3) >= 9980)
delayt(randab(k*0.16, k*0.196, v6));
else
delayt(randab(k*0.26, k*0.396, v6)); break;
case 10: // Перевірка логічної умови (тобто що значення не 0). При
помилці оператора - повторити  $T=0,29\pm0,11$  с
if(randab(0, 10000, v3) >= 9966)
k = 2;
delayt(randab(k*0.18, k*0.4, v6));
k = 1; break;
case 11: // Саккада + Зчитування третього значення:  $T=0,328\pm 0,068$  с;
 $P=0,998$ ; При помилці оператора - повторити  $T=0,178\pm0,018$  с
if(randab(0, 10000, v3) >= 9980)
delayt(randab(k*0.16, k*0.196, v10));

```

```

else
    delayt(randab(k*0.26, k*0.396, v10)); break;
case 12: // Перевірка логічної умови (тобто що значення не 0). При
помилці оператора - повторити  $T=0,29\pm0,11$  с
    if(randab(0, 10000, v3) >= 9966)
        k = 2;
    delayt(randab(k*0.18, k*0.4, v6));
    k = 1; break;

```

Розглянемо реалізацію аварійної ситуації налипання цементу та її усунення оператором (крок 6 укрупненого алгоритму роботи оператора розробленого у п.3.3):

