

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Другий (магістерський)
(рівень вищої освіти)

Розроблення комп'ютерно-інтегрованої системи управління для
координації функціонування підсистем підприємства
(тема)

Виконав:
студент 2 курсу, групи КІТПВм 20-1

Бондаренко Ю.В.
(прізвище, ініціали)

Спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва спеціальності)

Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерно-інтегровані технологічні процеси і виробництва
(повна назва освітньої програми)

Керівник проф. каф. КІТАМ Филипенко О.І
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри КІТАМ

(підпис)

Невлюдов І. Ш.
(прізвище, ініціали)

2021р.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ АКТ _____
Кафедра _____ КІТАМ _____
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність _____ 151 Автоматизація та комп'ютерно-інтегровані технології _____
Тип програми _____ Освітньо-професійна _____
Освітня програма _____ Комп'ютерно-інтегровані технологічні процеси і виробництва _____

(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАМ _____
(підпис)

« _ » _____ 2021р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студентові Бондаренку Юрію Володимировичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Розроблення комп'ютерно-інтегрованої системи управління для
координації функціонування підсистем підприємства _____

Затверджена наказом по університету від 08.11.2021 № 1697 _____

2. Термін подання студентом роботи до екзаменаційної комісії _____

3. Вихідні дані до роботи комп'ютерно-інтегрована система управління для координації
функціонування підсистем підприємства, мова програмування: строго-типізована,
розроблення алгоритма роботи системи, розроблення схема бази даних. _____

4. Перелік питань, що потрібно опрацювати в роботі _____

Аналіз технічного завдання. _____

Вступ. _____

Аналіз систем для координації підсистемами підприємства. _____

Математичне обґрунтування системи для координації функціонування підсистем. _____

Розробка комп'ютерно-інтегрованої системи для координації підсистем. _____

Охорона праці. _____

Висновки. _____

Додатки. _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

Слайди: комп'ютерна презентація у форматі ppt 10-15 слайдів.

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Приміт-ка
1	Аналіз технічного завдання	1.09.2021	Виконано
2	Аналіз завдання	10.09.2021	Виконано
3	Огляд матеріалу за темою роботи	21.09.2021	Виконано
4	Аналіз систем для координації підсистемами підприємства	25.09.2021	Виконано
5	Математичне обґрунтування систем для координації підсистем	3.10.2021	Виконано
6	Розробка комп'ютерно-інтегрованої системи для координації підсистем	25.10.2021	Виконано
7	Оформлення пояснювальної записки	15.11.2021	Виконано
8	Подання роботи на перевірку Інтернет-сервісом Unichesk	6.12.2021	Виконано
9	Подання роботи на рецензію	6.12.2021	Виконано
11	Подання роботи на підпис зав. кафедри	7.12.2021	
12	Подання атестаційної роботи в ЕК	7.12.2021	

Дата видачі завдання 08.11.2021

Студент _____
(підпис)

Бондаренко Ю.В
(прізвище, ініціали)

Керівник роботи _____
(підпис)

проф. каф. КІТАМ Филипенко О. І
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 110 с., 35 рис., 1 дод., 71 джерел.

КООРДИНАЦІЯ ПІДСИСТЕМАМИ, C#, .NET, WINDOWS FORMS КОМП'ЮТЕРНО-ІНТЕГРОВАНА СИСТЕМА.

Об'єкт дослідження – процеси координації та управління підсистем на виробничій лінії.

Предмет дослідження – комп'ютерно-інтегрована система управління для координації функціонування підсистем.

Метою кваліфікаційної роботи є підвищення техніко-економічних показників виробництва, за рахунок зменшення: витрат на обслуговування та відновлення окремих модулів систем управління координації, передбачення поломок у роботі системи, вартості продукції та проценту браку, витрат на відновлення модулів.

Методи дослідження – теоретичні та емпіричні дослідження, системний аналіз, формалізація.

У кваліфікаційній роботі досліджено засоби та алгоритми координації модулів виробництва для виробничих ліній на основі оперативного управління виробництвом (manufacturing execution system).

Проведено аналіз існуючих підсистем та алгоритмів для координації, схем та концепцій. Проаналізовано основні етапи автоматизованих ліній виготовлення та збірки автомобілів. Розроблено математичну модель лінії та порівню з запропонованою концепцією координації підсистемами.

На базі запропонованої концепції проведено розроблення комп'ютерно-інтегрованої систем для координації функціонування підсистем у автомобілебудуванні.

Результати кваліфікаційної роботи оформлено згідно [1-4] та пройшли апробацію на міжнародних конференціях [5-13] та результати викладено у статтях [14-17].

Упровадження розробленої системи можливе у складі автоматизованих систем на виробництві (складальних лініях), в навчальних лабораторіях у освітньому процесі на лабораторних практикумах та практичних заняттях в межах спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології та можуть бути корисними для фахівців, що пов'язані з промисловою автоматизацією.

ABSTRACT

Explanatory note: 110 pp., 35 figs., 1 appendix, 71 sources.

SUBSYSTEM COORDINATION, C #, .NET, WINDOWS FORMS COMPUTER-INTEGRATED SYSTEM.

The object of research – the processes of coordination and management of subsystems on the production line.

The subject of the research is a computer-integrated control system for coordinating the functioning of subsystems.

The purpose of the qualification work is to increase the technical and economic indicators of production by reducing: the cost of maintenance and restoration of individual modules of the coordination management system, anticipation of system failures, cost of production and percentage of shortages, costs of restoring modules.

Research methods - theoretical and empirical research, system analysis, formalization.

In the qualification work the means and algorithms of coordination of production modules for production lines on the basis of operative management of production (manufacturing execution system) are investigated.

The analysis of existing subsystems and algorithms for coordination, schemes and concepts is carried out. The main stages of automated car production and assembly lines are analyzed. A mathematical model of the line and in parallel with the proposed concept of coordination of subsystems is developed.

Based on the proposed concept, the development of computer-integrated systems to coordinate the functioning of subsystems in the automotive industry.

The results of the qualification work are designed according to [1-4] and have been tested at international conferences [5-13] and the results are presented in articles [14-17].

The implementation of the developed system is possible as part of automated systems in production (assembly lines), in training laboratories in the educational process in laboratory workshops and practical classes within the specialty 151 Automation and computer-integrated technologies and can be useful for professionals with industrial flavoring.