```

case 20: // Натиснути кнопку F11 (відкрити заслінку на вивантаження).
Потім включити обдування (вібратори), тобто натиснути кнопку F10:  $T=50\pm10$  с.
    delayt(randab(40, 60, v6)); break;
case 21: // Час переходу до екрану «Дозатор сухих»  $T=2,0\pm0,2$  с
    delayt(randab(1.8, 2.2, v6)); break;
case 22: // Натиснути кнопку F11 (відкрити заслінку на вивантаження) -
 $P=0,9951$ ;  $T=0,48\pm0,13$  с. При помилці оператора - повторити  $T=0,48\pm0,13$  с.
    if(randab(0, 10000, v3) >= 9951)
        k = 2;
    delayt(randab(k*0.35, k*0.61, v6));
    k = 1; break;
case 23: // Час на відкриття заслінки  $1,95\pm0,05$  с.
    delayt(randab(1.9, 2.0, v6)); break;
case 24: // Потім включити обдув (вібратори), тобто натиснути кнопку F10
-  $P=0,9951$ ;  $T=0,48\pm0,13$  с. При помилці оператора - повторити  $T=0,48\pm0,13$  с.
    if(randab(0, 10000, v3) >= 9951)
        k = 2;
    delayt(randab(k*0.35, k*0.61, v6));
    k = 1; break;
case 25: // Обдув дозатора. Час обдуву  $T=50\pm10$  с.
    delayt(randab(40, 60, v6)); break;
case 26: // Виключення обдуву (натискання кнопки F10)  $P=0,9951$ ;
 $T=0,48\pm0,13$  с. При помилці оператора - повторити  $T=0,48\pm0,13$  с.
    if(randab(0, 10000, v3) >= 9951)
        k = 2;
    delayt(randab(k*0.35, k*0.61, v6));
    k = 1; break;

```

Повний текст програми імітаційної моделі наведено у Додатку А.

3.5 Висновки до третього розділу

Розглянуто технологію побудови ІМ за допомогою спеціалізованої мови системи імітаційного моделювання загально-цільового призначення SIMC та структуру цієї системи.

Розроблено наступні алгоритми роботи імітаційної моделі:

- процесу дозування;
- діяльності оператора.

На основі розроблених алгоритмів побудовано імітаційну модель лінії приготування газобетонної суміші ЗАТ «Мотор Січ» у яку включено діяльність оператора АСУТП дозування матеріалів.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Вихідні дані для імітаційного експерименту

Вихідними даними для моделювання були історичні дані про помилки і несправності в АСУТП дозування матеріалів для виготовлення газобетону на ЗАТ «Мотор Січ» за 2018 рік, що отримані з архівів і буферів аварійних повідомлень, а також [10], [24].

Для моделювання було обрано три рецепти приготування газобетону - №1, №6 та №9 які наведені у табл. 4.1.

Таблиця 4.1 – Рецепти приготування газобетону

№ рецепту	ВКВ, кг	Цемент, кг	Суспензія, кг	Шлам, кг	Вода, кг
№1	100	110	25	300	50
№6	180,3	76,5	17,4	323,1	120,1
№9	160	81	14	290	140

Діаметри заслінок набору цементу і ВКВ дозатора сухих дорівнює 125 мм. Діаметри вхідних заслінок: дозатор шламу – 100 мм і 50 мм (для води); суспензії – 40 мм. Діаметри вихідних заслінок: змішувача – 250 мм; дозаторів сухих – 200 мм, суспензії – 50 мм і шламу – 100 мм. Час відкриття і закриття всіх заслінок рівномірно розподілений і дорівнює $1,95 \pm 0,05$ с. Швидкість витікання ВКВ прийнято $0,050 \text{ м}^3/\text{с}$. Продуктивність шнекового живильника становила $3,7 \times 10^{-3} \text{ м}^3/\text{с}$. Щільність шламу, суспензії і суміші становить $2,65 \times 10^3$, $2,7 \times 10^3$ і $1,8 \times 10^3 \text{ кг / м}^3$ відповідно. Перепад тиску при зниженні рівня в дозаторах шламу і суспензії враховувався для п'яти рівнів з кроком 20%. Директивний час для всіх рецептів не повинен перевищувати 17 хвилин при дотриманні точності дозування для шламу і води 1%, для суспензії і сухих 0,5%.

Крім можливих помилок оператора, в ІМ були враховані наступні несправності, що найбільш часто зустрічаються в ході ТП:

зависання матеріалу в бункері сухих (несправність надходить в модель відповідно до розподілу Пуассона з середньою інтенсивністю $\lambda = 1,04 \times 10^{-3}$ подій/с);

– перегрів електроприводу шнека (надходить в модель відповідно до розподілу Пуассона з середньою інтенсивністю $\lambda = 3,47 \times 10^{-5}$ подій/с);

– попадання в шнек стороннього тіла (розподіл Пуассона, $\lambda = 2,31 \times 10^{-5}$ подій/с); попадання на дозатори сторонніх предметів (розподіл Пуассона, $\lambda = 3,5 \times 10^{-6}$ подій/с);

– події «Відсутність матеріалу в бункері» суспензії, ВКВ і сухих матеріалів відбуваються з вірогідністю 0,03.

Час саккади ока $t_{Si} = 0,15$ с. Величини затримок 31-310 були наведені в п. 2.1.2. У разі, якщо дії оператора при перегріванні електроприводу шнека або попадання в шнек стороннього тіла не дали позитивного результату (ймовірність цієї події 0,02), він зобов'язаний викликати електрика для їх ліквідації. Час виклику електрика і ліквідації неполадок розподілений рівномірно і складає 90 ± 30 с, 540 ± 60 с і 1200 ± 600 с відповідно.

Частка аварій і несправностей дозатора сухих склала 90% від їх загальної кількості, тому в імітаційної моделі несправності інших дозаторів не враховувалися.

У зв'язку з тим, що вода підводиться до ділянки по стаціонарному трубопроводу, а витратні ємності шламбасейну мають великий обсяг і наповнюються з відділення помелу, ці ресурси вважалися необмеженими. При моделюванні ліквідації інших перерахованих вище причин несправностей враховувалося наступне:

– час приготування ВКВ і цементу (заповнення прийомних бункерів) розподілений рівномірно в інтервалі 750 ± 450 с;

– включення обдування на кроці 6 з ймовірністю 0,25 може не привести до збільшення ваги на дозаторі (зависання не було усунуто). Тоді

оператор усуває зависання протягом 105 ± 75 с використовуючи механічних знарядь праці;

- в разі ідентифікації перегріву приводу (в полі помилок - помилка №7 «Нет подтверждения работы частотного привода при наборе ИКВ!!» Або №21 «Нет подтверждения работы частотного привода при наборе цемента!»)

Оператор викликає електрика протягом розподіленого рівномірно інтервалу часу 90 ± 30 с. Час усунення несправності електриком розподілений рівномірно в інтервалі 540 ± 60 с;

- час видалення стороннього тіла з шнека розподілений рівномірно в інтервалі 1200 ± 600 с. При цьому оператор зайнятий протягом 410 ± 90 с;

- час візуального огляду і видалення оператором сторонніх предметів з дозатора сухих розподілений рівномірно в інтервалі 65 ± 50 с.

4.2 Результати імітаційного експерименту

Як приклад, на рис. 4.1 представлені результати моделювання діяльності оператора першої зміни, а також гістограма зайнятості оператора першої та другої зміни (рис. 4.2) за весь 15-й день моделювання (в усталеному режимі роботи) для рецепту №6.

Як впливає з результатів експерименту, наведених на рис. 4.1 і рис. 4.2, жоден з показників, що характеризують діяльність оператора, не перевищив рекомендованого в п. 2.1.2 значення.

Наприклад, максимальна зайнятість оператора була у оператора першої зміни в останній час його роботи – 10 хвилин (рис.4.2), що не перевищує показника 15 – 20 хвилин [19].

Підсумкові результати імітаційного моделювання операторської діяльності протягом 30-ти днів з двозмінним восьмигодинним робочим днем для рецепту №6 представлені на рис. 4.3.

СИМ-СИ++ v1.2	ЗНТУ	2016
Ж.К.Каминская С.Н. Сердюк		

Statistics for the 15th 8-hour shift:

Operator rate of employment $Z = 0.08$.
Coefficient of queue $p = 0.07$.

Maximum operator's busy time $T_{\max} = 38.19$ min.
Max queue length $K_{\max} = 3$.
The average queue length $K_{\text{sr}} = 1.98$.

Total time in queue = 564.51 min.
Number of queue entries = 4.00 .
Average waiting time in queue = 141.13 min.

Average processing time of information by the operator = 0.63 min.

Рисунок 4.1 – Результати моделювання діяльності оператора першої зміни
протягом 15-го дня моделювання

The histogram shows the operator's employment during the working day.
Employment operator's periods are shown every one hour.

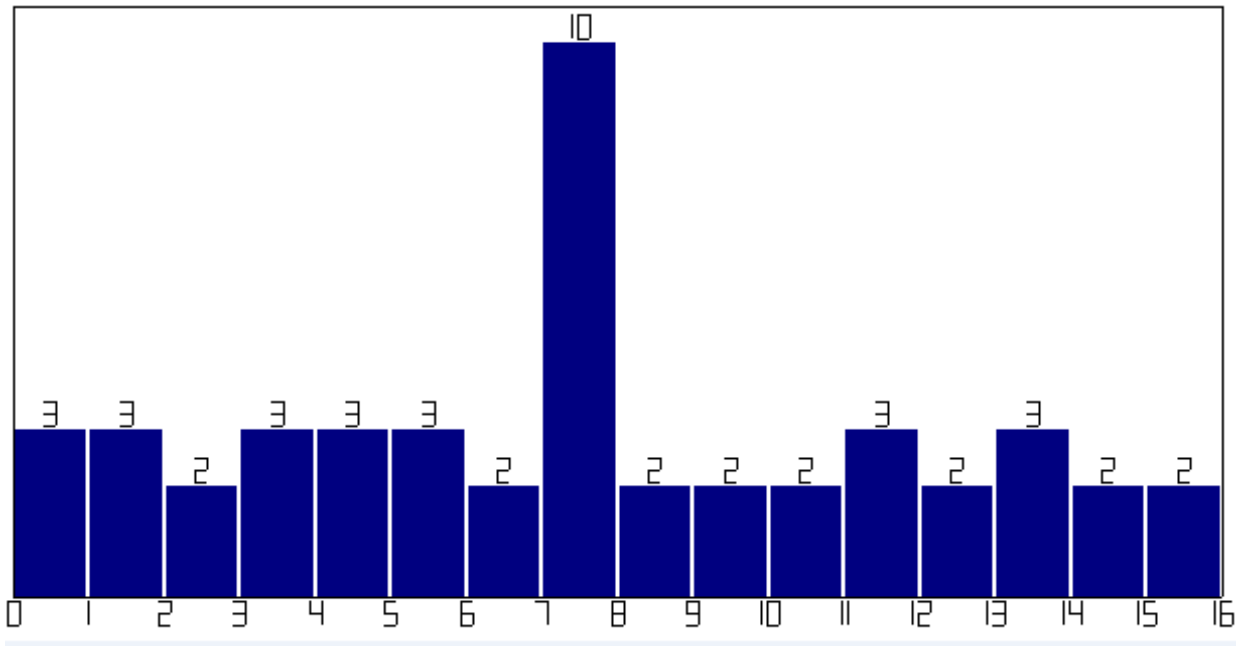


Рисунок 4.2 – Зайнятість оператора першої та другої зміни протягом 15-го дня
моделювання

Statistics for the work of operators during the month:

Operator rate of employment $Z = 0.09$.

Coefficient of queue $p = 0.16$.

Maximum operator's busy time $T_{\max} = 138.95$ min.

Maximum uninterrupted operator's busy time $T_{\text{umax}} = 8.24$ min.

Max queue length $K_{\max} = 6$.

The average queue length $K_{\text{sr}} = 5.27$.

Total time in queue = 1746.93 min.

Number of queue entries = 403.00 .

Average waiting time in queue = 4.33 min.

Average processing time of information by the operator = 1.02 min. .

Рисунок 4.3 – Результати моделювання діяльності оператора протягом місяця

Таким чином, запропонований метод дозволяє отримати всі необхідні для ергономічної оцінки операторської діяльності кількісні характеристики. Як видно з рис. 4.3, жоден з отриманих показників не перевищує оптимального або гранично допустимого (наведені в дужках) значень: $z = 0,09$ (не більше 0,75); $T_{\text{зан}} = 8,24$ хвилини (від 15 до 20 хвилин); $p = 0,16$ (не більше 0,4); $K_{\text{доп}} = 6$ (оптимальне значення 3, однак, як було зазначено вище, допускається максимальна довжина черги сигналів до 9). Це свідчить про правильний перерозподіл функцій між людиною і машиною на передпроектній стадії проектування ЛМІ SCADA-системи дозування, а також про відсутність інформаційних перевантажень оператора при інтерактивній взаємодії з даною версією ЛМІ. У зв'язку з цим, не потрібно внесення змін в ЛМІ або ТП.

4.3 Висновки до четвертого розділу

У даному розділі представлено вихідними дані для моделювання процесу виготовлення газобетону на ЗАТ «Мотор Січ». Проведено моделювання відповідної технологічної лінії протягом 30-ти днів з двозмінним восьмигодинним робочим днем.

Моделювання проводилося для виготовлення газобетону за трьома різними рецептами суміші.

Отримано як загальні результати моделювання, так і результати моделювання роботи кожної зміни впродовж робочого дня. Жоден з отриманих показників не перевищує оптимального або гранично допустимого значення, щодо діяльності оператора.

Це свідчить про правильний перерозподіл функцій між людиною і машиною на передпроектній стадії проектування ЛМІ АСУТП дозування, а також про відсутність інформаційних перевантажень оператора.

5 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

У дипломній кваліфікаційній роботі магістра досліджено та програмна реалізація методу оцінки діяльності оператора автоматизованої системи управління технологічним процесом.

Впровадження в експлуатацію розробленого програмного забезпечення (ПЗ) на виробничому підприємстві дозволить скоротити 0,75 ставки одного інженера з аналізу даних з заробітною платою 10 тис. гривень на місяць.

У розробці беруть участь програміст із заробітною платою 8000 грн. на місяць і консультант із заробітною платою 9000 грн. на місяць. На розробку проекту було витрачено два місяці. Розробка ПЗ розпочалася першого вересня і має бути виконано до кінця жовтня 2020 року.

5.1 Планування розробки програмного продукту

Весь комплекс розробки програмного продукту можна розділити на етапи. Для кожного етапу вказуються трудомісткість, кількість виконавців і тривалість робіт. Інформація з характеристики робіт по розробці програми зведена у табл. 5.1.

За даними табл. 5.1 складається графік планування розробки програмного продукту.

Таблиця 5.1 – Характеристика робіт по розробці програми

Найменування робіт	Трудомісткість		Виконавці	Тривалість, днів
	люд-дн	%		
1	2	3	4	5
Аналіз ПЗ	10	15,3	Програміст Консультант	5
Визначення вимог до ПП	10	15,3	Програміст Консультант	5

Продовження таблиці 5.1

1	2	3	4	5
Проектування структури програми	20	30,7	Програміст Консультант	10
Розробка схеми функціонування програми	5	7,69	Програміст	5
Створення програмного коду	10	15,3	Програміст	10
Тестування і налагодження програмного продукту	5	7,69	Програміст	5
Розробка програмної документації	5	7,69	Програміст	5
Разом	65	100		45

5.2 Визначення витрат на розробку програми

Для визначення витрат на розробку програми складають калькуляцію кошторисної вартості робіт, що включає статті:

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування на єдиний соціальний внесок (ЄСВ);
- матеріали та комплектуючі вироби;
- витрати на спеціальне обладнання (на оплату машинного часу ЕОМ для написання і налагодження програмних засобів);
- накладні витрати.

Витрати за статтею «Основна заробітна плата» складаються з планового фонду зарплати всіх категорій працівників, зайнятих в розробці програми [27]. Розрахунок зарплати на підставі даних про трудомісткості, представлений в табл. 5.2.

Таблиця 5.2 – Основна заробітна плата

Посада виконавця	Чисельність, чол	місячний оклад	Кількість місяців роботи	Сума ЗП, грн.
Програміст	1	8000	2	16000
Консультант	1	9000	1	9000
Разом	2			25000

Додаткову заробітну плату приймають рівною 10% від основної зарплати. Вона розраховується за формулою (5.1):

$$ЗП_{\text{дод}} = ЗП_{\text{осн}} \cdot 10\%. \quad (5.1)$$

Підставивши величину основної заробітної плати в дану формулу отримаємо: $ЗП_{\text{дод}} = 25000 \times 0.1 = 2500$ грн.

Відрахування на єдиний соціальний внесок (ЄСВ) визначають у відсотковому відношенні від сум основної та додаткової зарплати з урахуванням премій і доплат. Відрахування вираховуються за формулою (5.2):

$$ОТ = (ЗП_{\text{осн}} + ЗП_{\text{дод}}) \cdot 0,22. \quad (5.2)$$

Розраховуємо:

$$ОТ = (25000 + 2500) \cdot 0,22 = 6050 \text{ грн.}$$

Використовується два найменування матеріалів: картридж - 900 грн., папір - 100 грн. Витрати на матеріали розраховують за формулою (5.3):

$$M = \sum_{i=1}^n (\Pi_i \cdot N_i \cdot (1 + K_{\text{т.з}}) - \Pi_{\text{іо}} \cdot N_{\text{іо}}) \quad , \quad (5.3)$$

де M – витрати на матеріали, покупні напівфабрикати і комплектуючі вироби, грн.;

$K_{т.з}$ – коефіцієнт, що враховує транспортно-заготівельні витрати;

Π_i – ціна i -го найменування матеріалу, напівфабрикату і комплектуючого, грн.;

N_i – потреба в i -му матеріалі, напівфабрикаті та комплектуючому;

Π_{io} – ціна зворотних відходів i -го найменування матеріалу, грн;

N_{io} – кількість зворотних відходів i -го найменування;

n – кількість найменувань матеріалів, напівфабрикатів і комплектуючих.

Отримуємо:

$$\Pi_{io} = 0; N_{io} = 0; K_{т.з} = 0,05;$$

$$M = (1 + 0,05) \cdot (900 + 100) = 1050 \text{ грн.}$$

Разом витрати на матеріали становлять 1050 гривні.

У статтю «Витрати на спеціальне обладнання» входять витрати на придбання, транспортування, монтаж і налагодження нестандартного обладнання. Практично, в даному випадку, в цій статті враховуються витрати на оплату машинного часу ЕОМ для написання і налагодження програмних засобів. Для цього необхідно скласти кошторис «витрат на утримання і експлуатацію устаткування», виходячи з якого визначиться вартість одного машино-години роботи ПК, після множення якої на машинний час пішло на написання і налагодження програми отримаємо витрати на оплату машинного часу.

Амортизаційні відрахування визначають за формулою (5.4):

$$A = БВ \cdot \frac{H_a}{100}, \quad (5.4)$$

де БВ – балансова вартість обчислювальної техніки, грн.;

N_a – норма амортизаційних відрахувань на повне оновлення обчислювальної техніки, %.

Балансова вартість обчислювальної техніки становить 20000 грн. Тому амортизаційні відрахування становлять:

$$A = 20000 \cdot 0,25 = 5000 \text{ грн.},$$

Статтю "Експлуатація обладнання" розраховують підсумовуванням витрат на силову електроенергію і допоміжні матеріали. Витрати на електроенергію визначають за формулою (5.5):

$$C_3 = N_H \cdot \Phi_{\text{эф}} \cdot K_{\text{зв}} \cdot K_{\text{зм}} \cdot C_3, \quad (5.5)$$

де N_H – номінальна потужність ЕОМ, кВт;

$\Phi_{\text{эф}}$ – річний ефективний фонд часу роботи ЕОМ, машино–год;

$K_{\text{зв}}$ – середній коефіцієнт завантаження за часом;

$K_{\text{зм}}$ – коефіцієнт завантаження по потужності;

C_3 – ціна одного кВт · год електроенергії, грн / (кВт · год).

Номінальна потужність ЕОМ – 0,2 кВт. Річний ефективний фонд часу роботи ЕОМ ставити 1800 годин. Середні коефіцієнти завантаження за часом та за потужністю рівні відповідно 0,9 та 0,6. Ціна однієї кіловат–години електроенергії ставить 2,11 грн. Отримуємо:

$$C_e = 0,2 \cdot 1800 \cdot 0,9 \cdot 0,6 \cdot 2,11 = 410,18 \text{ грн.}$$

Приймаємо, що $C_e = 410$ грн.

Витрати за статтею "Заробітна плата обслуговуючих робітників та відрахування на соціальне страхування в інші фонди" визначають за формулою (5.6):

$$ЗП_{\text{обсл}} = ФЗП_{\Gamma} \cdot (1 + K_{\text{відрах.}}) \cdot \frac{t_{\text{обсл}}}{\Phi_{\text{еф.обсл}}} \cdot \frac{S_{\text{рп}}}{\Gamma_{\text{г}}} + E_{\text{зп}}, \quad (5.6)$$

де $ФЗП_{\Gamma}$ – річний фонд заробітної плати (основної та додаткової) обслуговуючих робітників, грн;

$K_{\text{відрах}}$ – коефіцієнт, що враховує відрахування на соціальне страхування і в інші фонди;

$t_{\text{обсл}}$ – час протягом року, необхідне технічне обслуговування ЕОМ, годин / рік. Приймаємо $t_{\text{обсл}} = 24$ год.;

$\Phi_{\text{еф.обсл}}$ – річний ефективний фонд часу обслуговуючого персоналу, годин / рік. Приймаємо, що річний ефективний фонд часу обслуговуючого персоналу становить 1800 годин

Заробітна плата обслуговуючого персоналу = 6000 грн/місяць.

$$ФЗП_{\Gamma} = 6000 \cdot 12 = 72000 \text{ грн.}$$

$$ЗП_{\text{обсл}} = 72000 \cdot (1 + 0,22) \cdot \frac{24}{1800} = 1171 \text{ грн.}$$

Суму витрат за статтею "Поточний ремонт обладнання" обчислимо як 3% від балансової вартості обладнання, оскільки неможливо заздалегідь визначити, скільки коштів необхідно витратити на придбання запасних частин наявного обладнання. Балансова вартість обчислювальної техніки становить 20000 грн. Тому сума витрат за статтею "Поточний ремонт обладнання" становить: $20000 \cdot 0,03 = 600$ грн.

Інших витрат за проєктом не передбачено, тому сума витрат за цією статтею дорівнює нуль гривень.

Кошторис витрат на утримання і експлуатацію устаткування наведений в табл. 5.3.

Таблиця 5.3 – Кошторис витрат на утримання і експлуатацію устаткування

Найменування статей витрат	Сума, грн.
Амортизація обладнання	5000
Заробітна плата основна і додаткова обслуговуючих робітників з відрахуваннями на соціальні заходи	1171
Експлуатація обладнання (крім витрат на поточний ремонт)	410
Поточний ремонт обладнання	600
Інші витрати	0
Разом	7181

Витрати на оплату машинного часу ЕОМ для написання і налагодження програмних засобів визначаються за формулою (5.7):

$$C_{\text{мо}} = P_{\text{екс}} \cdot t_{\text{мо}}, \quad (5.7)$$

де $C_{\text{мо}}$ – витрати на оплату машинного часу, грн;

$P_{\text{екс}}$ – експлуатаційні витрати на одну годину машинного часу цієї цифрової ЕОМ, грн. / машино–год.;

$t_{\text{мо}}$ – машинний час цифрової ЕОМ для написання і налагодження даного програмного продукту, машино–год.

$$P_{\text{екс}} = \frac{7181}{1800} = 3,99 \text{ грн. / машино–год.}$$

Для написання і налагодження даного програмного продукту ЕОМ експлуатується протягом 45 днів в одну зміну по 8 годин, що складає в сумі 360 годин. Отримуємо:

$$C_{\text{мо}} = 3,99 \cdot 360 = 1436 \text{ грн.}$$

До накладних витрат відносяться витрати на загальне управління і загальногосподарські потреби (заробітна плата апарату управління, канцелярські витрати і т.п.), утримання та експлуатацію будівель. Накладні витрати включаються до вартості розробки програми непрямым шляхом – у відсотках до основної заробітної плати розробників. Для обчислення цієї статті видатків у дипломній кваліфікаційній роботі приймемо, що накладні видатки становлять 50% від основної заробітної плати розробників, що складає 25000 грн. Тому сума накладних видатків становить: $25000 \times 0,5 = 12500$ грн.

Калькуляція кошторисної вартості робіт з розробки програми наведена в табл. 5.4.

Таблиця 5.4 – Калькуляція кошторисної вартості робіт з розробки програми

Найменування статей витрат	Сума, грн.
Основна заробітна плата	25000
Додаткова заробітна плата	2500
ЄСВ	6050
Матеріали і комплектуючі	1050
Витрати на спеціальне обладнання	1436
Накладні витрати	12500
Разом	48536

5.3 Розрахунок економічної ефективності програмного продукту

Річну економію від впровадження програми, що дозволяє знизити витрати машинного часу ЕОМ для рішення певного завдання в порівнянні із програмою, що застосовувалася раніше, визначають за формулою:

$$E = (T_{M_1} - T_{M_2}) \cdot B_{\text{екс.КТЗ}} + E_{\text{н}} \cdot \frac{\Phi_{\text{б.КТЗ}} (T_{M_1} - T_{M_2})}{\Phi_{\text{еф.КТЗ}}} - \frac{S_{\text{рп}}}{T_{\text{с}}} + E_{\text{зп}}, \quad (6.8)$$

де $T_{M_1} - T_{M_2}$ – машинний час, необхідний для розв’язання поставлених завдань, відповідно в старому та у новому варіантах, машино–год./рік;

$B_{\text{екс}}$ – експлуатаційні витрати, що доводяться на 1 год машинного часу ЕОМ, грн / машино–год.;

$E_{\text{н}}$ – нормативний коефіцієнт ефективності капітальних вкладень у засоби автоматизації;

$\Phi_{\text{б.КТЗ}}$ – балансова вартість ЕОМ (КТЗ), грн.;

$\Phi_{\text{еф.КТЗ}}$ – річний ефективний фонд часу роботи ЕОМ, машино–год.;

$S_{\text{рп}}$ – сумарні витрати на розробку програми, грн.;

$T_{\text{с}}$ – термін служби впроваджуваної програми до її морального зношування, років. Приймаємо цей термін 5 років;

$E_{\text{зп}}$ – економія на заробітній платі.

До впровадження даного ПЗ дані функції виконувались інженером з аналізу даних 1800 годин на рік, а після 450 годин на рік (економія складає 0,75 ставки одного технолога з заробітною платою 10 тис. гривень на місяць). Далі зазначені розрахунки:

$$E = (1800 - 450) \cdot 3,99 + \frac{0,2 \cdot (20000 \cdot (1800 - 450))}{1800} - \frac{48536}{5} + 10000 \cdot 0,75(1 + 0,22) \cdot 12 = 120479 \text{ грн.}$$

Коефіцієнт ефективності капітальних вкладень E и строк їхньої окупності $T_{ок}$ розраховуються за формулами (5.9) і (5.10):

$$E_{\phi} = \frac{E}{S_{\phi\pi}}, \quad (5.9)$$

$$T_{ок} = \frac{S_{\phi\pi}}{E}, \quad (5.10)$$

та складають:

$$E_{\phi} = \frac{120479}{48536} = 2,48,$$

$$T_{ок} = \frac{48536}{120479} = 0,4 \text{ року.}$$

Зведемо отримані техніко–економічні показники в табл. 5.5.

Таблиця 5.5 – Техніко–економічні показники

Показники	Сума	Одиниця виміру
Витрати на розробку програми	48608	грн.
Річна економія	120735	грн.
Строк окупності програми	0,4	рік

Таким чином, впровадження в експлуатацію розробленого програмного забезпечення на виробничому підприємстві дозволить скоротити 0,75 ставки одного інженера з аналізу даних з заробітною платою 10 тис. гривень на місяць. Економічні розрахунки показали, що строк окупності програми 0,4 року, що значно менше часу її нормативного строку окупності (5 років). Крім того, річна

економія при використанні програми становить 120479 грн. Виходячи з виконаних розрахунків, можна зробити висновок, що розробка даного програмного продукту є економічно доцільною.

6 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

6.1 Аналіз потенційних небезпек

Стан здоров'я користувача комп'ютера багато в чому визначається характером роботи, який відрізняється:

- підвищеним навантаженням на зоровий аналізатор – тривала робота з об'єктами малого розміру;
- інтелектуальним навантаженням – необхідність швидкого прийняття рішень, творча діяльність, постійне сприйняття і оцінка нової інформації, високий ступінь складності завдання;
- емоційним навантаженням – ступінь відповідальності за завдання для виконання, дефіцит часу, значимість помилки;
- монотонністю трудового процесу – багаторазове повторення одноманітних дій, тривалість зосередженого спостереження;
- гіподинамією – тривалим перебуванням оператора в одній позі без активних рухів.

Статистичні дані показують, що основними і найбільш часто виникаючими захворюваннями при роботі з комп'ютером є запалення очей, глаукома, катаракта, головний біль, хвороби серцево-судинної системи, біль у хребті, різь в очах, біль при русі рук, свербіж шкіри, дерматити, лущення, нервово-психічні розлади, болі в області м'язів обличчя і шиї.

Для створення сприятливих умов роботи з комп'ютером необхідна відповідність нормативно-правовим актам з охорони праці, які регламентують показники стану навколишнього середовища:

- достатню площу і об'єм приміщення, де встановлено комп'ютер;
- показники мікроклімату робочого приміщення (температура повітря, відносна вологість, та ін.);
- параметри світлового середовища робочого місця і робочої поверхні (наявність природного і штучного освітлення, освітленість екрана монітора,

освітленість горизонтальної поверхні столу, відсутність прямого і відбитого сліпучого відблиску джерел світла, відсутність пульсації освітленості);

- рівень шуму і вібрації робочих місць в межах норм;
- відсутність в повітрі робочої зони шкідливих речовин.

Нижче розглядаються заходи з охорони праці, виробничої санітарії, гігієни праці та пожежної безпеки при дослідженні, розробці та програмній реалізації методів оцінки діяльності оператора автоматизованої системи управління технологічним процесом (газобетонного виробництва).

6.2 Заходи із забезпечення безпеки

Забезпечення безпеки праці, організації оптимальних умов праці, а також захисту навколишнього середовища в умовах роботи з ПК досягається шляхом дотримання норм і проведення заходів, передбачених законодавчими актами України.

ПК підключаються до мережі з напругою 220 В і частотою 50Гц. Згідно ПУЕ приміщення відноситься до 1-ї категорії «Приміщення без підвищеної небезпеки» [28].

У приміщенні відсутні відкриті струмопровідні частини з напругою понад 12В. При виконанні правил ПУЕ ураження електричним струмом можливо тільки в разі несправності апаратури і живлячих кабелів.

Апаратура, яка перебуває під небезпечною для життя напругою, передбачає забезпечення електробезпеки. Всі металеві частини обчислювальної техніки заземлені на третій провід спеціальних триконтактних розеток, призначених для підключення ПК. Струмопровідні кабелі розташовано під дерев'яною підлогою, зверху покритим лінолеумом або у вигляді прихованої проводки в стінах, таким чином, контакт з кабелем виключається. Над кожною розеткою розміщено попереджувальний напис. Індикаторні лампи в блоках ЕОМ сигналізують про включення блоку в мережу [29].

Електрична ізоляція забезпечується конструкціями пристроїв і способами їх підключення. В процесі експлуатації періодично проводиться перевірка якості ізоляції, шляхом вимірювання її опору. Відповідно до ПУЕ [28] в електроустановках до 1000 В мінімальне значення опору ізоляції силових і освітлювальних електропроводок на ділянці між суміжними запобіжниками або за останніми запобіжниками між будь-яким проводом і землею становить не менше 0,5 МОм.

Відповідно до ПУЕ захисне заземлення застосовується в усіх електроустановках змінного струму під напругою 380В і вище і постійного струму 440В і вище. Захисному заземленню підлягають корпуси трансформаторів, оболонки кабелів, металеві огорожі електроустановок [29].

Вирівнювання потенціалів досягаються шляхом влаштування контурних заземлювачів. У приміщеннях ОЦ вирівнювання потенціалів здійснюється з'єднанням металевих стійок ЕОМ, корпусів апаратів, світильників та іншого обладнання з усіма доступними для дотику металевими конструкціями будівлі, споруди, а також з контуром-шиною захисного заземлення.

Електричний поділ мережі передбачає поділ мережі на окремі електрично не зв'язані між собою ділянки за допомогою розділових трансформаторів. Підвищення електробезпеки відбувається за рахунок зменшення ємності мережі та підвищення опору ізоляції фаз мережі щодо землі.

Зменшення напруги електроустановки істотно підвищує умови електробезпеки. Так при напрузі 10-12 В і електричному опорі тіла людини 5-6 кОм сила струму, що протікає через людину, не перевищить 2 мА.

Подвійна ізоляція забезпечує застосування крім основної, робочої ізоляції струмоведучих частин ще одного шару додаткової ізоляції, ізолюючої людину від металевих неструмоведучих частин, які можуть випадково виявитися під напругою. При застосуванні електроустаткування з подвійною ізоляцією не потрібно ні заземлення, ні занулення їх корпусів.

У ОЦ розрядні струми статичної електрики частіше за все виникають при дотику до будь-якого з елементів ЕОМ. Такі розряди небезпеки для людини не

представляють, але крім неприємних відчуттів вони можуть привести до виходу з ладу ЕОМ. Для зниження величини виникаючих зарядів статичної електрики в ОЦ покриття виконані з одношарового полівінілхлоридного антистатичного лінолеуму. Іншим методом захисту є нейтралізація заряду статичної електрики іонізованим газом. До загальних заходів захисту від статичної електрики в ОЦ можна віднести загальне та місцеве зволоження повітря [30].

6.3 Заходи з виробничої санітарії та гігієни праці

Для зниження шкідливого впливу комп'ютера на людину повинні дотримуватися певні вимоги до умов роботи, до робочого місця і крім того, повинен дотримуватися режим роботи за комп'ютером.

Об'ємно-планувальні рішення будівель та приміщень для роботи з візуальними дисплейними терміналами (ВДТ) ЕОМ повинні відповідати вимогам ДСанПіН 3.3.2.007-98 «Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [31]. Розміщення робочих місць з ВДТ ЕОМ в підвальних приміщеннях, на цокольних поверхах заборонено.

В обчислювальному центрі при дослідженні та програмній реалізації методів автоматичного визначення заголовків новин забезпечено виконання норми ДБН В.2.5-28-2018 «Природне и штучне освітлення» [32]. Відносно штучного освітлення: освітленість робочих місць в горизонтальній площині на рівні 0,8 м від підлоги не менше 400 лк, вертикальна освітленість в площині екранів дисплея – не менше 300 лк, коефіцієнт пульсації освітленості – не більше 10%.

Розрахуємо систему загального рівномірного штучного освітлення в приміщенні.

Приймаємо розміри приміщення довжина $A = 6$ м, ширина $B = 4$ м, висота $H = 4$ м.

Висота робочої поверхні складає $h = 0,7$ м.

Згідно з ДБН В.2.5-28-2018 «Природне і штучне освітлення. Норми проектування» освітлення нормуються як $E_n = 200$ Лк.

Розряд зорової роботи – III г.

Коефіцієнти відбиття поверхонь приміщення (від: стелі – ρ_c ; стін – $\rho_{ст}$; підлоги – $\rho_{п}$) у світлому приміщенні $\rho_c=70\%$; $\rho_{ст}=50\%$; $\rho_{п}=30\%$;

Обираємо світильник типу ЛВП (IP=54 – ступінь захисту світильника, $[L/h] = 1,3$ – числове значення коефіцієнта світильника).

Коефіцієнт запасу k_z враховує зниження рівня освітленості з часом в результаті забруднення та старіння ламп, світильників і поверхонь приміщення, приймається в залежності від виробничих умов. Приймаємо $k_z = 1,4$.

Коефіцієнт нерівномірності (мінімального) освітлення z (відношення середньої освітленості до мінімальної освітленості), для люмінесцентних ламп низького тиску, як правило дорівнює $z = 1,1$.

Розрахуємо кількість рядів світильників у приміщенні за формулою (6.1):

$$N_p = \frac{B}{(H - h_p) \left[\frac{L}{h} \right] + K_{т.з}) - \Psi_{io} \cdot N_{io}} , \quad (6.1)$$

та отримаємо:

$$N_p = \frac{4}{(4-0,7) \cdot 1,3} = 0,93 \approx 1.$$

Визначимо максимально припустиму відстань між рядами світильників за формулою (6.2):

$$L_{\max} = \frac{B}{N_p}, \quad (6.2)$$

де B – ширина приміщення,

N_p – кількість рядів, розрахунок зазначено нижче:

$$L_{max} = \frac{4}{1} = 4 \text{ м} ,$$

Розраховуємо висоту підвісу світильника над робочою поверхнею за формулою (6.3):

$$h = \frac{L_{max}}{\left[\frac{L}{h}\right]}, \quad (6.3)$$

та отримуємо результат:

$$h = \frac{4}{1,3} = 3,07 \text{ м} .$$

Знайдемо висоту звисання світильника від стелі за формулою (6.4):

$$h_3 = H - h_p - h , \quad (6.4)$$

де H – висота приміщення, h_p – висота робочої поверхні, h – висота підвісу світильника над робочою поверхнею.

$$h_3 = 4 - 0,7 - 3,07 = 0,23 \text{ м} .$$

Чисельне значення індексу приміщення визначаємо за рівнянням (6.5):

$$i = \frac{AB}{h(A+B)} , \quad (6.5)$$

де A – довжина приміщення, м;

B – ширина приміщення, м;

h – висота розміщення світильників над робочою поверхнею, м.

Розрахуємо значення індексу:

$$i = \frac{6 \cdot 4}{3,07 \cdot (6+4)} = \frac{24}{30,7} = 0,78 .$$

Значення коефіцієнта використання світлового потоку η вибирається в залежності від виду джерела світла, типу обраного світильника, коефіцієнтів відбиття поверхонь приміщення (ρ_c , $\rho_{ст}$, $\rho_{п}$) та індексу приміщення. В нашому випадку $\eta = 0,38$.

Визначаємо сумарний світловий потік освітлювальної установки у даному виробничому приміщенні за формулою (6.6):

$$\Phi_{\Sigma} = \frac{E_n \cdot S \cdot k_3 \cdot z}{\eta}, \quad (6.6)$$

де Φ_{Σ} – розрахункове значення сумарного світлового потоку у приміщенні, лм;

E_n – нормоване значення освітленості, лк;

S – площа освітлюваної поверхні, м²;

k_3 – коефіцієнт запасу;

z – коефіцієнт нерівномірності (мінімальної) освітленості;

η – коефіцієнт використання світлового потоку.

Знайдемо розрахункове значення сумарного світлового потоку:

$$\Phi_{\Sigma} = \frac{200 \cdot 6 \cdot 4 \cdot 1,4 \cdot 1,1}{0,38} = \frac{7392}{0,38} = 19452 \text{ лм} .$$

Визначаємо умовну загальну кількість світильників у приміщенні за формулою (6.7):

$$N^* = A \frac{AB}{L^2} \max. \quad (6.7)$$

Розрахуємо загальну кількість світильників:

$$N^* = \frac{6 \cdot 4}{16} = 1,5 \approx 2 \text{ шт.}$$

Розрахуємо світловий потік умовного джерела світла за формулою (6.8):

$$\Phi_{\text{л}}^* = \frac{\Phi_{\Sigma}}{N_{\text{л}}}, \text{ лм}, \quad (6.8)$$

де Φ_{Σ} – розрахункове значення сумарного світлового потоку у приміщенні, лм;

$N_{\text{л}}$ – загальна кількість ламп у приміщенні, шт, та зазначено формулою (6.9);

$$N_{\text{л}} = N \cdot n, \quad (6.9)$$

де n – кількість ламп у світильнику (в ЛВП світильнику знаходиться 2 лампи)

Розрахуємо загальну кількість ламп у приміщенні:

$$N_{\text{л}} = 2 \cdot 2 = 4 \text{ шт.}$$

Розрахуємо світловий потік умовного джерела світла:

$$\Phi_{\text{л}} = \frac{19452}{4} = 4863 \text{ лм.}$$

Обираємо тип стандартної лампи з найближчим значенням фактичного світлового потоку лампи $\Phi_{\text{л}}$ і знайти коефіцієнт m (співвідношення між

розрахунковим світловим потоком лампи $\Phi_{\text{л}}^*$ та фактичним світловим потоком вибраної стандартної лампи $\Phi_{\text{л}}$):

Обираємо лампу ЛБ65, $\Phi_{\text{л}} = 4800$ лм, та зазначимо формулу для підрахунку коефіцієнта (6.10):

$$m = \frac{\Phi_{\text{л}}^*}{\Phi_{\text{л}}} , \quad (6.10)$$

та розрахуємо:

$$m = \frac{4863}{4800} = 1,01 .$$

Визначаємо оптимальну (фактичну) кількість світильників у приміщенні за формулою (6.11):

$$N = N \cdot m . \quad (6.11)$$

Підрахуємо оптимальну кількість світильників:

$$N = 2 \cdot 1,01 = 2 \text{ шт.}$$

Фактична кількість ламп у приміщенні розраховується за формулою (6.12):

$$N_{\text{фл}} = N_{\text{ф}} \cdot n , \quad (6.12)$$

Підрахуємо кількість ламп у приміщенні:

$$N_{\text{фл}} = 2 \cdot 2 \text{ шт.}$$

Визначаємо загальну розрахункову освітленість E_p у приміщенні, що створюється при застосуванні стандартних ламп за формулою (6.13):

$$E_p = \frac{\Phi_{\text{л}} N_{\text{фл}} \eta}{Sk_z} \text{ лк} . \quad (6.13)$$

Підрахуємо загальну розрахункову освітленість:

$$E_p = \frac{4800 \cdot 4 \cdot 0,38}{6 \cdot 4 \cdot 1,4 \cdot 1,1} = \frac{7296}{36,96} = 197,4 \text{ лк} .$$

Звукоізоляція огорожувальних конструкцій приміщення з ВДТ забезпечує параметри шуму, що відповідають вимогам ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку» [33].

Нормовані параметри мікроклімату, іонного складу повітря, вмісту шкідливих речовин відповідають вимогам ДСН 3.36.042-99 «Санітарні норми мікроклімату виробничих приміщень» [34].

Віконні отвори приміщень для роботи з ВДТ обладнані регульованими пристроями. Для внутрішнього оздоблення приміщень з ВДТ використовують дифузно-відбивні матеріали з коефіцієнтами відбиття для стелі 0,7-0,8, для стін 0,5-0,6.

Покриття підлоги матове з коефіцієнтом відбиття 0,3-0,5. Поверхня підлоги рівна, неслизька, з антистатичними властивостями. В обробці інтер'єру приміщень ВДТ не застосовуються полімерні матеріали, що виділяють у повітря шкідливі хімічні речовини. Полімерні матеріали для внутрішнього оздоблення приміщень з ВДТ використовуються з наявністю дозволу органів і установ державної санітарно-епідеміологічної служби.

Приміщення розташоване на другому поверсі виробничого корпусу, що допустимо і відповідає перерахованим вище правилам і нормам.

Устаткування і організація робочого місця працюючих з ВДТ ЕОМ відповідає ергономічним вимогам з урахуванням характеру і особливостей трудової діяльності.

В НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [35] передбачені гігієнічні вимоги до організації обладнання робочих місць ВДТ ЕОМ і ПЕОМ. Конструкція робочого столу повинна відповідати сучасним вимогам ергономіки і забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплея, клавіатури, принтера) і документів. Висота робочої поверхні робочого столу з ВДТ повинні регулюватися в межах 680 ... 800 мм, а ширина і глибина – забезпечувати можливість виконання операцій в зоні досяжності моторного поля (рекомендовані розміри: 600 ... 1400 мм, глибина – 800..1000 мм). Приймаємо робочу поверхню з розмірами 1400 × 800. Щоб уникнути відблисків на екрані вікон доцільно розмістити робоче місце торцем до вікна не ближче ніж на 1 м. Площа дозволяє розмістити додаткові меблі для зберігання документів.

Для підвищення продуктивності праці, а також зниження рівня стомлюваності при роботі з ЕОМ передбачені десятихвилинні перерви через кожну годину [29]

Велике значення має задоволення обстановки ергономічним і естетичним вимогам. Цього домоглися за допомогою колірної рішення приміщення і обладнання обчислювального центру [36].

При організації роботи для збереження здоров'я, запобігання професійних захворювань і підтримки працездатності передбачені перерви для відпочинку. Тривалість обідньої перерви визначається чинним законодавством про працю і Правилами внутрішнього трудового розпорядку підприємства.

За характером трудової діяльності робота у відділі відноситься згідно з чинним класифікатором професій до розробників програм (інженерів-програмістів), які виконують роботу з відео терміналом і документацією, при необхідності, і з інтенсивним обміном інформацією з ЕОМ і значним ступенем прийняття рішень. Робота характеризується інтенсивною розумовою творчою працею з підвищеним напруженням зору, концентрацією уваги на фоні нервово-

емоційного напруження, вимушеною робочою позою, загальною гіподинамією, періодичної навантаженням на кисті верхніх кінцівок:

Встановлюються такі режими праці і відпочинку при роботі з ЕОМ при 8-годинній денній робочій зміні:

- для розробників програм із застосуванням ЕОМ призначений регламентований перерву для відпочинку тривалістю 15 хвилин через кожен годину роботи на ВДТ.

Всі перелічені вище заходи проведені з метою поліпшення умов праці програміста, і, отже, сприяє підвищенню ефективності праці [30].

6.4. Заходи безпеки у надзвичайних ситуаціях

6.4.1. Заходи з пожежної безпеки

Заходи з протипожежної безпеки розроблені для виробничого приміщення обладнаного ПК з ВДТ розроблені згідно вимог НАПБ А.01.001-2014 «Правила пожежної безпеки в Україні» [37].

Виходячи з аналізу речовин та матеріалів, які використовуються при роботі у приміщенні, обладнаному ПК з ВДТ:

- згідно ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги» [38] у приміщенні, обладнаному ПК з ВДТ можлива пожежа класу – А та Е; воно належить до категорії «Д» з пожежної небезпеки – простір у приміщенні, у якому перебувають тверді горючі речовини та матеріали.

Оскільки приміщення належить до категорії «Д» з пожежної небезпеки, тому відповідно до вимог ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги» [38] воно має II ступінь вогнестійкості.

У разі виникнення пожежі у приміщенні обладнаному ПК з ВДТ для евакуації персоналу відповідно до вимог ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги» передбачені виходи, по обидві сторони приміщення, з одного боку вікно (на пожежні сходи), а з іншого – вхідні двері. Згідно СНиП 2.09.02-85* «Производственные здания», відстань від найбільш

віддаленого робочого місця до найближчого евакуаційного виходу не обмежується.

Обладнання, силові та освітленні мережі приміщення, обладнаного ПК з ВДТ відповідають вимогам пожежної безпеки, оскільки виконані відповідно до вимог НПАОП 40.1-1.32-01 «Правила будови електроустановок. Електрообладнання спеціальних установок» [39], та мають ступінь захисту ізоляції обладнання IP44 яка відповідає класу пожежанебезпечної зони П-Па до якої належить приміщення.

Для виявлення початкової стадії загоряння, швидкого й точного оповіщення служби пожежної охорони про час і місце виникнення пожежі встановлені системи автоматичної пожежної сигналізації (АПС). Вони можуть самостійно пускати в хід установки пожежогасіння, коли пожежа не досягла великих розмірів. Системи АПС складаються з пожежних сповіщувачів, ліній зв'язку й прийомних пультів (станцій). Згідно вимог ДБН В.2.5-56:2014 «Системи протипожежного захисту», в приміщенні обладнаному ПК з ВДТ встановлена система пожежної й охоронної сигналізації «Фотон-ОФІС», яка забезпечує виявлення теплових і димових ознак пожежі і місця виникнення пожежі з точністю до місця розміщення датчика [29].

Оскільки приміщення що обладнане ПК з ВДТ має площу 24 м², тому відповідно до вимог п. 3.8 розділу «Типові норми належності вогнегасників» ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги» [40] для гасіння електроустановок, що знаходяться під напругою, передбачені вуглекислотні вогнегасники типу ВВК-3,5 у кількості 2 штуки (з розрахунку один вогнегасник с величиною заряду вогнегасної речовини 3 кг. і більше, на 20 м² площі приміщення). Відстань між вогнегасниками та місцями можливих загорянь не перевищує 10 м.

6.4.2. Заходи з цивільного захисту

Забезпечення сталої роботи комунально-енергетичних систем промислового об'єкта.

Стійкість роботи об'єкта - здатність виконувати задані функції не тільки в нормальних, а й в надзвичайних ситуаціях, попереджати виникнення аварій і катастроф на об'єкті. Зокрема, об'єкти виробничої сфери повинні випускати продукцію в необхідному обсязі, номенклатурі, заданої якості та вартості, що забезпечує конкурентоспроможність на ринку. Стійка робота об'єкта неможлива без урахування стійкості самого об'єкта.

Під стійкістю об'єкта розуміють здатність його інженерно-технічного комплексу (будівель, споруд, обладнання, інженерних, комунально-енергетичних, транспортних та інших систем і комунікацій) протистояти руйнівній дії джерел надзвичайних ситуацій (НС) .

До комунально - енергетичних систем відносяться:

- водопостачання;
- теплопостачання;
- електропостачання;
- газопостачання;
- каналізація.

Усі вище перераховані системи інженерно-технічного комплексу не тільки повинні бути стійкі при впливі на них природних і техногенних НС, але і не повинні стати джерелами НС.

Забезпечення стійкості роботи систем водопостачання промислового об'єкта досягається:

- виконанням норм інженерно-технічних заходів при будівництві;
- проведенням заходів щодо підвищення фізичної стійкості, перш за все трансформаторних підстанцій, насосних станцій, очисних споруд і трубопроводів;

- своєчасним ремонтом і ревізією окремих ділянок систем водопостачання;
- використанням декількох незалежних джерел води;
- можливістю подачі води з одного водопроводу в інший;
- наявністю резервних джерел води;
- застосуванням автоматичних систем сигналізації при аваріях і автоматичних засувках;
- кільцюванням водопровідної мережі в межах міста для обходу пошкоджених ділянок водопроводу;
- високоефективної і надійної системою очисних споруд.

Забезпечення стійкості роботи системи тепlopостачання промислового об'єкта досягається:

- виконанням норм інженерно-технічних заходів при будівництві;
- проведенням заходів щодо підвищення фізичної стійкості, перш за все котелень, насосних станцій, теплотрас і теплопроводів;
- прокладкою під землею газопроводів теплотрас;
- надійною роботою систем аварійного відключення ділянок теплотрас при аварійних ситуаціях;
- підвищенням надійності теплопроводів за рахунок кільцювання в межах міст і районів, де знаходяться об'єкти споживання;
- проведенням своєчасного профілактичного ремонту від корозії і засмічення [41].

Забезпечення стійкості роботи системи електропостачання промислового об'єкта досягається:

- виконанням норм інженерно-технічних заходів при будівництві;
- наявністю систем автоматики, аварійної сигналізації та захисту;
- проведення додаткових заходів по підвищенню стійкості наземних споруд (станції, підстанції, розподільчі пункти, трансформаторні станції тощо) та повітряних ліній електропередач;
- кільцюванням розподільчої мережі;

- дублюванням або резервування мереж електропостачання для об'єктів, аварії на яких в випадках відключення електропостачання можуть викликати надзвичайні ситуації;

- прокладкою електричних кабелів під землею в межах міст;
- заборона земляних робіт в містах без дозволу енергонагляду;
- своєчасним профілактичним ремонтом обладнання та його модернізацією;

- високим рівнем професійної підготовки обслуговуючого персоналу, його здатністю запобігати і діяти в питання надзвичайних ситуацій.

Забезпечення стійкості роботи систем газопостачання промислового об'єкта досягається:

- виконанням норм інженерно-технічних заходів при будівництві;
- проведенням додаткових заходів щодо підвищення фізичної стійкості будівель газорозподільних станцій і пунктів;

- прокладкою під землею газопроводів високого та середнього тиску;
- надійною роботою систем аварійного відключення ділянок газопроводу при аварійних ситуаціях;

- підвищенням надійності газопостачання об'єктів за рахунок кільцювання газопроводів в межах міст;

- подачею газу споживачеві не менше ніж через дві газорозподільні станції;

- впровадженням в диспетчерське управління і обслуговування газового господарства телемеханічних пристроїв і автоматики;

- високим рівнем професійної підготовки обслуговуючого персоналу, здатного запобігати або локалізувати аварійні ситуації.

Забезпечення стійкості роботи системи каналізації промислового об'єкта досягається:

- виконанням норм інженерно-технічних заходів при будівництві;
- своєчасним очищенням колекторів та інших ділянок при закупорці;

- роздільної системою каналізації за умови, що колектори обох частин системи з'єднані між собою, що дає можливість відключати окремі пошкоджені ділянки;
- узгодженням з органами санітарного нагляду місць скидання стічних вод;
- забезпеченням надійної роботи станцій перекачування;
- проведенням своєчасного профілактичного ремонту [42].

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи магістра було виконано наступні задачі:

- досліджено процеси управління та виготовлення газобетону, та моделі, що їх описують;
- досліджено структуру і процес функціонування СІМ;
- проведено порівняльний аналіз відомих СІМ, та обрано СІМ, необхідну для вирішення поставлених задач;
- розглянуто стан досліджень у області кількісної оцінки діяльності оператора;
- розроблено комплексну математичну модель, що формалізує діяльність як людини-оператора, так і фізичні процеси, що відбуваються в ході ТП;
- практично реалізовано комплексну математичну модель у вигляді ІМ процесу управління дозуванням матеріалів газобетону на ЗАТ «Мотор Січ»;
- проведено імітаційний експеримент та досліджено кількісні характеристики діяльності оператора технологічної лінії виробництва газобетону на ЗАТ «Мотор Січ».

Наукова новизна результатів дослідження полягає у тому, що дістала подальшого розвитку відома комбінована (аналітико-імітаційна) модель процесу автоматизованого управління ТП приготування газобетону шляхом розширення формального опису процесу функціонування оператора для різних видів неполадок в ТП.

Практична цінність роботи полягає в тому, що на підставі досліджень розроблено ІМ, експерименти з якою забезпечують вирішення задачі підвищення ефективності та якості діяльності оператора АСУТП.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Каминская, Ж.К. Интеллектуальная поддержка эргономического проектирования интерфейсов АСУ ТП / Ж.К. Каминская, С.Н. Сердюк // Информационные технологии. Проблемы и решения: Международная научно-практическая конференция, редкол.: Ф.У. Еникеев и др. – Уфа: Восточная печать, 2014. – С.164–169.
2. Сердюк, С.М. Ергономічні питання побудови людино-машинних систем: навчальний посібник / С.М. Сердюк. – Запоріжжя: ЗНТУ, 2014. – 334 с.
3. Сажнев, Н.П. Производство ячеистобетонных изделий. Теория и практика / [Н.П. Сажнев, В.Н. Гончарик, Г.С. Гарнашевич и др.]. – Минск: НПООО «Стринко», 2004. – С. 4-7.
4. Beitzel, H. Quality-assured manufacture of self-compacted concrete by changing the relevant machine parameters in concrete mixers / H. Beitzel // BFT. – 2007. – № 1. – P. 32–45.
5. Кулинич, Э.М. Повышение эффективности автоматизации процессов управления технологическими комплексами приготовления газобетона: дис. ... канд. техн. наук: 05.13.07 «Автоматизация процессов управления» / Э.М. Кулинич. – Запорожье: ЗНТУ, 2013. – 157 с.
6. Душков, Б.А. Основы инженерной психологии. Учебник для студентов вузов / Б.А. Душков, А.В. Королев, Б.А. Смирнов. – М.: Академический проект, 2002. – 576 с.
7. Максимей, И.В. Имитационное моделирование на ЭВМ / И.В. Максимей. – М.: Радио и связь, 1988. – 232 с.
8. Зинченко, В.П. Основы эргономики / В.П. Зинченко, В.М. Мунипов. – М.: МГУ, 1979. – 344 с.
9. Информационно-управляющие человеко-машинные системы: исследование, проектирование, испытание. Справочник / А.Н. Адаменко, А.Т. Ашеро́в, И.Л. Бердников и др.; под общ. ред. А. И. Губинского. – М.: Машиностроение, 1993. – 528 с.

10. Каминская, Ж.К. Метод оценки деятельности оператора автоматизированной системы управления технологическим процессом газобетонного производства / Ж. К. Каминская, С.Н. Сердюк, Э.М. Кулинич // Радіоелектроніка, інформатика, управління. – Запоріжжя: ЗНТУ, 2019. – № 2. – С.177 – 188

11. Плотников, А.М. Анализ современного состояния и тенденции развития имитационного моделирования в Российской Федерации / [А.М. Плотников, Ю.И. Рыжиков, Б.В. Соколов и др.] // Труды СПИИРАН. – 2013. – Вып. 2(25). – С. 42–112.

12. Карпов, Ю.Г. Имитационное моделирование систем: введение в моделирование с помощью AnyLogic 5 / Ю.Г. Карпов. – СПб.: БХВ-Петербург, 2005. – 390 с.

13. Скородумов, П.В. Имитационное моделирование экономических систем: программные средства и направления их совершенствования / П.В. Скородумов // Проблемы развития территории. – Вологда: ВолНЦ РАН, 2015. – № 2(76). – С.62 – 72

14. Чернышова, Н. Н. Имитационное моделирование бизнес-процессов: учеб.-метод. пособие / Н. Н. Чернышова. – Н. Новгород : НГУ им. Лобачевского, 2010. – 28 с.

15. Кудрявцев, Е.М. Основы имитационного моделирования различных систем. / Е.М. Кудрявцев. — М.: ДМК пресс, 2004. – 320 с.

16. Коровин, А.М. Анализ подходов и программного обеспечения для имитационного моделирования социальных и экономических систем / А.М. Коровин // Вестник ЮУрГУ. Серия: Компьютерные технологии, управление, радиоэлектроника. – Челябинск: Издательство ЮУрГУ, 2012. – № 36. – С.98 – 100.

17. Михеева, Т.В. Обзор существующих программных средств имитационного моделирования при исследовании механизмов функционирования и управления производственными системами / Т.В. Михеева // Известия Алтайского государственного университета. Управление,

вычислительная техника и информатика. – Барнаул: Известия АГУ, 2012. – № 36. – С.87 – 90.

18. Свідोцтво про реєстрацію авторських прав на твір №74996 Україна. Комп'ютерна програма “Система імітаційного моделювання СИМ-СИ” (“SIMC”) [Текст] / С. М. Сердюк, Ж.К. Камінська; власн. ЗНТУ; дата реєстрації в ДСІВ України 27.11.2017 р.

19. Смирнов, Б.А. Инженерная психология. Практические занятия / Б.А. Смирнов. – К.: Вища школа, 1979. – 192 с.

20. Советов, Б.Я. Моделирование систем / Б.Я. Советов, С.А. Яковлев. – М.: Высш. шк., 2001. – 343 с.

21. Reilly, R. Eye movement control in reading: A simulation of some word-targeting strategies / R. Reilly, J. K. O'Regan // Vision Research. – 1998. № 38. – P. 303-317

22. Rayner, K. Integrating information across eye movements / K. Rayner, G.W. McConkie, D. Zola // Cognitive Psychology. – 1980. № 12. – P. 206-226

23. Справочник по инженерной психологии / Под ред. Б. Ф. Ломова. – М.: Машиностроение, 1982. – 368 с.

24. Каминская, Ж.К. Моделирование и анализ процессов функционирования автоматизированной системы управления технологическим процессом производства газобетона. / Ж.К. Каминская, Кулинич Э.М., С.Н. Сердюк // Радіоелектроніка, інформатика, управління. – 2018. № 4. – С. 214–225.

25. Осадчий, В.В. Уточнение взаимосвязей и параметров электромеханической системы дозирования вибрационного типа / В.В. Осадчий // Електромеханічні системи, методи моделювання і оптимізації: IX Міжнародна науково-практична конференція, Кривий Ріг, 15-17 травня 2007 р.: тези доповідей. – Вип.4 (45), Част. 2. – Кривий Ріг: КГПУ, 2007. – С. 19-21.

26. Видинеев, Ю.Д. Автоматическое непрерывное дозирование жидкостей / Ю.Д. Видинеев – М.: Энергия, 1967 – 109 с.

27. Остапенко, В.В. Методичні вказівки по економічному

обґрунтуванню дипломних проєктів для студентів спеціальностей 7.080403 «Програмне забезпечення автоматизованих систем», 7.080402 «Інформаційні технології проєктування» і 7.091501 «Комп'ютерні та інтелектуальні системи мережі» всіх форм навчання [Текст] / В.В. Остапенко, Е.М. Касьян, Є.М. Сердюк. – Запоріжжя : ЗНТУ, 2009. – 22 с.

28. Правила улаштування електроустановок [Текст]. Міненерговугілля України, 2017. – 617с.

29. Бедрій Я.І. Основи охорони праці: Навчальний посібник для студентів вищих навчальних закладів. Тернопіль: Навчальна книга – Богдан, 2014. – 240 с.

30. Атаманчук П. Охорона праці в галузі. Навчальний посібник / П. Атаманчук, В. Мендерецький, О. Панчук, Р. Билык. К: Центр навчальної літератури, 2017. – 322 с.

31. ДСанПіН 3.3.2.007-98 «Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин»

32. ДБН В.2.5-28:2018. Природне та штучне освітлення. Посібник: проблеми природного і штучного освітлення. / К: Етіп, 2019. – 180 с.

33. ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку / МОЗ України, 1999 – 24 с.

34. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень. / Харків: Форт, 2000 – 24 с.

35. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. НПАОП 0.00-7.15-18. Державні санітарні правила і норми. / Харків: Форт, 2018 – 48 с.

36. Курс лекцій по «Эргономике». Ю.М. Мадиевский. – Под ред. д-ра. техн. наук., проф. Березуцкого В.В. – Х.: НТУ «ХПИ», 2015. – 294 с

37. Правила пожежної безпеки в Україні. 2015 / Харків : Форт, 2015 – 124 с.

38. ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги». / Київ: Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2017. – 39 с.

39. НПАОП 40.1-1.32-01 «Правила будови електроустановок. Електрообладнання спеціальних установок». / Харків: Індустрія, 2010. – 118 с.
40. ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги» / Харків: Форт, 2004 – 48 с.
41. Васійчук В.О., Гончарук В.Є., Качан С.І., Мохняк С.М. Основи цивільного захисту: Навч. посібник / Львів, 2010.- 384 с.
42. Стеблюк М.І. Цивільна оборона та цивільний захист: Підручник.– К: Знання-Прес, 2007.– 487 с.

ДОДАТОК А
Текст програми