ЗМІСТ

Перелік скорочень	7
Вступ	8
Розділ 1 Аналіз систем для координації управління підсистемами підприємства.....	10
1.1 Координація управління підсистемами	10
1.2 Алгоритми координації	12
1.3 Схеми координації підприємства	13
1.4 Координаційна архітектура	15
1.5 Аналіз лінії на основі якої буде розроблено систему.....	17
1.5.1 Основні технологічні процеси виробництва в автомобільній промисловості	18
1.6 Аналіз аналогів комп'ютерно-інтегрованих систем для координації функціонування підсистем підприємства	19
1.7 Висновки до розділу 1	26
Розділ 2 Математичне обґрунтування комп'ютерно-інтегрованої системи управління виробничими процесами на складальній автомобільній лінії підсистемами підприємства	27
2.1 Обґрунтування схеми управління виробничими процесами.....	27
2.2 Розробка математичної моделі управління виробничими процесами складальної автомобільної лінії.....	33
2.3 Висновки до розділу 2	57
Розділ 3 Розробка комп'ютерно-інтегрованої системи управління для координації функціонування підсистем підприємства	58
3.1 Розробка схеми комп'ютерно-інтегрованої системи управління для координації функціонування підсистем підприємства	58
3.2 Вибір середовища розробки.....	60
3.3 Розробка алгоритмічного забезпечення комп'ютерно-інтегрованої системи	69
3.4 Розробка схеми бази даних комп'ютерно-інтегрованої системи.....	71
3.5 Розробка програмного забезпечення комп'ютерно-інтегрованої системи	73
3.6 Розрахунок технічних затрат на програмний засіб	85
3.7 Висновки до розділу 3	87
3.8 Охорона праці.....	87
Висновки	90

Перелік джерел посилань	91
Додаток А Лістинг програми	98
Додаток Б Презентація.....	109

ПЕРЕЛІК СКОРОЧЕНЬ

БД – база даних;

ДВС – двигун внутрішнього згорання;

КІВ – комп'ютерно-інтегроване виробництво;

НДДКР – науково-дослідні та дослідно-конструкторські роботи;

ПЗ – програмне забезпечення;

ПКВ – планування та контроль виробництва;

САПР – система автоматизованого проектування;

ТП – технологічний процес;

ЧПУ – числове програмне управління;

API (application programming interface) – програмний інтерфейс програми;

CAD (ComputerAided Design) – комп'ютерна підтримка проектування;

CAM (Computer Aided Manufacturing) – комп'ютерна підтримки підприємства;

CAQ (Computer Aided Quality) – контроль якості;

IoC (Inversion of Control) – контейнер інверсії залежностей;

IoT (internet of things) – мережа збору даних;

MES (manufacturing execution system) – оперативне управління виробництвом;

PDM (Product Data Management) – система управління даними про виріб;

PRCPSP (Preemption Resource-Constrained Project Scheduling Problem) – обмеження на ресурси, з перериванням на вимоги обслуговування.

ВСТУП

На сьогоднішній день, наша країна перебуває на етапі проходження періоду переходу від автоматизованого виробництва до автоматичного, що змінюється через призму концепції та положень Industry 4.0 – нової промислової революції у виробництві.

Будь-яке автоматизоване виробництво, на сучасному етапі, повинно мати системи, або модулі управління, що слугують для координації підсистем усього підприємства.

Управління в такому випадку відбувається через застосування окремих модулів, що відповідають за моніторинг кожного окремого під процесу.

Для таких процесів розробляються програмні модулі, які об'єднують в собі моніторинг і оперативне регулювання під процесів, що дозволяє тією чи іншою мірою зменшити ризик пошкодження або виведення з ладу усіх ланок виробництва.

Але, як правило, використання високотехнічних програмних засобів, що представлені на ринку компаніями монополістами, є високо вартісним та не завжди задовольняє потреби окремого підприємства. Тому, створення комп'ютерно-інтегрованої системи управління для координації функціонування підсистем підприємства – є актуальним завданням і своєчасним.

На сьогоднішній день кращим варіантом для комп'ютерно-інтегрованих систем вважається програмне забезпечення (ПЗ), яке потребує невисоких вимог в підтримці та супроводі програмного продукту.

Оскільки все більше компаній покладаються на бізнес партнерів або постачальників, координація діяльність через ланцюг постачальників стає критичною.

Швидке зростання інформаційних технологій зараз відкриває для компаній унікальну можливість координувати свою діяльність зі своїми клієнтами та постачальниками для покращення їхньої чутливості.

У даній роботі представлено існуючі підходи координації підприємством, важливі питання координації, такі як архітектура, алгоритми, та схеми координації підсистемами.

Задачі координації функціонування підсистем на основі баз даних і систем управління є актуальною задачею, так як більшість компаній в світі намагаються мінімізувати виробництво браку та не бажаних пошкоджень дорого вартісної техніки.

Дана задача може бути вирішена за допомогою ПЗ в яке буде входити як спостереження за працездатністю певного модуля так і моніторинг параметрів, завдяки чому можна завчасно передбачити не бажану роботу модуля чи помилку. Данні програмні засоби частіше за все виступають в вигляді ієрархії підсистем.

Опираючись на вище сказане, метою кваліфікаційної роботи обрано підвищення техніко-економічних показників виробництва, за рахунок зменшення: витрат на обслуговування та відновлення окремих модулів систем управління координації, передбачення поломок у роботі системи, вартості продукції та проценту браку, витрат на відновлення модулів.

Об'єкт дослідження – процеси координації та управління підсистем на виробничій лінії.

Предмет дослідження – комп'ютерно-інтегрована система управління для координації функціонування підсистем.

Предмет дослідження – комп'ютерно-інтегрована система для координації функціонування підсистем.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати існуючі системи для координації функціонування підсистем підприємства, алгоритми, концепції та схеми координації;
- використовуючи теоретичні та емпіричні дослідження, системний аналіз, формалізацію провести обґрунтування вибору етапів розробки системи;
- розробити комп'ютерно-інтегровану систему для координації функціонування підсистем на виробництві, алгоритм роботи та схему бази даних;
- оформити роботу згідно ДСТУ 3008-15 [1] та керуючись положеннями про вищу освіту [2, 3] та Стандартами освіти та науки України [4].

Упровадження розробленого програмного засобу можливе в комп'ютерно-інтегрованих автоматизованих системах на виробництві для складальних ліній, так і в навчальних лабораторіях у освітньому процесі на лабораторних практикумах та практичних роботах в межах спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології та можуть цікавим для фахівців, що пов'язані з промисловою автоматизацією.

Під час навчання на другому (магістерському) рівні матеріали, що були використані для написання атестаційної роботи апробовано на міжнародних конференціях [5 – 13] та результати висвітлено статтях [14 – 17].

РОЗДІЛ 1

АНАЛІЗ СИСТЕМ ДЛЯ КООРДИНАЦІЇ УПРАВЛІННЯ ПІДСИСТЕМАМИ ПІДПРИЄМСТВА

На сьогоднішній день глобалізації мережевих інтеграцій, отримала велике розповсюдження та має вплив на конкурентоспроможність міжнародного виробництва. Ключовим питанням інтеграції є координація діяльністю розподілених об'єктів таким чином, щоб мати необхідні данні про розміщення цих об'єктів [18,19].

З розвитком всесвітньо монополістичних підприємств виробничі компанії змістили свою увагу від єдиного підприємства до міжнародних виробничих мереж [4]. Тим часом стійкість у мережі є ключовим питанням у реагуванні на жорстку міжнародну конкуренцію та конкурентною перевагою у світовому виробництві [20, 21].