```

#include "simc.h"
#include <cmath>
#include <iomanip>
#include <algorithm>
#include <bits/stdc++.h>
using namespace std;
const double pi = 3.14; // Число Пи
const double g = 9.81; // Ускорение свободного падения (м/с2)
const double p_norm = 101325; // Атмосферное давление (Па)
// Вычисляем поток сухих компонентов из расходной ёмкости
double potok_suxie(double d, double Az, double v) {
    // Az - угол открытия заслонки
    double S = (pi * d * d) / 4 * (1 - cos(Az)); // площадь открытия заслонки
    return S * v;
}
// Вычисляем поток сухих компонентов в дозатор
double potok_doz(double F, double P) {
    // F - поток компонентов из расходной ёмкости
    // P - производительность шнекового питателя
    if(P >= F) return F;
    else return P;
}
// Вычисляем перепад давления для пяти уровней компонентов с шагом в 20%
double dtp(double p, double h) {
    double p_comp = 0;
    for(int i = 8; i >= 0; i -= 2) {
        p_comp += abs(p * g * h - p * g * h * i / 10);
    }
    return p_comp / 5;
}
// Вычисляем скорость истечения суспензии и шлама
double v_zhidkie(double p, double h) {
    double delta_p = dtp(p, h);
    return sqrt(delta_p * 2 * g / p) / 100;
}
// Вычисляем скорость истечения воды
double v_water(double p) {
    double delta_p = 500000;
    return sqrt(delta_p * 2 * g / p) / 100;
}
// Вычисляем поток жидких компонентов из расходной ёмкости и дозатора
double potok_zhidkie(double d, double Az, double v, double R) {
    // Az - угол открытия заслонки
    double S = (pi * d * d) / 4 * (1 - cos(Az)); // площадь открытия заслонки
    return R * S * v;
}
// Вычисляем массу компонента
double massa(double V, double p) {
    return V * p;
}
// Вычисляем коэффициент занятости оператора
double K_zanOp(double t, double T) {
    // t_rob - общее время, в течении которого оператор не занят обработкой
    // информации
    // T - общее время пребывания оператора на рабочем месте
    double t_rob = T - t;
    return 1 - t_rob / T;
}
// Вычисляем коэффициент очереди
double p_ochered(double N_och, double n) {
    // N_och - количество сигналов, обработанных оператором в условиях очереди
    // на обслуживание
    // n - общее количество сигналов, которые поступили

```