Механізми координації – це план координації діяльності окремих дочірніх ділянок та «отримання максимуму» від далекосяжних заходів діяльність. Крім того, через різні пріоритети виробництва, починаючи від якості, гнучкості, часу доставки, координація об'єктів у мережевому виробництві, очевидно, більша. Допоки узгоджено таку координацію механізми повинні бути адаптовані до конкретних видів діяльності, в яких використовується план, що має враховувати пріоритети мережевого виробництва для розподілу переваг від мережевої інтеграції [22, 23].

1.1 Координація управління підсистемами

Формальні механізми координації пов'язані з базовою структурою організаційних одиниць та діяльність у мережах. Ці механізми втілені в організаційних структурах і процедурах, іншими словами, формальна координація будує структурний підхід до взаємозв'язків між об'єктами.

Крім того, централізація, тобто рівень і положення органу прийняття рішень, і стандартизація процесів і процедур включена до офіційних механізмів координації. Типовий приклад формальних механізмів координації включає обмін типовими документами.

Крім того, численні дослідження вказують на контроль штабом, наприклад, високий рівень контролю підприємства необхідний для впорядкування виробництва [24], тоді як централізований контроль може здійснюватися лише тоді, коли

розпорошені об'єкти узгоджені спільною метою та структурою. Тому концепція контролю демонструє сильні елементи формальних механізмів координації [25, 26].

Неформальні засоби координації можуть бути досягнуті за допомогою взаємодії та спілкування в межах закладів, навчання, розвитку програми та переведення менеджерів. Наприклад, ця техніка – широка передача керівників для створення словесної мережі та входження «емігрантів» до її складу [8].

Як формальний підхід являє собою тонкий механізм нематеріальних підходів, тобто, їх неможливо легко встановити в організації, їх успіх залежить від постійності відносин між об'єктами [27].

Так само спілкування, як засіб неформальної координації, між приймаючою установою та штабом має вирішальне значення для усієї офшорної (наукоємної) діяльності [18].

Будь яка задача координації представляється в вигляді ієрархічних систем. Загальна схема координації у дворівневій системі зводиться до такого, що елементи передаються в центр набору варіантів роботи.

Кожний варіант є векторним показником елемента, припустимим, з погляду його локальних обмежень. На основі отриманих від елементів варіантів центр формує план, оптимальний з погляду всієї системи [2]. Цей план передається елементам і деталізується ними [28 – 30].

Проте, під час моделювання складних систем неможливо врахувати достатньо велику кількість реальних факторів, оскільки, це приведе до ускладнення системи.

Тому, в модель доводиться вводити лише обмежену кількість таких факторів, які з тих чи інших міркувань вважають найважливішими. При цьому можливі два підходи [31, 32].

Невраховані в описі моделі фактори можна вважати абсолютно неважливими і повністю їх ігнорувати під час прийняття рішень з використанням цієї моделі.

З іншого боку, за другим підходом можна явно не вводити неважливі фактори в математичну модель, але враховувати їх вплив, припустивши, що реакція моделі на ту або іншу дію (вибір альтернативи) може бути відома лише наближено або нечітко.

Такі структури є ієрархічними і до переваг автоматизованого управління, в якій на нижньому рівні знаходиться велика кількість нескладних задач, а на вищих рівнях – невелика кількість складних задач, належать зниження загальної вартості опрацювання інформації в системі та підвищення пропускну здатності хост-машин в мережі, а також стійкість до завад [33].

Критичні для системи функції продовжують виконуватися локальними системами керування при виході з ладу хост-машини або лінії зв'язку. Загальна задача оптимального керування ієрархічними системами, зазвичай, ставиться як статична оптимізаційна задача, тобто розглядається задача функціонування на достатньо великих проміжках часу, під час яких динамікою перебігу процесів можна знехтувати.

Таким чином, можна сказати, що оптимально побудувати ієрархічну дворівневу систему на формальній координації, зумовлено можливістю реалізувати централізацію, тобто рівні і положення органу прийняття рішень, і стандартизацією процесів і процедур включена до офіційних механізмів, а також декомпозицією модулів що дає більш точну оцінку ризиків.

1.2 Алгоритми координації

Існує два види алгоритмів координації систем управління координації підсистем підприємства: ітеративні; без ітеративні.

У наявних сьогодні ітеративних процедурах (алгоритм Данцига-Вульфа, алгоритм Корнаї-Ліптака, методи, що ґрунтуються на введенні функції Лагранжа або її різних модифікаціях, алгоритми оптимізації, узагальнена схема ітеративних алгоритмів Алієва та Ліберзона) оптимальне рішення визначається в процесі ітеративного обміну інформацією між центром і елементами [34, 35].

Хоча є кілька варіантів реалізації подібних алгоритмів, найбільш поширеним являється алгоритм:

а) починаючи з можливого вирішення скороченої основної програми, необхідно сформулювати нові цільові функції для кожного під завдання, так що під завдання пропонуватимуть рішення, які покращують поточну мету основної програми;

б) під завдання повторно вирішуються з урахуванням нових цільових функцій. Оптимальне значення для кожної задачі пропонується головній програмі;

в) основна програма включає один або всі нові рішення, які було згенеровано рішеннями під задач на основі відповідної здатності цих рішень поліпшити мету вихідної задачі;

г) основна програма виконує x ітерацій симплекс-алгоритму, де x – кількість включених стовпців;

г) якщо мету досягнуто – перейдіть до кроку а. Інакше – продовжіть;

д) головна програма не може бути додатково покращена будь-якими новими рішеннями з під задач, тому – повернення.

На кожному кроці ітеративного процесу розв'язуються локально-оптимальні задачі елементів і координуюча задача центру. За методами координації, побудованими на основі без ітеративних алгоритмів, координація здійснюється в результаті одноразового обміну інформацією між рівнями.

Переважно без ітераційні алгоритми зводяться до побудови множини ефективних рішень для організаційних ієрархічних систем [36].

Таким чином, можна зробити висновок, що ітеративний алгоритм оптимальний за критерієм конкретизації задач, та їх вирішенням на кожному кроці управління та моніторингу.

1.3 Схеми координації підприємства

Типовим прикладом складних систем може бути КІВ (комп'ютерно-інтегроване виробництво), де необхідно координувати різні інженерні завдання та обмінюватися даними між різними спеціалізованими інструментами. Інформаційний потік підприємства можна класифікувати на дві частини: технічні дані та управлінські дані [19].

Системи управлінської частини включає переважно додатки для інжинірингу бізнесу, фондові управління, але також планування та контроль виробництва (ПКВ).

Технічний трек складається, наприклад, із застосуваннями для підтримки дизайну продукту (CAD), програмування з числовим управлінням машини (CAM) або контроль якості (CAQ).

Координація підсистем у КІВ має вирішальне значення для забезпечення послідовних даних про продукцію (наприклад, специфікацій) у додатку до CAD, PDM [37].

Оскільки кожна підсистема зберігає дані в окремому приватному сховищі та використовує власну модель даних з урахуванням спеціальностей відповідних додатків, існує багато залежностей між підсистемами.

Приклад наведено на рис 1.1: працівник відділу науково-дослідних та дослідно-конструкторських робіт (НДДКР) інженерної компанії змінює дизайн продукту X (наприклад, через деякі фатальні збої) у системі САПР, що призвело до оновлення специфікації (матеріалу) у базовій системі баз даних [38, 39].

Процес координації слід почати, відстежуючи зміни, наприклад, до КПП та до інженерної системи бізнесу до оновлення виробничих планів та внесення змін до процесу закупівлі деталей.

Навіть, якщо співробітник НДДКР виконує свої операції локально і не знає про операції, які мають бути виконані чітко та привести систему до чітко визначеного стану.

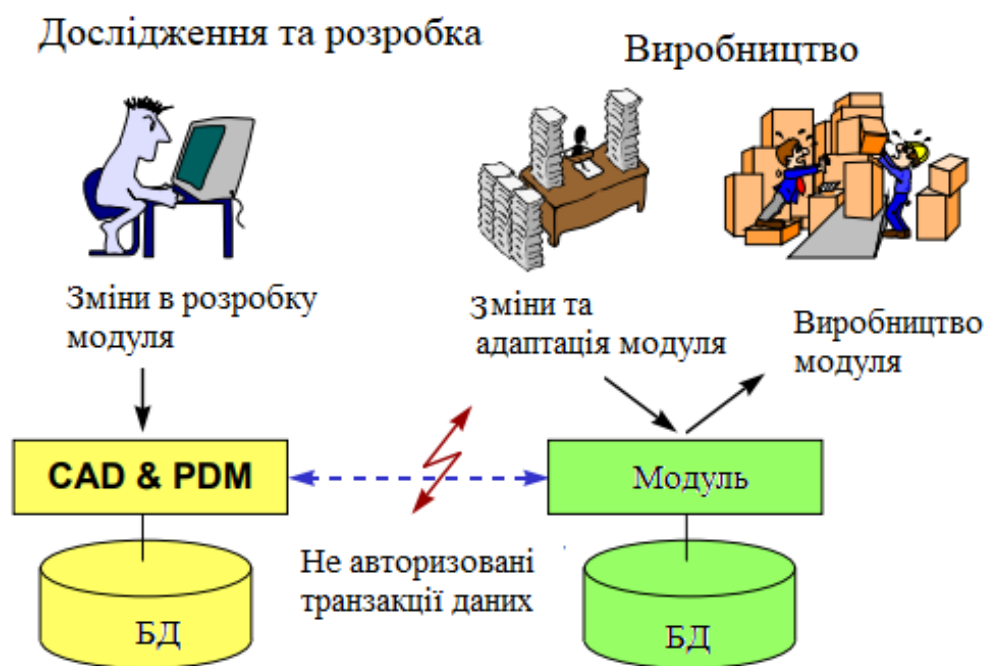


Рисунок 1.1 – Координація підсистем у КІВ [32]

Ще важливіше, якщо зміни виконуються не тільки в відділу НДДКР, а також у виробничому відділі (наприклад, для адаптації певних властивостей продукту), що веде до одночасних процесів координації.

Сам процес координації має бути застосований в залежності від складених систем (наприклад, загальносистемну узгодженість даних про продукт), гарантуючи, що операції у відповідних підсистемах є такими, що виконуються з певними гарантіями виконання [40].

У КІВ були докладені різні зусилля, щоб подолати проблему застосування (радше базових) залежностей: уникнення суперечностей, що призводить до повторення даних про продукт, зберігання під контролем гетерогенних та автономних прикладних систем.

Перший підхід – зміна пріоритету або мети збереження місцевості автономності прикладних систем-учасників та інтегрування локальних програм у глобальні.

Тоді, доступ до глобальних даних можливий лише через інтегровані програми, де глобальна цілісність забезпечується традиційним способом [2].

Подальшим розвитком підходу є запровадження додаткової бази даних поза існуючими прикладними системами додатково копіюючи глобальні дані [3].

Тоді, глобальні обмеження цілісності можуть бути визначені для глобальних даних інтеграційної бази даних [40].

Однак, додаткові накладні витрати слід розглядати як реплікацію, яка призводить до додаткових витрат.

Синхронні механізми спілкування також можуть використовувати для збереження послідовності даних після локальних операцій та забезпечення глобальної цілісності [10, 17].

Замість накладних витрат на централізовану базу даних, необхідно докласти додаткових зусилля, визначивши між прикладну комунікацію. Це призводить до досить вільного взаємозв'язку прикладних систем і не дозволяє підтримувати складні координаційні процеси в складених системах [41, 42].

Враховуючи описані факти, можна визначити, що система повинна бути досить простою та не мати надмірних зв'язків між модулями. Можна сказати, що оптимальне рішення розроблення модулів незалежних один від одного, та запровадити додаткову БД для логування збоїв.

1.4 Координаційна архітектура

Автономія підсистем додатків призводить до випадків де порушуються локальні операції в межах однієї підсистеми глобальної послідовності, що вимагають координаційні процеси у складених системах [43, 44].

Підхід базується на агентах, «сидячи» поверх підсистем і надсилаючи повідомлення про діяльність місцевої підсистеми координатору, який викликає глобальні координаційні процеси.

На практиці найпоширеніший підхід до гарантування загальносистемної послідовності у всіх додатках підсистеми – делегування відповідальності користувачеві, який повинен забезпечити відповідні дії, що виконуються у всіх окремих підсистемах, [45, 46].

Налаштовуються навіть спеціально розроблені програмні засоби «інформаційного насосу», які дають змогу користувачеві перенаправляти і перекладати дані та операції між додатками підсистем.

Однак, більшість із цих інструментів пов'язують лише дві різні системи, але не є таким, що автоматично виявляють операції, що порушують глобальну послідовність, а також не виконують автоматично операції, необхідні для відновлення загальносистемної послідовності.

Дана архітектура координації може бути основою для автоматичного відновлення загальносистемної послідовності, але після детальнішого обговорення розширення гнучкої обробки несправностей та їх реалізації не може гарантувати виконання «поверх» існуючої вже архітектури.

Обов'язковою умовою цього є – розширення підсистем для координації спеціальними агентами, які надають їх з функціональності бази даних. Загальну архітектуру координації наведено на рис. 1.2.

Мета координації полягає у забезпеченні залежностей між прикладними системами, хоча дані змінюються локально – за допомогою відповідних програм [47, 48].

Як бажана автономність додатків дозволяє такі локальні операції, що врешті решт порушують глобальну послідовність, необхідно надати для кожного спеціального агента підсистема для виявлення локальних операцій.

Моніторинг локальних операцій, що порушують глобальну послідовність, це передумова для подальших процесів координації ініційованих глобальних координаторів.