```

        return N_och / n;
    }
    // Вывод информации о неполадке
    void nepoladka(double time, int d) {
        // time - количество секунд, с начала рабочего месяца до неполадки
        int s, ch, m;
        s = time / 28800;
        time -= s*28800;
        if(s > 1) {
            time += 28800*(s-1);
            s = 1;
        }
        s++;
        ch = time / 60 / 60;
        time -= ch*60*60;
        m = time / 60;
        time -= m*60;
        outfile << d << " day " << s << " shift " << ch << " hour(s) " << m << "
minute(s) " << int(floor(time)) << " second(s) - ";
    }
    // Генератор случайных чисел по Пуассоновскому распределению
    random_device rd1, rd2, rd3, rd4;
    mt19937 gen1(rd1()), gen2(rd2()), gen3(rd3()), gen4(rd4());
    int poisson(double lb, mt19937& gen) {
        poisson_distribution<> d(lb);
        auto t = d(gen);

        return t;
    }
    int main() {
        double d_vx_syxie = 0.15, // входной диаметр заслонки дозатора сухих (м)
        d_vx_sysp = 0.04, // входной диаметр заслонки дозатора суспензии
d_vx_shl = 0.1, // входной диаметр заслонки дозатора шлама (м)
        d_vx_vod = 0.05, // входной диаметр заслонки дозатора воды (м)
        d_vix_syxie = 0.2, // выходной диаметр заслонки дозатора сухих (м)
        d_vix_sysp = 0.05, // выходной диаметр заслонки дозатора суспензии
d_vix_shl = 0.1, // выходной диаметр заслонки дозатора шлама (м)
        d_vix_vod = 0.1, // выходной диаметр заслонки дозатора воды (м)
        d_smes = 0.250, // выходной диаметр заслонки смесителя (м)
        v_IKV = 0.050, // скорость истечения ИКВ (м3/сек)
        v_cem = 0.050, // скорость истечения цемента (м3/сек)
        P = 0.0037037037, // производительность шнекового питателя (м3/сек)
        Az_IKV = pi/2, //Az - угол открытия заслонки ИКВ (рад)
        Az_cem = pi/2, //Az - угол открытия заслонки цемента (рад)
        Az_vod = pi/2, //Az - угол открытия заслонки воды (рад)
        Az_shl = pi/2, //Az - угол открытия заслонки шлама (рад)
        Az_sys = pi/2, //Az - угол открытия заслонки суспензии (рад)
        Az_smes = pi/2, //Az - угол открытия заслонки смесителя (рад)

        M_IKV = 180.3, // необходимая масса ИКВ по рецепту (кг)
        M_cem = 76.5, // необходимая масса цемента по рецепту (кг)
        M_water = 120.1, // необходимая масса воды по рецепту (кг)
        M_sysp = 17.4, // необходимая масса суспензии по рецепту (кг)
        M_shlam = 323.1, // необходимая масса шлама по рецепту (кг)
        t_pr_dop = 0, // допустимое время пребывания информации на
обслуживании (сек)
        potok_IKV_izemk, // поток ИКВ из расходной ёмкости в дозатор
(м/сек)

        potok_IKV_vdoz, // поток ИКВ в смеситель из дозатора (м/сек)
        potok_IKV_izdoz, // поток ИКВ в смеситель из дозатора (м/сек)
        m_IKV_vdoz, // масса ИКВ, поступающая в дозатор (кг/сек)
        m_IKV_vsmes, // масса ИКВ, поступающая в смеситель (кг/сек)

```

```

    potok_cem_izemk, // поток цемента из расходной ёмкости в дозатор
(м/сек)
    potok_cem_vdoz, // поток цемента в смеситель из дозатора (м/сек)
    potok_cem_izdoz, // поток цемента в смеситель из дозатора (м/сек)
    m_cem_vdoz, // масса цемента, поступающая в дозатор (кг/сек)
    m_cem_vsmes, // масса цемента, поступающая в смеситель (кг/сек)
    R = 0.95, // Коэффициент расхода R
    p_syx = 1300, // плотность сухих компонентов (кг/м3)
    pvod = 1000, // плотность воды (кг/м3)
    psys = 2700, // плотность суспензии (кг/м3)
    pshl = 2650, // плотность шлама (кг/м3)
    psmesi = 1800, // плотность смеси (кг/м3)
    h_emk_sysp = 1, // высота расходной емкости суспензии (м)
    h_emk_shlam = 2.6, // высота расходной емкости шлама (м)
    h_emk_smesi = 3, // высота смесителя (м)
    v_wat, // скорость истечения воды
    v_sysp, // скорость истечения суспензии
    v_shlam, // скорость истечения шлама
    v_smes, // скорость истечения смеси
    potok_water_izemk, // поток воды, поступающей в дозатор
    potok_water_izdoz, // поток воды, поступающей в смеситель
    m_wat_vdoz, // масса воды, поступающей в дозатор
    m_wat_vsmes, // масса воды, поступающей в смеситель
    potok_sysp_izemk, // поток суспензии, поступающей в дозатор
    potok_sysp_izdoz, // поток суспензии, поступающей в смеситель
    m_sysp_vdoz, // масса суспензии, поступающей в дозатор
    m_sysp_vsmes, // масса суспензии, поступающей в смеситель
    potok_shlam_izemk, // поток шлама, поступающего в дозатор
    potok_shlam_izdoz, // поток шлама, поступающего в смеситель
    m_shl_vdoz, // масса шлама, поступающей в дозатор
    m_shl_vsmes, // масса шлама, поступающей в смеситель
    potok_izsmes, // поток смеси из смесителя
    m_smes, // масса смеси из смесителя

    count[31], // время занятости оператора за каждый цикл (сек)
    Z, Z2, // коэффициент занятости оператора
    p, p2, // коэффициент очереди
    q_lm[31], // значения средней длины очереди
    q_lm_sr = 0, // средняя длина очереди
    t_ozh_dop = 0, t_ozh_dop2 = 0, // допустимое время ожидания
информации в очереди
    t_obr_oper = 0, t_obr_oper2 = 0, // время обработки информации
оператором
    q_task = 0, // количество заданий, выполненных в условиях очереди1
    q_task2 = 0, // количество заданий, выполненных в условиях очереди2
    q_time = 0, // общее время выполнения заданий
    task_total = 0, // общее количество выполненных заданий1
    task_total2 = 0, // общее количество выполненных заданий2
    q_time_er = 0, // общее время нахождения в очереди1
    q_time_er2 = 0, // общее время нахождения в очереди2
    fstShift = 0,
    max_buzy = 0,
    random = 0,
    time_zav = 0,
    gisto = 0;
    time8 = 8*60*60, //8-часовой рабочий день (сек) = 28800с * 2 смены

    int
= 57600с
    q_mq = 0, // максимальная длина очереди
    q_mq2 = 0, // максимальная длина очереди
    zagr_smes = 0, // загруженность смесителя (сколько кг в нем)
    Pya = 0, // случайное число по Пуассону
    k = 1,
    zhour = 0,

```

```

zan_hour[31],
z_op1 = 0,
z_op2 = 0,
mix_end = 0,
cycle = 1, // счётчик циклов
need_stat_day = 15,
month = 30;
// флаги поломок
bool end = false,
peregrev = false,
fst = true,
done = true,
done2 = true,
done3 = true,
loose = false,
s15th = true;
enum oper {free, seized};
pfacility oper, // прибор-оператор
f_shlam_water, // прибор-дозатор шлама и воды
f_syxie, // прибор-дозатор сухих компонентов
f_sysp, // прибор-дозатор суспензии
f_smes; // прибор-смеситель
pqueue q_work, q_error;
initlist(80000);
newfac(oper, "Operator");
newfac(f_shlam_water, "Sludge and water dispenser");
newfac(f_syxie, "Dry component dispenser");
newfac(f_sysp, "Suspension dispenser");
newfac(f_smes, "Mixer");
newqueue(q_work, "Zan_Oper");
newqueue(q_error, "Zan_Oper_Nepoladki");
// вычисляем поток ИКВ из расходной ёмкости в дозатор, из дозатора и в
смеситель
potok_IKV_izemk = potok_suxie(d_vx_syxie, Az_IKV, v_IKV);
potok_IKV_vdoz = potok_doz(potok_IKV_izemk, P);
potok_IKV_izdoz = potok_suxie(d_vix_syxie, Az_IKV, v_IKV);
// вычисляем массу ИКВ, поступающую в дозатор и в смеситель
m_IKV_vdoz = massa(potok_IKV_vdoz, p_syx);
m_IKV_vsmes = massa(potok_IKV_izdoz, p_syx);
// вычисляем поток цемента из расходной ёмкости в дозатор, из дозатора и в
смеситель
potok_cem_izemk = potok_suxie(d_vx_syxie, Az_cem, v_cem);
potok_cem_vdoz = potok_doz(potok_cem_izemk, P);
potok_cem_izdoz = potok_suxie(d_vix_syxie, Az_cem, v_cem);
// вычисляем массу цемента, поступающую в дозатор и в смеситель
m_cem_vdoz = massa(potok_cem_vdoz, p_syx);
m_cem_vsmes = massa(potok_cem_izdoz, p_syx);
// вычисляем скорость истечения воды
v_wat = v_water(pvod);
// вычисляем скорость истечения суспензии
v_sysp = v_zhidkie(psys, h_emk_sysp);
// вычисляем скорость истечения шлама
v_shlam = v_zhidkie(pshl, h_emk_shlam);
// вычисляем скорость истечения смеси
v_smes = v_zhidkie(psmesi, h_emk_smesi);
// вычисляем поток воды из расходной ёмкости
potok_water_izemk = potok_zhidkie(d_vx_vod, Az_vod, v_wat, R);
// вычисляем поток суспензии из расходной ёмкости
potok_sysp_izemk = potok_zhidkie(d_vx_sysp, Az_sys, v_sysp, R);
// вычисляем поток шлама из расходной ёмкости
potok_shlam_izemk = potok_zhidkie(d_vx_shl, Az_shl, v_shlam, R);
// вычисляем поток воды из дозатора
potok_water_izdoz = potok_zhidkie(d_vix_vod, Az_vod, v_wat, R);

```

```

// вычисляем поток суспензии из дозатора
potok_syvsp_izdoz = potok_zhidkie(d_vix_syvsp, Az_sys, v_syvsp, R);
// вычисляем поток шлама из дозатора
potok_shlam_izdoz = potok_zhidkie(d_vix_shl, Az_shl, v_shlam, R);
// вычисляем массу воды, поступающую в дозатор
m_wat_vdoz = massa(potok_water_izemk, pvod);
// вычисляем массу суспензии, поступающую в дозатор
m_syvsp_vdoz = massa(potok_syvsp_izemk, psys);
// вычисляем массу шлама, поступающую в дозатор
m_shl_vdoz = massa(potok_shlam_izemk, pshl);
// вычисляем массу воды, поступающую в смеситель
m_wat_vsmes = massa(potok_water_izdoz, pvod);
// вычисляем массу суспензии, поступающую в смеситель
m_syvsp_vsmes = massa(potok_syvsp_izdoz, psys);
// вычисляем массу шлама, поступающую в смеситель
m_shl_vsmes = massa(potok_shlam_izdoz, pshl);
// вычисляем поток смеси из смесителя
potok_izsmes = potok_zhidkie(d_smes, Az_smes, v_smes, R);
// вычисляем массу смеси из смесителя
m_smes = massa(potok_izsmes, psmesi);
double recipe_time[4] = {M_IKV / m_IKV_vdoz,
                          M_cem / m_cem_vdoz,
                          M_syvsp / m_syvsp_vdoz,
                          M_shlam / m_shl_vdoz};

double recipe[4] = {M_IKV,
                    M_cem,
                    M_syvsp,
                    M_shlam};
double recipe_second[4] = {m_IKV_vdoz,
                           m_cem_vdoz,
                           m_syvsp_vdoz,
                           m_shl_vdoz};

// Вычисляем, какое вещество будет набираться в дозатор дольше всех
double *max_speed = max_element(recipe_time, recipe_time+3);
int dist = distance(recipe_time, max_speed);
count[0] = 0;
outfile << "Proizvoditelnost shneka: " << P << "<BR>";
outfile << "Potok syxix iz emkosti: " << potok_IKV_izemk << "<BR>";
outfile << "Vremya nabora IKV po recepty: " << recipe_time[0] << "<BR>";
outfile << "Vremya nabora cementa po recepty: " << recipe_time[1] <<
"<BR>";
outfile << "Vremya nabora syspensii po recepty: " << recipe_time[2] <<
"<BR>";
outfile << "Vremya nabora shlama po recepty: " << recipe_time[3] << "<BR>";
outfile << "Vremya nabora vodi po recepty: " << M_water / m_wat_vdoz <<
"<BR><BR>";
outfile << "Vremya vigruzki shlama iz dozatora: " << M_shlam / m_shl_vsmes
<< "<BR>";
outfile << "Vremya vigruzki syxix iz dozatora: " << (M_IKV + M_cem) /
((m_IKV_vsmes + m_cem_vsmes) / 2) << "<BR><BR>";
string stat;
outfile << fixed << setprecision(2);
//starttrace();
for(int day = 1; day <= 30; day++) {
    initcreate(1, 0); // запуск системы
    initcreate(34, 0); // на дозатор попадают посторонние предметы
    initcreate(157, 0); // Перегрев привода
    initcreate(162, 0); // Инородное тело попадает в шнек
    initcreate(180, 0); // Завис материал
    if(day == 29)
        initcreate(154, 54000); // запись данных для гистограммы за первый
        час n-го дня работы
    end = false;

```

```

time_zav = 0;
gisto = 0;
cycle = 1;
while(!end) {
    plan();
    switch(sysevent) {
        case 1: inqueue(q_work);
                seize(oper);
                break;
        case 2: // Включить рубильник и дождаться загрузки системы.
                delayt(randab(28, 32, v1));
                break;
        case 3: // Проверка, нет ли системных ошибок. Ошибок нет
                delayt (0.4);
                break;
        case 4: // Нажать кнопку «Общий вид» P=0,9973;T=0,18±0,03с
                if(randab(0, 10000, v3) >= 9973)
                    k = 2;
                delayt(randab(k*0.15, k*0.21, v2));
                k = 1;
                break;
        case 5: // Переход в экран «Общий вид» T=2,0±0,2 с
                delayt(randab(1.8, 2.2, v4));
                break;
        case 6: // Включить тумблер «ЦЕПИ УПРАВЛЕНИЯ» P=0,9979;
                if(randab(0, 10000, v3) >= 9979)
                    k = 2;
                delayt(randab(k*0.62, k*0.66, v5));
                k = 1;
                break;
        // Проверка нулевых значений в 3-х дозаторах «Суспензии»,
        «Шлама», «Сухих»
        case 7: // Считывание первого: T=0,150 + 0,028=0,178±0,018
                if(randab(0, 10000, v3) >= 9980)
                    k = 2;
                delayt(randab(k*0.16, k*0.196, v6));
                k = 1;
                break;
        case 8: // Проверка логического условия (т.е. что значения
                не 0) T=0,29±0,11 с; P=0,9966
                if(randab(0, 10000, v3) >= 9966)
                    k = 2;
                delayt(randab(k*0.18, k*0.4, v6));
                k = 1;
                break;
        case 9: // Саккада + Считывание второго: T=0,328± 0,068 с;
                if(randab(0, 10000, v3) >= 9980)
                    delayt(randab(k*0.16, k*0.196, v6));
                else
                    delayt(randab(k*0.26, k*0.396, v6));
                break;
        case 10: // Проверка логического условия (т.е. что значения
                не 0) T=0,29±0,11 с; P=0,9966
                if(randab(0, 10000, v3) >= 9966)
                    k = 2;
                delayt(randab(k*0.18, k*0.4, v6));
                k = 1;
                break;
    }
}

```

```

case 11: // Саккада + Считывание третьего: T=0,328± 0,068
с; P=0,998;
    if(randab(0, 10000, v3) >= 9980)
        delayt(randab(k*0.16, k*0.196, v10));
    else
        delayt(randab(k*0.26, k*0.396, v10));
    break;
case 12: // Проверка логического условия (т.е. что значения
не 0) T=0,29±0,11 с; P=0,9966
    if(randab(0, 10000, v3) >= 9966)
        k = 2;
    delayt(randab(k*0.18, k*0.4, v6));
    k = 1;
    break;
case 13: // Вероятность того, что где-то будет наливание
(или не нулевое значение) в 3-х дозаторах P=0,99
    inqueue(q_error);
    if(randab(0, 10000, v3) > 9900)
        next(53);
    break;
case 14: // Переключили тумблер на ручной режим «РУЧ»
P=0,9969; T=0,84±0,12 с
    if(randab(0, 10000, v3) >= 9969)
        k = 2;
    delayt(randab(k*0.82, k*1.06, v6));
    k = 1;
    break;
// Переход в «Стартовый экран дозировки» и выбор нужного
дозатора
case 15: // Нажатие кнопки F14 P=0,9951; T=0,48±0,13 с.
    if(randab(0, 10000, v3) >= 9951)
        k = 2;
    delayt(randab(k*0.35, k*0.61, v6));
    k = 1;
    break;
case 16: // Время перехода в экран «Стартовый экран
дозировки» T=2,0±0,2 с
    delayt(randab(1.8, 2.2, v6));
    break;
case 17: // ПРИ ВЫБОРЕ ДОЗАТОРА СУХИХ
    if (randab(0, 10000, v3) > 1000)
        next(53);
    break;
case 18: // Выбор нужного дозатора (нажатие кнопки)
P=0,9961; T=0,48±0,13 с.
    if(randab(0, 10000, v3) >= 9961)
        k = 2;
    delayt(randab(k*0.35, k*0.61, v6));
    k = 1;
    break;
case 19: // Задержка на переход в форму «ДОЗАТОР СУХИХ»
T=2,0±0,2 с
    delayt(randab(1.8, 2.2, v6));
    break;
case 20: // Нажать кнопку F11 (открыть заслонку на
выгрузку). Затем включить обдув (вибраторы), т.е. нажать F10 T=50±10 с.
    delayt(randab(40, 60, v6));
    break;
case 21: // Время перехода в экран «Дозатор сухих»
T=2,0±0,2 с
    delayt(randab(1.8, 2.2, v6));
    break;

```