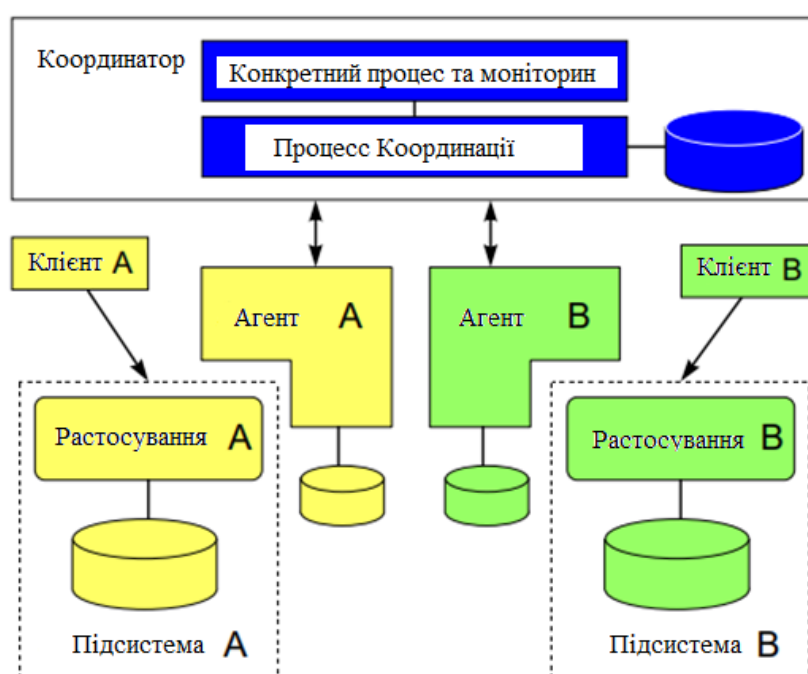


Рисунок 1.2 – Загальна архітектура для координації автономних підсистем [32]

У межах цього координаційні процеси, операції з відповідними особами підсистеми (які опосередковуються відповідними агентами) виконуються для відновлення узгодженості [49, 50].

Гарантії, реалізовані в процесі координації, сильно залежать від послуг базових агентів, і вони також залежать від готовності підсистем до співпраці. Інші підсистеми/агенти можуть мати різні властивості [51].

Таким чином можна сказати, що кожен модуль повинен мати свій ModulProvider для постачання функціонування модуля та його моніторингу, що дає змогу аналізувати виробництво та параметри модуля в кожен момент часу.

1.5 Аналіз лінії на основі якої буде розроблено систему

Було проведено аналіз [40] сфер високотехнологічних та найбільш розвинутих виробництв було визначено, що автоматизоване автомобілебудування – одна з найперспективніших галузей, що може бути використана для розроблення комп'ютерно-інтегрованої системи управління координації функціонування підсистем підприємства [41].

Отже, розроблення програмного засобу буде побудована на прикладці крупновузлової складальної лінії легкових автомобілів.

Типова лінія збирання готових автомобілів складається з виробничих підрозділів з виготовлення різних частин та компонентів комплектуючих.

Компоненти-комплектуючі автомобіля можуть бути виготовлені на головному підприємстві або замовленні в окремих корпоративних організаціях.

Автоматизоване виробництво автомобілів бере свій початок приблизно у 19 сторіччі. У Північній Америці, Європі та після Другої світової війни в Японії автомобільна та транспортна промисловість сконцентрована у кількох корпораціях, що мають свої філії, які виконують складальні операції у Південній Африці, Африці та Азії для збуту продукції на цих ринках.

Міжнародна торгівля готовими автомобілями зростає, починаючи з 1970-х років, а торгівля через філії вихідним обладнанням та запасними частинами до автомобілів має ще більше значення.

Карта автомобільного виробничого процесу представлено на рис. 1.3.

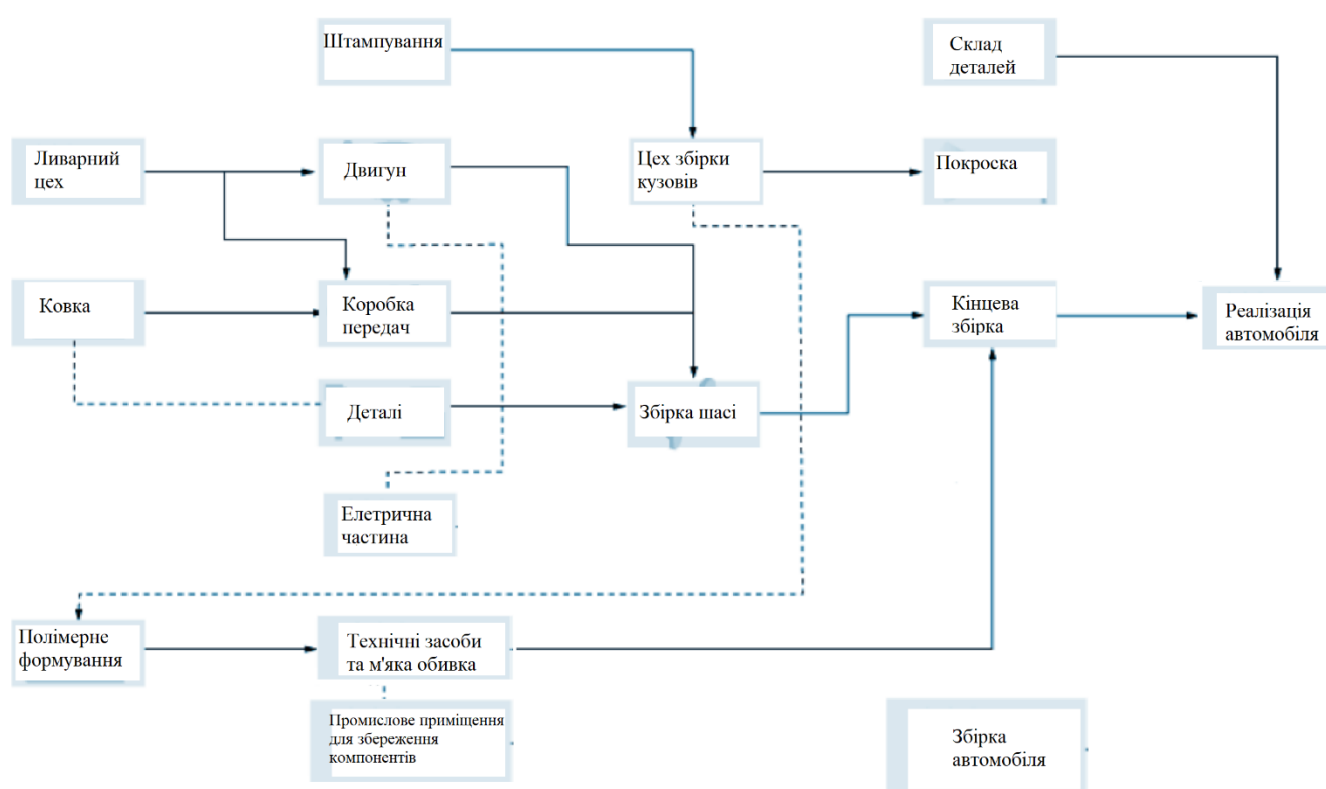


Рисунок 1.3 – Карта автомобільного виробничого процесу

1.5.1 Основні технологічні процеси виробництва в автомобільній промисловості

Одним з фундаментальних технологічних процесів у автоматизованому автомобілебудуванні є лиття.

Технологічний процес (ТП) лиття включає наступні етапи [52]:

- виготовлення зразка-форми бажаного виробу;
- виконання форми, шляхом заливання піску та сполучного матеріалу навколо зразка, а також її ущільнення та трамбування;
- зняття зразка, вставка ливарного стрижня та складання форми;
- розплавлення та переробка металу в печі;
- заливання розплавленого металу у форму;
- охолодження металевого лиття;
- видалення форми та ливарного стрижня шляхом «вибивання» (для малих виливків) та шляхом вібраційних сит (струшування) або шляхом водоструминного промивання;
- видалення зайвого металу (наприклад, метал у ливнику – ливниковому каналі для заливки розплавленого металу у форму) та обпаленого піску з кінцевої виливки (очищення) шляхом продування металевим порошком, ручного сколювання та шліфування [53 – 54].

Металеве лиття виробничого типу є типовим процесом автомобілебудування. Металеве лиття застосовується в автомобільній промисловості для блоків циліндрів двигунів, головок циліндрів та інших деталей. Існує два основних типи металевого лиття: сіро-чавунне лиття та еластично-чавунне лиття [55 – 56].

Вище згадані процеси являються собою більш декомповані процеси та представляють не повний перелік всіх процесів, як головні етапи виготовлення автомобіля можна виділити наступні умовні модулі виробництва:

- збірки кузова;
- модуль збірки двигуна внутрішнього згорання (ДВС);
- модуль збірки салону;
- модуль збірки шасі;
- модуль з'єднання кузова та шасі;
- модуль контролю якості.

Вище згадані модулі будуть вміщувати в себе від 5 до 8 параметрів за якими оператор матиме можливість спостерігати.

1.6 Аналіз аналогів комп'ютерно-інтегрованих систем для координації функціонування підсистем підприємства

На даний момент, із подібних систем існує декілька аналогів ПЗ для координації функціонування підприємством, основним аналогом який буде розглядатися це – MES (manufacturing execution system) оперативне управління виробництвом, даний програмний засіб була розроблений компанією «Рарус» [42 – 44].

Продукт призначений для використання в наступних підрозділах виробничого підприємства:

- планово-економічний відділ;
- виробничо-диспетчерський відділ;
- відділ матеріально-технічного забезпечення;
- відділ технічного контролю;
- відділ кадрів та заробітної плати;
- відділ головного конструктора;
- відділ головного технолога;
- виробничі підрозділи [42 – 44].

Продукт дозволяє:

- спеціалістам планово-економічного відділу – планувати обсяги виробництва, контролювати терміни виконання плану, контролювати обсяги відрядної заробітної плати робітників;
- спеціалістам виробничо-диспетчерського відділу – моделювати виробничу програму, формувати маршрути виробництва, здійснювати оперативний контроль виробництва, контролювати терміни виконання технологічних операцій;
- спеціалістам відділу матеріально-технічного забезпечення – отримувати графіки потреби у матеріалах та напівфабрикату, фіксувати відпустку матеріалів у виробництво;
- спеціалістам відділу технічного контролю – враховувати брак під час виготовлення виробів, контролювати показники якості та їх зміну на всіх етапах виробництва;
- менеджерам із продажу – отримувати інформацію про здійсненість замовлення у зазначені покупцем терміни;
- начальникам виробництв – отримувати інформацію про заплановані операції, призначати виконавців, контролювати виконання технологічних операцій;
- робітникам – оперативне отримання планів робіт зміни, відбивати виконання операцій [42 – 44].

Поліпшення техніко-економічних показників виробництва під час впровадження системи оперативного управління виробництвом досягається за рахунок:

- скорочення часу виконання замовлень шляхом раціонального завантаження обладнання, зниження обсягів непродуктивної праці, зменшення простоїв та часу зберігання матеріалів, що підлягають обробці;
- вдосконалення системи обліку та звітності, спрощення та впорядкування виробничого документообігу.

Зовнішні та внутрішні виробничі замовлення призначені для оформлення кількісних та календарних потреб у продукції та напівфабрикатах. Терміни виконання виробничих замовлень розраховуються з фактичних можливостей виробництва, і коригуються при відображенні виконання операцій [42 – 44].

Основними завданнями підсистеми оперативне планування виробництва є:

- формування виробничої програми;
- формування розкладу відповідно до заданих критеріїв оптимізації, для виконання заданого обсягу робіт з урахуванням обмежень та виробничої ситуації, що склалася, на момент розрахунку [42 – 44].

Контроль термінів виконання контрольних значень (замовлень, маршрутних листів, технологічних операцій).

Оперативний контроль виробництва та реагування на відхилення від заданого графіка, перерахунок розкладу для відображення управлінських рішень.

Для вирішення завдання побудови розкладу в «MES Оперативне управління виробництвом» застосовується математичний апарат теорії розкладів, зокрема евристичний алгоритм, що поєднує жадібні стратегії та стратегії обмеженого перебору, елементи теорії графів. Розглядається окремий випадок побудови розкладу виконання робіт з урахуванням відносин попереднього та обмежень на ресурси, з перериваннями обслуговування вимог (PRCPSP – Preemption Resource-Constrained Project Scheduling Problem).

Алгоритм дозволяє розміщувати технологічні операції на осі часу робочих центрів без дискретних інтервалів з урахуванням таких обмежень:

- умови передування технологічних операцій;
- доступність основних робочих центрів.

Виробнича програма дозволяє згрупувати джерела даних планування виробництва та на підставі цих даних виконати розв'язування, сформувати партії запуску, маршрути виробництва та потреби операцій маршрутів. При формуванні виробничої програми враховуються такі параметри, як: специфікації, технологічні карти, кратність, мінімальні партії запуску, умови попередження [34].

Документ «Виробнича програма» не складає розклад завантаження робочих центрів, а лише формує маршрути виробництва. Результатом документа є безліч операцій, послідовностей виконання, залежностей, нормативного часу їх виконання.

Виробнича програма не містить горизонту планування, а лише додає до поточних невиконаних операцій нову велику кількість технологічних операцій з умовами виконання (у тому числі директивні терміни виконання). Механізм побудови розкладу розподіляє безліч операцій і прагне розмістити операції з мінімально можливим часом відхилень від директивних [32].

Об'єктом планування є технологічні операції виробничої програми. Технологічні операції виробничої програми та залежності між ними формуються на підставі даних джерел планування з урахуванням технології виготовлення. Технологічні операції виробничої програми містять необхідну інформацію про нормативні терміни виконання, залежність від інших операцій, а також іншу аналітичну інформацію (замовлення на виробництво тощо).

Механізми оперативного планування виробництва дозволяють розмістити безліч технологічних операцій виробничої програми на графіку завантаження робочих центрів відповідно до графіків роботи робочих центрів, а також з урахуванням обраних обмежень та критеріїв оптимізації.

Розподіл технологічних операцій провадиться з урахуванням встановлених пріоритетів [18,29].

Варіанти розподілу технологічних операцій, що підтримуються:

- пряме планування – від перших до останніх операцій у технологічному маршруті;
- зворотне планування – від останніх до перших операцій у технологічному маршруті.

У разі виникнення браку, поломки обладнання, відставання від планових показників часу виконання операцій відбувається відхилення від графіка виробництва, для актуалізації даних з виробництва виробу необхідно здійснити перепланування графіка виробництва.