```

        case 22: // Нажать кнопку F11 (открыть заслонку на
выгрузку) - P=0,9951; T=0,48±0,13 с.
            if(randab(0, 10000, v3) >= 9951)
                k = 2;
            delayt(randab(k*0.35, k*0.61, v6));
            k = 1;
            break;
        case 23: // Время открытия заслонки 1,95±0,05 с.
            delayt(randab(1.9, 2.0, v6));
            break;
        case 24: // Затем включить обдув (вибраторы), т.е. нажать
кнопку F10 - P=0,9951; T=0,48±0,13 с. При ошибке - повторить T=0,48±0,13 с.
            if(randab(0, 10000, v3) >= 9951)
                k = 2;
            delayt(randab(k*0.35, k*0.61, v6));
            k = 1;
            break;
        case 25: // Обдув дозатора. Время обдува T=50±10 с.
            delayt(randab(40, 60, v6));
            break;
        case 26: // Выключение обдува (нажатие кнопки F10)
P=0,9951; T=0,48±0,13 с.
            if(randab(0, 10000, v3) >= 9951)
                k = 2;
            delayt(randab(k*0.35, k*0.61, v6));
            k = 1;
            break;
        case 27: // Ожидание снятия избыточного давления в дозаторе
2,0±0,1 с.
            delayt(randab(1.9, 2.1, v6));
            break;
        case 28: // Закрыть заслонку (нажать кнопку F12) -
P=0,9951; T=0,48±0,13 с.
            if(randab(0, 10000, v3) >= 9951)
                k = 2;
            delayt(randab(k*0.35, k*0.61, v6));
            k = 1;
            break;
        case 29: // Время закрытия заслонки 1,95±0,05 с.
            delayt(randab(1.9, 2.0, v6));
            break;
        case 30: // Считывание значений массы компонентов с экрана
«Дозатор сухих» T=0,178±0,018 с; P=0,998
            if(randab(0, 10000, v3) >= 9980)
                k = 2;
            delayt(randab(k*0.16, k*0.196, v6));
            k = 1;
            break;
        case 31: // Проверка логического условия (т.е. что значения
0) T=0,29±0,11 с; P=0,9966
            if(randab(0, 10000, v3) >= 9966)
                k = 2;
            delayt(randab(k*0.18, k*0.4, v6));
            k = 1;
            break;
        case 32: // После этого, считаем, что весы в 0
            zagr_smes = 0;
            break;
        case 33:
            next(41);
            break;
        case 34: // По Пуассоновскому закону
create(1);

```

```

        break;
case 35: Pya = poisson(3.5e-6, gen1);
        break;
case 36: if(!Pya)
        next(40);
        break;
case 37: nepoladka(systime, day);
        outfile << "debris caught on the dispenser." <<

"<BR>";

        inqueue(q_error);
        infac(oper);
        break;
case 38: // Визуальный осмотр дозатора T=65±50 с
        random = randab(15, 115, v6);
        if(max_buzy < random) {
            max_buzy = random;
        }
        gisto += random;
        delayt(random);
        break;
case 39: outqueue(q_error);
        outfac(oper);
        break;
case 40: destroy();
        break;
// Возврат в форму «Общий вид»
case 41: // Нажать кнопку F14 P=0,9973; T=0,18±0,03 с.
        if(randab(0, 10000, v3) >= 9973)
            k = 2;
        delayt(randab(k*0.15, k*0.21, v6));
        k = 1;
        break;
case 42: // Время перехода в экран T=2,0±0,2 с
        delayt(randab(1.8, 2.2, v6));
        break;
case 43: //Нажать кнопку «Смеситель» P=0,9973; T=0,18±0,03с
        if(randab(0, 10000, v3) >= 9973)
            k = 2;
        delayt(randab(k*0.15, k*0.21, v6));
        k = 1;
        break;
case 44: // Время перехода в экран «Смеситель»T=2,0±0,2 с
        delayt(randab(1.8, 2.2, v6));
        break;
case 45: // Включаем мешалку (нажать кнопку F4) P=0,9973;

T=0,18±0,03 с.

        if(randab(0, 10000, v3) >= 9973)
            k = 2;
        delayt(randab(k*0.15, k*0.21, v6));
        k = 1;
        break;
case 46: // Открываем заслонку, нажатием кнопки F11

P=0,9973; T=0,18±0,03 с.

        if(randab(0, 10000, v3) >= 9973)
            k = 2;
        delayt(randab(k*0.15, k*0.21, v6));
        k = 1;
        break;
case 47: // Время открытия заслонки 1,95±0,05 с
        delayt(randab(1.9, 2.0, v6));
        break;
case 48: // Очищаем Смеситель
        delayt(zagr_smes / m_smes);

```

```

        zagr_smes = 0;
        break;
    case 49: // Закрываем заслонку, нажатием кнопки F12
P=0,9973; T=0,18±0,03 с.
        if(randab(0, 10000, v3) >= 9973)
            k = 2;
        delayt(randab(k*0.15, k*0.21, v6));
        k = 1;
        break;
    case 50: // Время закрытия заслонки 1,95±0,05 с
        delayt(randab(1.9, 2.0, v6));
        break;
    case 51: // Переход в экран «Общий вид»: нажатием кнопки
F14 P=0,9973; T=0,18±0,03 с.
        if(randab(0, 10000, v3) >= 9973)
            k = 2;
        delayt(randab(k*0.15, k*0.21, v6));
        k = 1;
        break;
    case 52: // Время перехода в экран T=2,0±0,2 с
        delayt(randab(1.8, 2.2, v6));
        break;
    // Выбор рецепта
    case 53: // Нажать кнопку «Задать рецепт»: P=0,9973;
T=0,18±0,03 с.
        outqueue(q_error);
        if(randab(0, 10000, v3) >= 9973)
            k = 2;
        delayt(randab(k*0.15, k*0.21, v6));
        k = 1;
        break;
    case 54: // Время перехода в экран T=2,0±0,2 с
        delayt(randab(1.8, 2.2, v6));
        break;
    case 55: // Выбор рецепта из списка (11наименований):
P=0,9919, T=1,128±0,09 с.
        outqueue(q_work);
        if(randab(0, 10000, v3) >= 9919)
            k = 2;
        delayt(randab(k*1.119, k*1.137, v8));
        k = 1;
        break;
    case 56: // Нажать кнопку «Пуск» на пульте P=0,9973;
T=0,18±0,03 с.
        inqueue(q_work);
        if(oper->status == free)
            seize(oper);
        if(randab(0, 10000, v3) >= 9973)
            k = 2;
        delayt(randab(k*0.15, k*0.21, v6));
        k = 1;
        break;
    case 57: // Убедиться в готовности линии к пуску, что
индицируется миганием зелёной лампочки кнопки «Пуск»: P=0,95; T=0,7±0,25 с.
        if(randab(0, 10000, v3) >= 9500)
            k = 2;
        delayt(randab(k*0.45, k*0.95, v6));
        k = 1;
        break;
    case 58: //После нажатия "Пуск" - оператор в экране «Общий
вид» контролирует набор веса в течение T=11±3 с.
        delayt(randab(8, 14, v6));
        break;

```

```

case 59: // Набираем ИКВ, цемент, шлам и суспензию в
дозаторы по рецепту
    outqueue(q_work);
    max_buzy = (recipe[dist] - 20) /
recipe_second[dist];
    delayt(max_buzy);
    break;
case 60: // Уменьшаем точность набора до 10% за 20 кг
    delayt(20 / (recipe_second[dist] * 0.1));
    break;
case 61: // Нажать эту кнопку F12. P=0,9973; T=0,18±0,03 с.
    if(randab(0, 10000, v3) >= 9973)
        k = 2;
    delayt(randab(k*0.15, k*0.21, v6));
    k = 1;
    break;
case 62: outfac(oper);
    break;
case 63: seize(f_shlam_water);
    break;
case 64: // Высыпаем шлам в смеситель
    delayt(M_shlam / m_shl_vsmes);
    zagr_smes += M_shlam;
    break;
case 65: // Загружаем воду в дозатор шлама, выливаем воду в
смеситель
    delayt((M_water / m_wat_vdoz) + (M_water /
m_wat_vsmes));
    zagr_smes += M_water;
    break;
case 66: outfac(f_shlam_water);
    break;
case 67: seize(f_syxie);
    break;
case 68: // Затем выгружается все, что в дозаторе сухих в
смеситель, в это время в дозатор шлама начинает загружаться новая порция
    delayt((M_IKV + M_cem) / ((m_IKV_vsmes +
m_cem_vsmes) / 2));
    zagr_smes += (M_IKV + M_cem);
    break;
case 69: outfac(f_syxie);
    break;
case 70: seize(f_smes);
    break;
case 71: // Перемешивается в течение T=60±3 с
    delayt(randab(57, 63, v6));
    break;
case 72: outfac(f_smes);
    break;
case 73: seize(f_syxp);
    break;
case 74: // Напоследок выгружается ал.суспензия
    delayt(M_syxp / m_syxp_vsmes);
    zagr_smes += M_syxp;
    break;
case 75: outfac(f_syxp);
    break;
case 76: seize(f_smes);
    break;
case 77: // Все это перемешивается в течении T=220±20 с
    mix_end = randab(200, 240, v1) + systime;
    break;
case 78: outfac(f_smes);

```

```

        break;
case 79: seize(oper);
        break;
case 80: // С вероятностью P=0,03 оператор может забыть
пополнить материал
        if(randab(0, 10000, v3) > 300)
            next(103);
        break;
// Авария дозатора сухих: нет материала в бункере сухих
(оператор забыл наполнить) P=0,03
case 81: // Мигает кнопка «Авария». Восприятие зрительного
сигнала T=1,1±0,1 с
        nepoladka(sysstime, day);
        outfile << "there is no material in the hopper of
dry materials." << "<BR>";
        inqueue(q_error);
        delayt(randab(1.0, 1.2, v6));
        break;
case 82: // Саккада + считывание первых 18 символов
T=0,523±0,087 с, P=0,998.
        inqueue(q_work);
        if(randab(0, 10000, v3) >= 9980)
            k = 2;
        delayt(randab(k*0.436, k*0.61, v6));
        k = 1;
        break;
case 83: // Фиксация + считывание последних 15 знаков
T=0,728±0,033 с P=0,998.
        if(randab(0, 10000, v3) >= 9980)
            k = 2;
        delayt(randab(k*0.695, k*0.761, v6));
        k = 1;
        break;
case 84: // Нажатие кнопки XXX. F14P=0,9973; T=0,18±0,03 с.
        if(randab(0, 10000, v3) >= 9973)
            k = 2;
        delayt(randab(k*0.15, k*0.21, v6));
        k = 1;
        break;
case 85: // Время перехода в экран «Дозатор сухих»
T=2,0±0,2 с
        delayt(randab(1.8, 2.2, v6));
        break;
case 86: // Саккада + Считывание: T=0,328± 0,068 с;
P=0,998;
        if(randab(0, 10000, v3) >= 9980)
            delayt(randab(k*0.16, k*0.196, v6));
        else {
            delayt(randab(k*0.26, k*0.396, v6));
        }
        break;
case 87: // Проверка логического условия (т.е. что значения
не 0) T=0,29±0,11 с; P=0,9966
        if(randab(0, 10000, v3) >= 9966)
            k = 2;
        delayt(randab(k*0.18, k*0.4, v6));
        k = 1;
        break;
case 88: // Нажать кнопку «ДЕБЛОКИРОВКА» P=0,9973;
T=0,18±0,03 с.
        if(randab(0, 10000, v3) >= 9973)
            k = 2;
        delayt(randab(k*0.15, k*0.21, v6));

```

```

        k = 1;
        break;
    case 89: // Нажать кнопку «ПУСК» P=0,9973; T=0,18±0,03 с.
if(randab(0, 10000, v3) >= 9973)
        k = 2;
        delayt(randab(k*0.15, k*0.21, v6));
        k = 1;
        break;
// Проверка работы заслонок (Визуально Анимация и ВЕС)
    case 90: // Проверка анимации T=2,2±0,2 с
        delayt(randab(2.0, 2.4, v6));
        break;
    case 91: // Проверка веса Саккада + считывание 5 символов
T=0,328±0,068 с
        delayt(randab(0.26, 0.396, v6));
        break;
    case 92: // Нажать кнопку «ПРЕРЫВАНИЕ» (она мигает)
P=0,9973; T=0,18±0,03 с.
        if(randab(0, 10000, v3) >= 9973)
            k = 2;
            delayt(randab(k*0.15, k*0.21, v6));
            k = 1;
            break;
    case 93: // Нажать кнопку F14 для перехода в основное меню.
P=0,9973; T=0,18±0,03 с.
        if(randab(0, 10000, v3) >= 9973)
            k = 2;
            delayt(randab(k*0.15, k*0.21, v6));
            k = 1;
            break;
    case 94: // Загрузка формы T=2,0±0,2 с
        delayt(randab(1.8, 2.2, v6));
        break;
    case 95: // Нажать кнопку «Дозатор сухих» . P=0,9973;
T=0,18±0,03 с.
        if(randab(0, 10000, v3) >= 9973)
            k = 2;
            delayt(randab(k*0.15, k*0.21, v6));
            k = 1;
            break;
    case 96: // Переключить тумблер на «РУЧНОЙ» P=0,9979;
T=0,64±0,02 с.
        if(randab(0, 10000, v3) >= 9979)
            k = 2;
            delayt(randab(k*0.62, k*0.66, v6));
            k = 1;
            break;
    case 97: // Нажать F6 (открыть заслонку) . P=0,9973;
T=0,18±0,03 с.
        if(randab(0, 10000, v3) >= 9973)
            k = 2;
            delayt(randab(k*0.15, k*0.21, v6));
            k = 1;
            break;
    case 98: // Включить шнек (Нажать F2 и удерживать ее) ВЕС.
P=0,9973; T=0,18±0,03 с.
        if(randab(0, 10000, v3) >= 9973)
            k = 2;
            delayt(randab(k*0.15, k*0.21, v6));
            k = 1;
            break;
    case 99: // Считывать показания веса в течение T=10±2 с
        delayt(randab(8, 12, v6));

```

```

        break;
case 100: // Включает кнопку «Обрушение» P=0,9973;
    if(randab(0, 10000, v3) >= 9973)
        k = 2;
    delayt(randab(k*0.15, k*0.21, v6));
    k = 1;
    break;
case 101: // Проверяет показания веса в течение T=15±5 с
    delayt(randab(10, 20, v6));
    break;
case 102: // Наполнить бункер T=750±450 с
    if(loose) {
        outqueue(q_work);
        next(185);
    }
    else {
        outqueue(q_work);
        outqueue(q_error);
        delayt(randab(300, 1200, v6));
    }
    break;
case 103: // Если следующий цикл имеет №2, то оператор
может открыть настройки
    if(cycle+1 != 2)
        next(146);
    break;
// 2-й цикл
case 104: // Вероятность того, что будет дисбаланс
(отклонение от заданного) весов вообще P=0,75
    if(randab(0, 10000, v3) >= 7500)
        next(146);
    break;
case 105: // Переход в «Общий вид» по нажатию кнопки
F14P=0,9973; T=0,18±0,03 с.
    inqueue(q_work);
    if(randab(0, 10000, v3) >= 9973)
        k = 2;
    delayt(randab(k*0.15, k*0.21, v6));
    k = 1;
    break;
case 106: // Время перехода в экран «Общий вид» T=2,0±0,2 с
    delayt(randab(1.8, 2.2, v6));
    break;
case 107: // Саккада на поле +считывание значений T=0,328±
0,068 с; P=0,998;
    if(randab(0, 10000, v3) >= 9980)
        delayt(randab(k*0.16, k*0.196, v6));
    else {
        delayt(randab(k*0.26, k*0.396, v6));
    }
    break;
case 108: // Саккада + Считывание второго значения: T=0,150
+ 150+ 0,028=0,328± 0,068 с; P=0,998;
    if(randab(0, 10000, v3) >= 9980)
        delayt(randab(k*0.16, k*0.196, v6));
    else {
        delayt(randab(k*0.26, k*0.396, v6));
    }
    break;
case 109: // Проверка логического условия (ИЛИ, т.е. что
значения не равны) T=0,29±0,11 с; P=0,9966
    if(randab(0, 10000, v3) >= 9966)

```