Основна мета диспетчеризації – забезпечення та підтримання нормального перебігу виробничого процесу без збоїв і порушень прийнятого для підприємства виробничого циклу, відстеження виконання планів випуску готової продукції всіх стадіях її виготовлення.

Формування здійснимого оперативного плану виробництва неможливо без зворотного зв'язку про хід виконання запланованих операцій.

Основними завданнями підсистеми диспетчеризації виробництва є:

- формування завдань виконання технологічних операцій;
- оперативне відображення ходу виконання операцій;
- контролює терміни виконання технологічних операцій;
- формування документів випуску;
- формування змінно-добових завдань;

Оперативний план виробництва є джерелом даних призначення виконання робіт робочим центрам, виходячи з яких на короткостроковий період (зміна, доба) формується документ змінно-добове завдання. Склад робіт, час виконання, робочі центри, потреби може бути скориговані майстром зміни. Документ фіксує запланований час та є підставою для виконання [12, 8].

Інструмент призначений для навігації за призначеними для виконання операціями та управління ходом виробництва в реальному режимі часу.

З урахуванням встановлених відборів (у тому числі передбачена робота зі сканерами штрих-кодів), для кожної операції відображається необхідна інформація:

- позначення технологічної операції з прив'язкою до робочого центру та нормативним часом виконання;
- семафорна система стану виконання, часу виконання.

Інструмент дозволяє змінювати стан виконання (початок, перерву, завершення тощо), відображати завершення чи відмову від виконання операції, здійснювати контроль виробництва. Передбачено можливість групової обробки операцій. Зміна станів виконання операцій відбивається на графіку завантаження обладнання.

Операції, не виконані з будь-яких причин, при виконанні процедури перепланування знову розміщуються на графіку завантаження обладнання і плануються до виконання [7,54].

Відображення факту виконання операції при завершенні або відмові від виконання фіксується наступна інформація:

- фактичний термін виконання;
- виконавець (передбачена можливість відображення кількох виконавців із зазначенням КТУ) та сума до нарахування;
- вхідні/вихідні ресурси.

Інформація заповнюється автоматично та може бути змінена.

Технологічний процес, прописаний у маршрутних картах, може завжди відбуватися у одному цеху. Якщо виконується кооперація всередині заводу, тобто. деталь віддається на обробку до іншого його підрозділу, цехом-здавачем виписується вихідне робоче вбрання, яке разом з деталлю віддається до цеху-виконавця.

Доплатні наряди – це наряди, якими оплачуються роботи, які стосуються виробництва основний продукції, тобто. до виготовлення, на які виписуються маршрутні картки, у тих випадках, коли виникає необхідність у непередбачених маршрутною карткою технологічних операціях.

Робочі наряди – це наряди, якими оплачується робота, яку не передбачено виписки маршрутних карт.

У системі реалізовано можливість настроювання регламентних завдань для розсилки звітів відповідальним особам.

Можливе використання різних налаштувань звітів при формуванні звітів, що розсилаються, різним відповідальним.

Для відображення оперативного випуску продукції та вироблення по відрядній заробітній платі у конфігурації Управління виробничим підприємством, передбачені документи майстра зміни:

- «Звіт майстра зміни»;
- «Звіти про склад зміни»;
- «Відрядний наряд».

Документи прикладних рішень є повністю ідентичними і дозволяють сформувавши регламентні документи для відображення в обліку.

Документ «Звіт майстра зміни» призначений для оперативного відображення відомостей: випускати продукцію за зміну; кількість витрачених протягом зміни матеріалів.

Для оперативного обліку часу, відпрацьованого співробітниками підприємства, призначено документ «Звіт про склад зміни» [58].

Документ «Звіт про склад зміни» дозволяє відобразити дані про співробітників, які працювали у зміні, а також тих, хто мав працювати, але був відсутній. Для кожного співробітника можна вказати ознаку явки, час явки та фактично відпрацьований час.

Якщо співробітник був відсутній, то можна вказати причину неявки та заміщуючого співробітника.

Документ призначений для реєстрації фактичного відрядного виробітку в управлінському, бухгалтерському та податковому обліку витрат, а також для цілей обчислення заробітної плати: з одного боку, у документі описуються виконані за нарядом роботи; з іншого боку, визначається склад виконавців (наприклад, бригада робітників), які здійснили роботу.

Управління даними про склад продукції та напівфабрикатів, про маршрути проходження виробів по виробничих підрозділах та складах - це важливий елемент процесу управління виробництвом, за допомогою якого здійснюється організація та планування виробництва [45].

Основним засобом планування матеріальних витрат за виготовлення продукції є специфікації.

Специфікація містить перелік матеріалів та інших складових, що використовуються під час виробництва виробу. Всі описи матеріалів, напівфабрикатів, комплектуючих та готових виробів зберігаються в інформаційній базі у загальному номенклатурному списку, що спрощує процес складання специфікацій та, зрештою, полегшує планування та управління.

Для кожної комплектуючої, що входить до специфікації, можна вказати, щодо чого буде виконуватися розрахунок її споживання (норматив) – щодо кількості готової продукції або кількості основної сировини. Як основна сировина вказується одна з комплектуючих, що входять до складу специфікації.

У специфікації може бути враховано утворення відходів у результаті випуску виробу. Зворотні відходи підлягають реєстрації як матеріали, придатні до використання діяльності підприємства.

Можна керувати функціональністю специфікації шляхом увімкнення/вимкнення видимості закладок та реквізитів у табличних частинах у формі специфікації. Передбачено два рівні налаштування функціональності специфікацій.

Перший рівень визначається налаштуваннями, які задаються для конкретного користувача. Налаштування, задані для користувача, будуть використовуватися для нього за умовчанням при створенні нових або відкритті існуючих специфікацій.

Другий рівень налаштування – це керування функціональністю конкретної специфікації. Управління може здійснюватися користувачем у процесі роботи зі специфікацією.

Для швидкого заповнення табличних частин специфікації можна скопіювати або додати дані з аналогічних табличних частин інших специфікацій.

Дані всіх закладок однієї специфікації можна скопіювати в іншу специфікацію або використовувати для заповнення інформацію про фактичні витрати матеріалів.

Комплектуючий або готовий виріб може складатися з інших елементів списку номенклатури. У свою чергу, воно може бути включене до специфікації іншого виробу.

Специфікація може передбачати кілька технологічних операцій на кількох робочих центрах. Тому мінімальним елементом опису виробничого процесу конфігурації є технологічна карта виробництва, що представляє собою опис технологічної операції, що виконується в робочому центрі [32].

Для цілей оперативного планування виробництва весь технологічний процес можна як послідовної комбінації технологічних карт виробництва – описів наборів технологічних операцій.

Склад об'єктів для опису структури виробів запозичений на підставі типової конфігурації «Управління виробничим підприємством для України», редакція 1.3 та при спільному використанні з облікової інформації синхронізується.

У конфігурації реалізовано рольовий підхід до організації робочих місць. Набір ролей, достатній для виконання користувачами певних функцій, які відповідають посадовим обов'язкам, формує профіль користувача.

Призначення даної підсистеми – прийом та передача інформації з/в облікову або управлінську систему підприємства.