```

        k = 2;
        delayt(randab(k*0.18, k*0.4, v6));
        k = 1;
        break;
case 110: // Саккада на поле +считывание значений T=0,328±
0,068 с; P=0,998;
        if(randab(0, 10000, v3) >= 9980)
            delayt(randab(k*0.16, k*0.196, v6));
        else {
            delayt(randab(k*0.26, k*0.396, v6));
        }
        break;
case 111: // Саккада + Считывание второго значения: T=0,150
+ 150+ 0,028=0,328± 0,068 с; P=0,998;
        if(randab(0, 10000, v3) >= 9980)
            delayt(randab(k*0.16, k*0.196, v6));
        else {
            delayt(randab(k*0.26, k*0.396, v6));
        }
        break;
case 112: // Проверка логического условия (т.е. что
значения не равны) T=0,29±0,11 с; P=0,9966.
        if(randab(0, 10000, v3) >= 9966)
            k = 2;
        delayt(randab(k*0.18, k*0.4, v6));
        k = 1;
        break;
case 113: // Нажать кнопку F14 P=0,9973; T=0,18±0,03 с.
if(randab(0, 10000, v3) >= 9973)
        k = 2;
        delayt(randab(k*0.15, k*0.21, v6));
        k = 1;
        break;
case 114: // Время перехода в экран «Настройки дозатора
сухих» T=2,0±0,2 с
        outqueue(q_work);
        delayt(randab(1.8, 2.2, v6));
        break;
case 115: // Дисбаланс цемента
        if(randab(0, 10000, v3) >= 4000)
            next(128);
        break;
// Изменение первых трех строк
case 116: // Прочитать первую строку «Упреждение...»
T=1,326±0,163 с. P=0,998. inqueue(q_work);
        if(randab(0, 10000, v3) >= 9980)
            k = 2;
        delayt(randab(k*1.163, k*1.489, v6));
        k = 1;
        break;
case 117: // Ввести новое значение (3 символа ): Нажать
кнопку T=0,18±0,03 с P=0,9973.
        if(randab(0, 10000, v3) >= 9973)
            k = 2;
        delayt(randab(k*0.15, k*0.21, v6));
        k = 1;
        break;
case 118:// Затем T=2,4±0,24 с P=0,995.
if(randab(0, 10000, v3) >= 9950)
        k = 2;
        delayt(randab(k*2.16, k*2.64, v6));
        k = 1;
        break;

```

```

        case 119: // Саккада на 2-ю строку и считывание первых 16
СИМВОЛОВ T=0,493±0,084 с, P=0,998.
            if(randab(0, 10000, v3) >= 9980)
                k = 2;
            delayt(randab(k*0.409, k*0.569, v6));
            k = 1;
            break;
        case 120: // Фиксация + считывание последних 18 знаков
T=0,4±0,373=0,773±0,037 с, P=0,998.        if(randab(0, 10000, v3) >= 9980)
            k = 2;
            delayt(randab(k*0.736, k*0.81, v6));
            k = 1;
            break;
        case 121: // Ввести новое значение (4 символа ): Нажать
кнопку T=0,18±0,03 с P=0,9973.
            if(randab(0, 10000, v3) >= 9973)
                k = 2;
            delayt(randab(k*0.15, k*0.21, v6));
            k = 1;
            break;
        case 122: // Затем T=3,2±0,32 с, P=0,995. При ошибке -
повторить T=3,2±0,32 с
            if(randab(0, 10000, v3) >= 9950)
                k = 2;
            delayt(randab(k*2.88, k*3.52, v6));
            k = 1;
            break;
        // Прочитать третью строку «Набор...»
        case 123: // Саккада на 3-ю строку и считывание 22 символов
T=0,583±0,093 с, P=0,998.
            if(randab(0, 10000, v3) >= 9980)
                k = 2;
            delayt(randab(k*0.49, k*0.676, v6));
            k = 1;
            break;
        case 124: // Фиксация T=0,4 с
            delayt(0.4);
            break;
        // Ввести новое значение (5 символов)
        case 125: // Нажать кнопку T=0,18±0,03 с P=0,9973.
if(randab(0, 10000, v3) >= 9973)
            k = 2;
            delayt(randab(k*0.15, k*0.21, v6));
            k = 1;
            break;
        case 126: // Затем T=4,0±0,4 с, P=0,995.
if(randab(0, 10000, v3) >= 9950)
            k = 2;
            delayt(randab(k*3.6, k*4.4, v6));
            k = 1;
            break;
        case 127: outqueue(q_work);
            break;
        case 128: // Дисбаланс ИКВ
            if(randab(0, 10000, v3) >= 3000)
                next(146);
            break;
        // Прочитать четвертую строку «Упреждение...»
        case 129: // Считывание первых 17 символов T=0,508±0,086 с,
P=0,998.
            inqueue(q_work);
            if(randab(0, 10000, v3) >= 9980)
                k = 2;
            delayt(randab(k*0.422, k*0.594, v6));

```

```

        k = 1;
        break;
    case 130: // Фиксация T=0,4 с
        delayt(0.4);
        break;
    case 131: // Считывание 7 символов T=0,208±0,021 с,
P=0,998.
        if(randab(0, 10000, v3) >= 9980)
            k = 2;
        delayt(randab(k*0.187, k*0.229, v6));
        k = 1;
        break;
    // Ввести новое значение (3 символа )
    case 132: // Нажать кнопку T=0,18±0,03 с P=0,9973.
if(randab(0, 10000, v3) >= 9973)
        k = 2;
        delayt(randab(k*0.15, k*0.21, v6));
        k = 1;
        break;
    case 133: // Затем T=2,4±0,24 с, P=0,995. При ошибке -
повторить T=2,4±0,24 с
        if(randab(0, 10000, v3) >= 9950)
            k = 2;
        delayt(randab(k*2.16, k*2.64, v6));
        k = 1;
        break;
        case 134: // Саккада на 5-ю строку и
считывание первых 17 символов T=0,508±0,086 с, P=0,998.
        if(randab(0, 10000, v3) >= 9980)
            k = 2;
        delayt(randab(k*0.422, k*0.594, v6));
        k = 1;
        break;
    case 135: // Фиксация T=0,4 с
        delayt(0.4);
        break;
    case 136: // Считывание 14 символов T=0,333±0,033 с,
P=0,998.
        if(randab(0, 10000, v3) >= 9980)
            k = 2;
        delayt(randab(k*0.3, k*0.366, v6));
        k = 1;
        break;
    // Ввести новое значение (4 символа )
    case 137: // Нажать кнопку T=0,18±0,03 с P=0,9973.
if(randab(0, 10000, v3) >= 9973)
        k = 2;
        delayt(randab(k*0.15, k*0.21, v6));
        k = 1;
        break;
    case 138: // Затем T=3,2±0,32 с, P=0,995. При ошибке -
повторить T=3,2±0,32 с
        if(randab(0, 10000, v3) >= 9950)
            k = 2;
        delayt(randab(k*2.88, k*3.52, v6));
        k = 1;
        break;
    case 139: // Саккада на 6-ю строку и считывание 18символов
T=0,523±0,087 с, P=0,998.
        if(randab(0, 10000, v3) >= 9980)
            k = 2;
        delayt(randab(k*0.436, k*0.61, v6));
        k = 1;
        break;

```

```

        case 140: // Нажать кнопку T=0,18±0,03 с, P=0,9973.
if(randab(0, 10000, v3) >= 9973)
            k = 2;
            delayt(randab(k*0.15, k*0.21, v6));
            k = 1;
            break;
        case 141: // Затем T=4,2±0,42 с, P=0,995.
if(randab(0, 10000, v3) >= 9950)
            k = 2;
            delayt(randab(k*3.78, k*4.62, v6));
            k = 1;
            break;
        case 142: // Саккада на продолжение 6-й строки и считывание
8символов T=0,235±0,024 с, P=0,998.
if(randab(0, 10000, v3) >= 9980)
            k = 2;
            delayt(randab(k*0.211, k*0.259, v6));
            k = 1;
            break;
        // Ввести новое значение (3 символа)
        case 143: // Нажать кнопку T=0,18±0,03 с, P=0,9973.
if(randab(0, 10000, v3) >= 9973)
            k = 2;
            delayt(randab(k*0.15, k*0.21, v6));
            k = 1;
            break;
        case 144: // Затем T=2,4±0,24 с, P=0,995.
if(randab(0, 10000, v3) >= 9950)
            k = 2;
            delayt(randab(k*2.16, k*2.64, v6));
            k = 1;
            break;
        case 145: outqueue(q_work);
            break;
        case 146: outfac(oper);
            break;
        case 147: seize(f_smes);
            break;
        case 148: // Выгрузка (очистить Смеситель)
nepoladka(sysstime, day);
outfile << "cleaning the mixer." << "<BR>";
random = zagr_smes / m_smes;
gisto += random;
delayt(random);
zagr_smes = 0;
            break;
        case 149: outfac(f_smes);
            break;
        case 150: seize(oper);
            break;
        // В момент окончания цикла гаснет зелёный сигнал кнопки
«Пуск»
        case 151: // Убедиться, что лампочка «Пуск» погасла
if(randab(0, 10000, v3) >= 9500)
            k = 2;
            delayt(randab(k*0.45, k*0.95, v6));
            k = 1;
            break;
        case 152: // Цикл закончился, оператор освобожден
outfac(oper);
            break;
        case 153: { cycle++;
            if(sysstime > time8 && fst) {

```

```

fst = false;
fstShift = q_work->timeq + gisto;
q_task2 = q_error->ci; // количество заданий,
выполненных в условиях очереди
task_total2 = (q_work->ci + q_error->ci); //
общее количество выполненных заданий
q_mq2 = q_error->mq + 1; // запоминаем макс
длину очереди
q_time_er2 = q_error->timeq; // общее время
нахождения в очереди
// вычисляем коэффициент занятости оператора
Z2 = K_zanOp(fstShift, time8);
// вычисляем коэффициент очереди
p2 = p_ocheder(q_task2, task_total2);
// вычисляем среднее время обработки информации
оператором
t_obr_oper2 = fstShift / task_total2;
// вычисляем допустимое время ожидания
информации в очереди
if(t_pr_dop) t_ozh_dop2 = t_pr_dop -
t_obr_oper2;

ostringstream strs;
strs << fixed << setprecision(2);

// Вывод статистики для одной 8-ми часовой
смены
strs << "<BR><b>" << "Statistics for the day #1
8-hour shift:" << "</b><BR>";
Z = " << Z2 << " ." << "<BR>";
strs << "<BR>" << "Operator rate of employment
strs << "Coefficient of queue p = " << p2 << "
." << "<BR><BR>";
strs << "Maximum operator's busy time T_max = "
<< fstShift/60 << " min." << "<BR>";
strs << "Max queue length K_max = " << q_mq2 <<
" ." << "<BR>";
strs << "The average queue length K_sr = " <<
q_error->lm + 1 << " ." << "<BR><BR>";
strs << "Total time in queue = " <<
q_time_er2/60 << " min." << "<BR>";
strs << "Number of queue entries = " << q_task2
<< " ." << "<BR>";
strs << "Average waiting time in queue = " <<
q_time_er2/q_task2/60 << " min." << "<BR><BR>";
if(t_pr_dop) strs << "Allowable time waiting
for information in the queue = " << t_ozh_dop2 << " sec." << "<BR>";
strs << "Average processing time of information
by the operator = " << t_obr_oper2/60 << " min." << "<BR><BR>";
stat = strs.str();
}
need_stat_day = 29;
if(day == need_stat_day && systime > time8 &&
s15th) {
s15th = false;

fstShift = q_work->timeq + gisto;
q_task2 = q_error->ci; // количество заданий,
выполненных в условиях очереди
task_total2 = (q_work->ci + q_error->ci); //
общее количество выполненных заданий
q_mq2 = q_error->mq + 1; // запоминаем макс
длину очереди

```

```

нахождения в очереди                                q_time_er2 = q_error->timeq; // общее время

// вычисляем коэффициент занятости оператора
Z2 = K_zanOp(fstShift, time8);
// вычисляем коэффициент очереди
p2 = p_ochered(q_task2, task_total2);
// вычисляем среднее время обработки информации

оператором
t_obr_oper2 = fstShift / task_total2;
// вычисляем допустимое время ожидания

информации в очереди
if(t_pr_dop) t_ozh_dop2 = t_pr_dop -
t_obr_oper2;

ostringstream strs;
strs << fixed << setprecision(2);

// Вывод статистики для одной 8-ми часовой смены
strs << "<BR><b>" << "Statistics for the day #"
<< need_stat_day << " 8-hour shift:" << "</b><BR>";
strs << "<BR>" << "Operator rate of employment
Z = " << Z2 << " ." << "<BR>";
strs << "Coefficient of queue p = " << p2 << "
." << "<BR><BR>";
strs << "Maximum operator's busy time T_max = "
<< fstShift/60 << " min." << "<BR>";
strs << "Max queue length K_max = " << q_mq2 <<
" ." << "<BR>";
strs << "The average queue length K_sr = " <<
q_error->lm + 1 << " ." << "<BR><BR>";

strs << "Total time in queue = " <<
abs(q_time_er2/60) << " min." << "<BR>";
strs << "Number of queue entries = " << q_task2
<< " ." << "<BR>";
strs << "Average waiting time in queue = " <<
abs
(q_time_er2/q_task2/60) << " min." <<
"<BR><BR>";

if(t_pr_dop) strs << "Allowable time waiting
for information in the queue = " << t_ozh_dop2 << " sec." << "<BR>";
strs << "Average processing time of information
by the operator = " << t_obr_oper2/60 << " min." << "<BR><BR>";
stat += strs.str();
}
if(systemtime > time8*2) {
end = true;
destroy();
}
else {
next(56);
}}
break;
case 154: // запись данных для гистограммы за час работы
if(zhour != 31) {
create(120);
}
break;
case 155: zan_hour[zhour] = q_work->timeq + gisto;
zhour++;
break;
case 156: destroy();
break;
// Перегрев привода

```

```

create(1);

case 157: // По Пуассоновскому закону
    break;
case 158: Pya = poisson(3.47e-5, gen4);
    break;
case 159: if(!Pya || !done)
    next(179);
    break;
case 160: if(time_zav > systime)
    delayt(time_zav - systime);
    break;
case 161: nepoladka(systime, day);
    outfile << "drive overheating." << "<BR>";
    peregreiv = true;
    done = false;
    next(165);
    break;
// Инеродное тело попадает в шнек
case 162: // По Пуассоновскому
    create(1);
    break;
case 163: Pya = poisson(2.31e-5, gen3); break;
case 164: if(!Pya || !done2)
    next(179); break;
case 165: inqueue(q_work);
    inqueue(q_error);
    infac (oper); break;
case 166: // Саккада + считывание первых 18 символов
T=0,523±0,087 с, P=0,998.
    if(randab(0, 10000, v3) >= 9980)
        k = 2;
    delayt(randab(k*0.436, k*0.61, v6));
    k = 1; break;
case 167: // Фиксация + считывание следующих 18 знаков
T=0,4+0,373±0,773±0,037 с, P=0,998.
    if(randab(0, 10000, v3) >= 9980)
        k = 2;
    delayt(randab(k*0.736, k*0.81, v6));
    k = 1; break;
case 168: // Фиксация + считывание следующих 19 знаков
T=0,788±0,038 с, P=0,998.
    if(randab(0, 10000, v3) >= 9980)
        k = 2;
    delayt(randab(k*0.75, k*0.826, v6));
    k = 1; break;
case 169: // Фиксация + считывание следующих 8 знаков
T=0,623±0,022 с, P=0,998.
    outqueue(q_work);
    if(randab(0, 10000, v3) >= 9980)
        k = 2;
    delayt(randab(k*0.601, k*0.645, v6));
    k = 1; break;
case 170: if(!peregreiv)
    next(175); break;
case 171: // Оператор вызывает электрика
    peregreiv = false;
    random = z_op1 = randab(60, 120, v7);
    if(max_buzy < random) {
        max_buzy = random;
    }
    gisto += random; //drive overheating
    delayt(random); break;
case 172: outfacs(oper); break;

```

```

        case 173: // Вызов электрика
delayt(randab(480, 600, v7) - z_op1); break;
        case 174:
            next(178); break;
        case 175: // Оператор занят в течение 410 ± 90 с
            nepoladka(systime, day);
            done2 = false;
            outfile << "a foreign object got into the auger."

<< "<BR>";

            random = z_op2 = randab(320, 500, v7);
            if(max_buzy < random) {
                max_buzy = random;
            }
            gisto += random;
            delayt(random); break;
        case 176: outfac(oper); break;
        case 177: // Вызов электрика
delayt(randab(600, 1800, v7) - z_op2); break;
        case 178: done = true;
            done2 = true;
            outqueue(q_error);
            break;
        case 179: destroy();break;
        // Завис материал.
        case 180: create(1); break;
        case 181: Pya = poisson(1.74e-4, gen2); break;
        case 182: if(!Pya || !done3 || time_zav > systime)
            next(189); break;
        case 183: nepoladka(systime, day);
            outfile << "loose material." << "<BR>";
            loose = true;
            done3 = false;
            time_zav = systime + randab(9*60, 16*60, v4);
            inqueue(q_error);
            infac(oper); break;
        case 184:
            next(82); break;
        case 185: // оператор включает вибраторы в течение
распределенного равномерно интервала времени 15±5 с
            random = randab(10, 20, v2);
            gisto += random;
            delayt(random); break;
        case 186: if(randab(0, 10000, v3) >= 2500)
            next(188); break;
        case 187: // оператор ликвидирует зависание механически
random = randab(30, 180, v3);
            if(max_buzy < random) {
                max_buzy = random;
            }
            gisto += random;
            delayt(random); break;
        case 188: outqueue(q_error);
            outfac(oper);
            loose = false;
            done3 = true; break;
        case 189: destroy(); break;
    }
}
for(int i = 1; i < 31; i++) {
    count[0] += count[i];
}

Z = K_zanOp(q_time, time8*month*2);
p = p_ochedred(q_task, task_total);

```

```

sort(count, count+month+1);
for(int j = 1; j <= month; j++) {
    q_lm_sr += q_lm[j];
}

t_obr_oper = q_time / (task_total * 100);
if(t_pr_dop) t_ozh_dop = t_pr_dop - t_obr_oper;
// Формирование результатов
outfile << stat;
outfile << "<BR><b>" << "Statistics for the work of operators during the
month:" << "</b><BR>";
outfile << "<BR>" << "Operator rate of employment Z = " << Z << " ." <<
"<BR>";
outfile << "Coefficient of queue p = " << p << " ." << "<BR><BR>";
outfile << "Maximum operator's busy time T_max = " << count[30]/60 << "
min." << "<BR>";
outfile << "Maximum uninterrupted operator's busy time T_umax = " <<
max_buzy/60 << " min." << "<BR>";
outfile << "Max queue length K_max = " << q_mq << " ." << "<BR>";
outfile << "The average queue length K_sr = " << abs(q_lm_sr)/month << " ."
<< "<BR><BR>";
outfile << "Total time in queue = " << q_time_er/60 << " min." << "<BR>";
outfile << "Number of queue entries = " << q_task << " ." << "<BR>";
outfile << "Average waiting time in queue = " << q_time_er/q_task/60 << "
min." << "<BR><BR>";
if(t_pr_dop) outfile << "Allowable time waiting for information in the
queue = " << t_ozh_dop << " sec." << "<BR>";
outfile << "Average processing time of information by the operator = " <<
t_obr_oper/60 << " min." << "<BR><BR>";
int temp1, temp2;
phistogram h_hour;
newhist(h_hour, 0, 60, 31, 1, "Operator Employment per hour");
temp1 = zan_hour[0], temp2 = 0;
for(int l = 1; l < 31; l++) {
    temp2 = zan_hour[l];
    zan_hour[l] = ceil(abs(zan_hour[l] - temp1));
    temp1 = temp2;
}

for(int l = 1; l < 31; l++) {
    if(zan_hour[l] > 120) {
        zan_hour[l+1] += (zan_hour[l] - 120);
        zan_hour[l] = 120;
    }
    if(zan_hour[30] > 120) {
        zan_hour[30] = 120;
    }
}

// заполняем гистограмму:
for(int j = 1, tim = 0; j < 31; j++, tim+=2) {
    for(int y = 0; y < zan_hour[j]; y++) {
        tabulate(h_hour, tim);
    }
}

outfile << "The histogram shows the operator's employment during the
working hour." << "<BR>";
outfile << "Employment periods are shown every 2 minutes." << "<BR>";
prnhist(h_hour); // печать гистограммы занятости за час
cout << "Exit\n";
return 0;
}

```

ДОДАТОК Б
Копії слайдів презентації

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА
РЕАЛІЗАЦІЯ МЕТОДІВ ОЦІНКИ
ДІЯЛЬНОСТІ ОПЕРАТОРА
АВТОМАТИЗОВАНОЇ СИСТЕМИ
УПРАВЛІННЯ ТЕХНОЛОГІЧНИМ
ПРОЦЕСОМ

РОЗРОБИВ: ст. групи КНТ-119м

Мохаммед Х.С.

КЕРІВНИК: доцент, к.т.н.

Сердюк С.М.

Рисунок Б.1 – Слайд №1

- **Об'єкт дослідження** – процес управління складними технологічними процесами (ТП).
- **Предмет дослідження** – моделі, методи і засоби оцінки діяльності оператора автоматизованої системи управління технологічним процесом (АСУТП).
- **Мета роботи** – підвищення ефективності та якості діяльності оператора АСУТП шляхом розроблення методу кількісної оцінки діяльності оператора та імітаційного моделювання ТП..



НУ "ЗП", Кафедра програмних засобів

2

Рисунок Б.2 – Слайд №2

Завдання дослідження

- дослідити процеси управління та виготовлення газобетону, та моделі, що їх описують;
- дослідити структуру і функціонування систем імітаційного моделювання (СІМ);
- провести аналіз відомих СІМ;
- розглянути стан досліджень у області кількісної оцінки діяльності оператора;
- розробити комплексну математичну модель, що формалізує діяльність як людини-оператора, так і фізичні процеси, що відбуваються в ході ТП;
- практична реалізація комплексної математичної моделі у вигляді імітаційної моделі процесу управління дозуванням матеріалів газобетону на ЗАТ «Мотор Січ»;
- провести імітаційний експеримент та дослідити кількісні характеристики діяльності оператора.

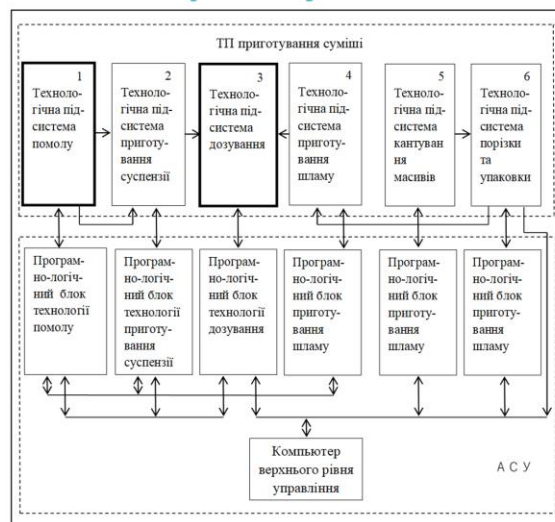


НУ "ЗП", Кафедра програмних засобів

3

Рисунок Б.3 – Слайд №3

Структурна схема технологічних підсистем лінії приготування газобетону

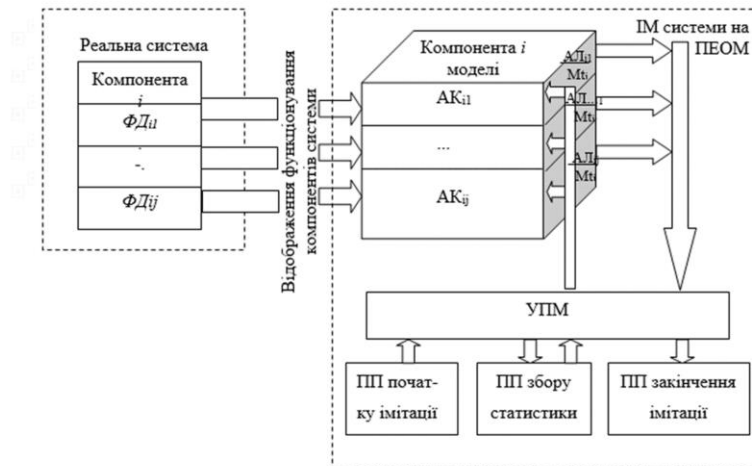


НУ "ЗП", Кафедра програмних засобів

4

Рисунок Б.4 – Слайд №4

Схема переходу від реального об'єкта до його моделі



НУ "ЗП", Кафедра програмних засобів

5

Рисунок Б.5 – Слайд №5

Порівняльний аналіз середовищ імітаційного моделювання

Характеристика	AnyLogic	Arena	Simulink	GPSS World	SIMC
Розробник	The AnyLogic Company	Rockwell Automation Inc.	MathWorks Inc.	Minute Software	НУ «ЗП»
Області використання	Бізнес процес. Соціальна динаміка. Управління процесами і персоналом. Виробництво. Логістика. Стратегічне проектування.	Бізнес процес. Виробництво. Логістика. Склад. Центр обробки викликів.	Медичні дослідження. Системи управління, фінансові розрахунки. Обробка сигналів і зображень.	Виробництво. Системи масового обслуговування.	Виробництво. Системи масового обслуговування.
Мова програмування, спеціалізована мова	Java	SIMAN	MATLAB	GPSS	SIMC
Однотимчасне використання всіх парадигм моделювання	+	+	+	+	+
3D-візуалізація	+	+	+	+	+
Доступ до "ядра"	+	+	+	+	+
Стандартні теоретичні розподіли	29	13	>30	>20	10
Об'єктно-орієнтований підхід	+	+	+	+	+
Документування результатів моделювання	+	+	+	+	+
Підтримка середовища розробки	+	+	+	+	+
Умови ліцензії	Безкоштовна тільки версія Personal Learning Edition	759303 руб.	MATLAB and Simulink Student Suite \$55	+	+



НУ "ЗП", Кафедра програмних засобів

6

Рисунок Б.6 – Слайд №6

Приклад показників кількісної оцінки діяльності оператора

1. Загальний час інформаційного пошуку:

$$t = \sum_{i=1}^n (t_{Fi} + t_{Si}).$$

де t_{Fi} – час зчитування i -го слова (рядка);

t_{Si} – час i -го переміщення погляду (саккади ока).

2. Середній час лексичної ідентифікації слова (зчитування):

$$t_{Ri} = base5 + 15(length - 5) + 40(1 - freq).$$

де $base5$ – середній час (близько 150 мс), необхідний для ідентифікації п'ятибуквеного слова;

$length$ – довжина слова;

$freq$ – значення $\log_{10}$ частоти зустрічаємості слова.

3. Коефіцієнт черги:

$$\rho = \frac{N_q}{n}$$

де N_q – кількість сигналів, оброблених оператором в умовах черги на обслуговування;

n – загальна кількість сигналів, що надійшли до оператора.

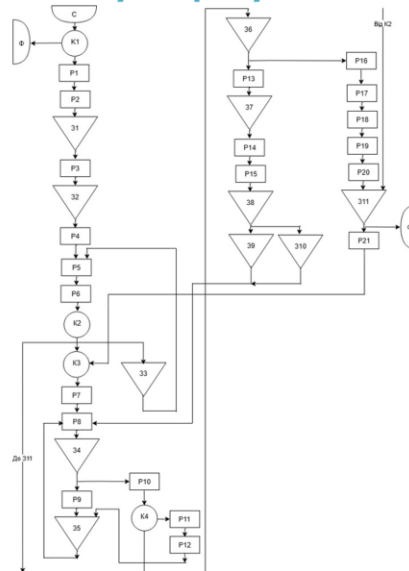


НУ "ЗП", Кафедра програмних засобів

7

Рисунок Б.7 – Слайд №7

Функціональна мережа «Відсутність матеріалу в бункері сухих»



НУ "ЗП", Кафедра програмних засобів

8

Рисунок Б.8 – Слайд №8

Показники типових функціональних одиниць функціональної мережі

Показники безпомилкового виконання робочих операцій

Показник	Операція				
	P1	P2, P4-P6, P8, P9, P11-P17, P19-P21	P4	P7, P18	P10
Ймовірність В1	0,9998	0,9951	0,9979	0,9986	0,9986
Часовий	M,c	1,1	0,48	0,25	0,18
	S,c	0,11	0,13	0,05	0,02

Показники безпомилкового виконання контрольних операцій

Показник		Операція	
		K1	K2, K3, K4
Ймовірність	K11	0,999	0,994
	K00	0,985	0,970
Часовий	M,c	0,28	1,4
	S,c	0,03	0,5



НУ "ЗП", Кафедра програмних засобів

9

Рисунок Б.9 – Слайд №9

Приклад бібліотеки типових функціональних структур

Номер, зміст і схема ТФС	Еквівалентна ТФО	Показник	Розрахункова формула
1	2	3	4
1.Послідовне виконання робочих операцій Р	Робоча операція	Ймовірність безпомилкового виконання	$B = \prod_{i=1}^n B_i^1$, де i – номер ТФО
		Математичне очікування часу виконання (T)	$M(T) = \sum_{i=1}^n M(T_i)$
		Дисперсія T	$D(T) = \sum_{i=1}^n D(T_i)$
		Операція згортки	$P_{\alpha}(T_0) = (F_{p1} * \dots * F_{pn})(T_0)$ символ "*" – означає операцію згортки
2. n-кратне повторення робочої операції одного успішного результату	Робоча операція	Ймовірність безпомилкового виконання	$B = 1 - (1 - B_i^1)^n$
		Математичне очікування T	$M(T) = nM(T_i)$
		Дисперсія T	$D(T) = nD(T_i)$
		Операція згортки	$P_{\alpha}(T_0) = (F_p * F_i)(T_0)$



НУ "ЗП", Кафедра програмних засобів

10

Рисунок Б.10 – Слайд №10

Результати згортки функціональної мережі

- ймовірність безпомилкового виконання

$$B^1 = 0,9411;$$

- математичне сподівання $M(T) = 96,354$ с;

- середнє квадратичне відхилення часу виконання $S(T) = 12,43$ с.



НУ "ЗП", Кафедра програмних засобів

11

Рисунок Б.11 – Слайд №11

Структура системи SIMC



НУ "ЗП", Кафедра програмних засобів

12

Рисунок Б.12 – Слайд №12

Приклад коду імітаційної моделі

```

case 2: // Включити рубильник і дочекатися завантаження системи. T = 30
± 2 c
    delayt(randab(28, 32, v1)); break;
case 3: // Перевірка, чи немає системних помилок. Помилки немає P = 1;
T = 0,4 c
    delayt (0.4); break;
case 4: // Натиснути кнопку «Общий вид» P=0,9973; T=0,18±0,03 c. При
помилці оператора - повторити T=0,18±0,03 c
    if(randab(0, 10000, v3) >= 9973)
        k =2;
        delayt(randab(k*0.15, k*0.21, v2));
        k = 1; break;

```



НУ "ЗП", Кафедра програмних засобів

13

Рисунок Б.13 – Слайд №13

Загальні результати моделювання

Statistics for the work of operators during the month:

Operator rate of employment $Z = 0.09$.
Coefficient of queue $p = 0.16$.

Maximum operator's busy time $T_{\max} = 138.95$ min.
Maximum uninterrupted operator's busy time $T_{\max} = 8.24$ min.
Max queue length $K_{\max} = 6$.
The average queue length $K_{\text{sr}} = 5.27$.

Total time in queue = 1746.93 min.
Number of queue entries = 403.00 .
Average waiting time in queue = 4.33 min.

Average processing time of information by the operator = 1.02 min.

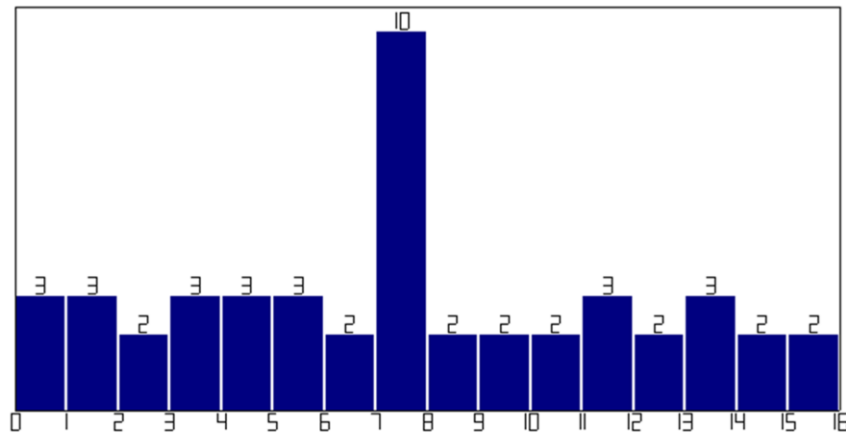


НУ "ЗП", Кафедра програмних засобів

14

Рисунок Б.14 – Слайд №14

Приклад зайнятості оператора першої та другої зміни (15-й день)



НУ "ЗП", Кафедра програмних засобів

15

Рисунок Б.15 – Слайд №15

Висновки

В результаті виконання кваліфікаційної роботи магістра було виконано наступні задачі:

- досліджено процеси управління та виготовлення газобетону, та моделі, що їх описують;
- досліджено структуру і процес функціонування CIM;
- проведено порівняльний аналіз відомих CIM, та обрано CIM, необхідну для вирішення поставлених задач;
- розглянуто стан досліджень у області кількісної оцінки діяльності оператора;
- розроблено комплексну математичну модель, що формалізує діяльність як людини-оператора, так і фізичні процеси, що відбуваються в ході ТП;
- практично реалізовано комплексну математичну модель у вигляді ІМ процесу управління дозуванням матеріалів газобетону на ЗАТ «Мотор Січ»;
- проведено імітаційний експеримент та досліджено кількісні характеристики діяльності оператора технологічної лінії виробництва газобетону на ЗАТ «Мотор Січ».



НУ "ЗП", Кафедра програмних засобів

16

Рисунок Б.16 – Слайд №16

Висновки

- Наукова новизна результатів дослідження полягає у тому, що дістала подальшого розвитку відома комбінована (аналітико-імітаційна) модель процесу автоматизованого управління ТП приготування газобетону шляхом розширення формального опису процесу функціонування оператора для різних видів неполадок в ТП.
- Практична цінність роботи полягає в тому, що на підставі досліджень розроблено імітаційну модель, експерименти з якою забезпечують вирішення задачі підвищення ефективності та якості діяльності оператора АСУТП.



ІНУ "ЗП", Кафедра програмних засобів

17

Рисунок Б.17 – Слайд №17

**Доклад закінчено,
дякую за увагу!**



ІНУ "ЗП", Кафедра програмних засобів

18

Рисунок Б.18– Слайд №18