Основні функції, реалізовані у підсистемі:

інтеграція з типовою конфігурацією «Управління виробничим підприємством для України», редакція 1.3 для розширення функціоналу в частині оперативного управління виробництвом на поопераційному рівні.

Даний програмний засіб є багатофункціональним та має велику ціну яка становить 99000 грн за одну ліцензію на використання.

1.7 Висновки до розділу 1

Проаналізували існуючі системи для координації функціонування підсистем підприємства, алгоритми, концепції та схеми координації.

Виявили що для даного виду ПЗ оптимально побудувати ієрархічну дворівневу систему на формальній координації, це зумовлено можливістю реалізувати централізацію і стандартизацією процесів і процедур включених до механізмів, а також декомпонувати модулі що дає більш точну оцінку ризиків. Декомпозиція в свою чергу призведе до спрощення системи не утворення простих зв'язків між модулями.

Кожен модуль в свою чергу повинен мати своє джерело даних модуля та окремий інтерфейс , що дає змогу аналізувати виробництво та параметри модуля в кожен момент часу.

Приймаючи до уваги виконаний аналіз необхідно виконати:

- математичне обґрунтування концепції координації;
- розробку алгоритм роботи комп'ютерно-інтегрованої системи;
- розробку схему БД;
- розробку інтерфейсу системи.

РОЗДІЛ 2

МАТЕМАТИЧНЕ ОБҐРУНТУВАННЯ КОМП'ЮТЕРНО-ІНТЕГРОВАНОЇ СИСТЕМИ УПРАВЛІННЯ ВИРОБНИЧИМИ ПРОЦЕСАМИ НА СКЛАДАЛЬНІЙ АВТОМОБІЛЬНІЙ ЛІНІЇ ПІДСИСТЕМАМИ ПІДПРИЄМСТВА

Математичні моделі є абстрактними сутностями, що використовує математичний мову, що позначається як поведінка системи.

Математична модель в основному використовуються в інженерних описах процесів таких як: фізичні, біологічні, і електромеханіці.

Математичні моделі можуть мати багато форм, в тому числі не обмежені динамічні системи, статистичні моделі, різні еквівалентності, або гра теоретичних моделей [55].

Існують шість основних груп змінних:

- змінні рішень;
- вхідні змінні;
- змінні станів;
- екзогенні змінні;
- дискретні змінні;
- вихідні змінні [56].

Безсумнівно вони можуть бути множинними параметрами їхнього типу, що є як правило, репрезентовані векторами.

Проблема математичного моделювання класифікується в моделях чорного ящика або білого ящика, в залежності від того, як багато дослідних відомостей є наявними в системі.

Модель чорного ящика є системою, яка не є першою відомою доступною [57].

Модель білого ящика (також називається скляною коробкою або чистим боксом) є системою, де все необхідне для отримання інформації є наявним [55].

2.1 Обґрунтування схеми управління виробничими процесами

За класичною схемою виробнича лінія на якій виготовляється будь-який продукт має декілька модулів, не залежно від того, складальні модулі чи модулі обробки заготовок, кожен модуль в свою чергу регулюється певною програмою яка

запрограмована в станок з числовим програмним управлінням (ЧПУ) якщо модуль обробки та в програмний блок маніпулятора якщо це складальна лінія [57].

Кожен регулятор (монітор з параметрами) переглядається певним працівником який оцінює точність, працездатність та коректність роботи модуля.

При русі заготовки по конвеєру з модуля $M1$ в модуль $M2$ взаємозв'язані параметри заготовки можуть бути втрачені та оператор 1 та оператор 2 просто не нададуть значення мінімальним відхиленням які можуть призвести до подальшого браку, або в найгіршому сценарії виведення зі строю модуля.

Як наведено на рис. 2.1 виробництво може складатися з декількох модулів P наприклад з 5 або більше.

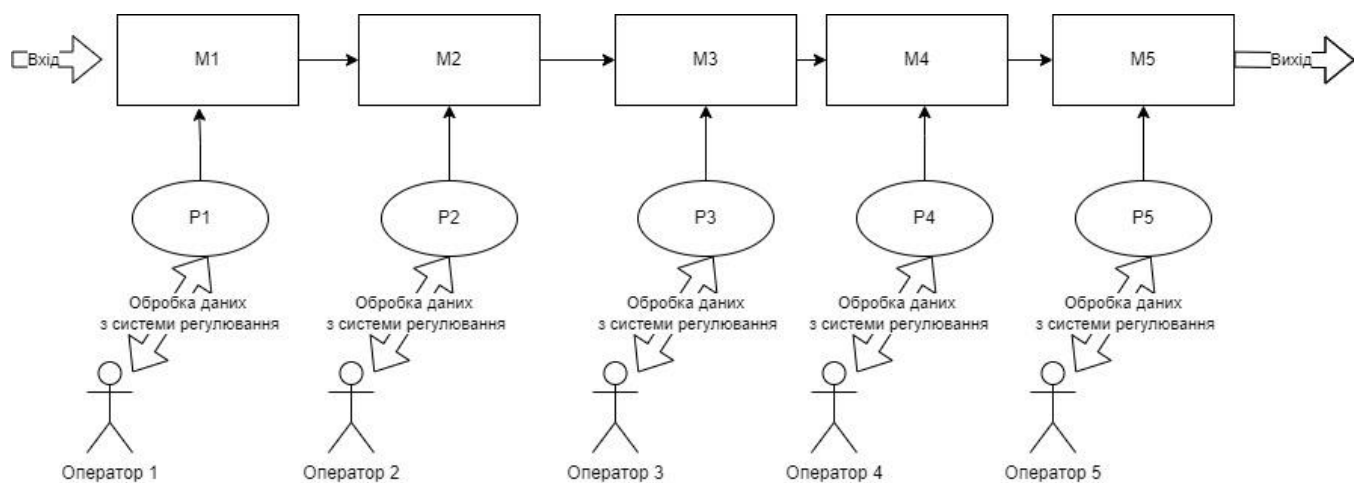


Рисунок 2.1 – Модель виробництва працюючого за класичною схемою

На кожному етапі обробки заготовки ймовірність втрати пильності будь-яким оператором збільшується, що приводить до збільшення ймовірності виникнення браку, а це в свою чергу втрачені ресурси.

При виникненні небажаних ситуацій оператор може відволіктися від регулятора, що призведе до пошкодження обладнання, що може призвести до великих затрат на його відновлення, або взагалі зупинити всю лінію.

Окрім класичної схеми координації є ще централізована схема координації підсистемами.

Особливістю централізованої системи являється те що, данні з усіх модулів обробляються в одному регуляторі та при виникненні критичних відхилень від норми можуть вимкнути модуль самостійно та зберегти дороге обладнання, подібні зупинки можуть виникати для обслуговування модулів.

Як видно з рис. 2.2, на якому приведена типова схема з централізованою координацією, існує лише один регулятор, і один оператор, навіть, якщо оператор

втрата пильності то виникне небажана ситуація, то все закінчиться сприятливо для підприємства, так як модуль – вимкнеться. Такі системи є більш ресурсозатратними, але стабільніші для виробництва.

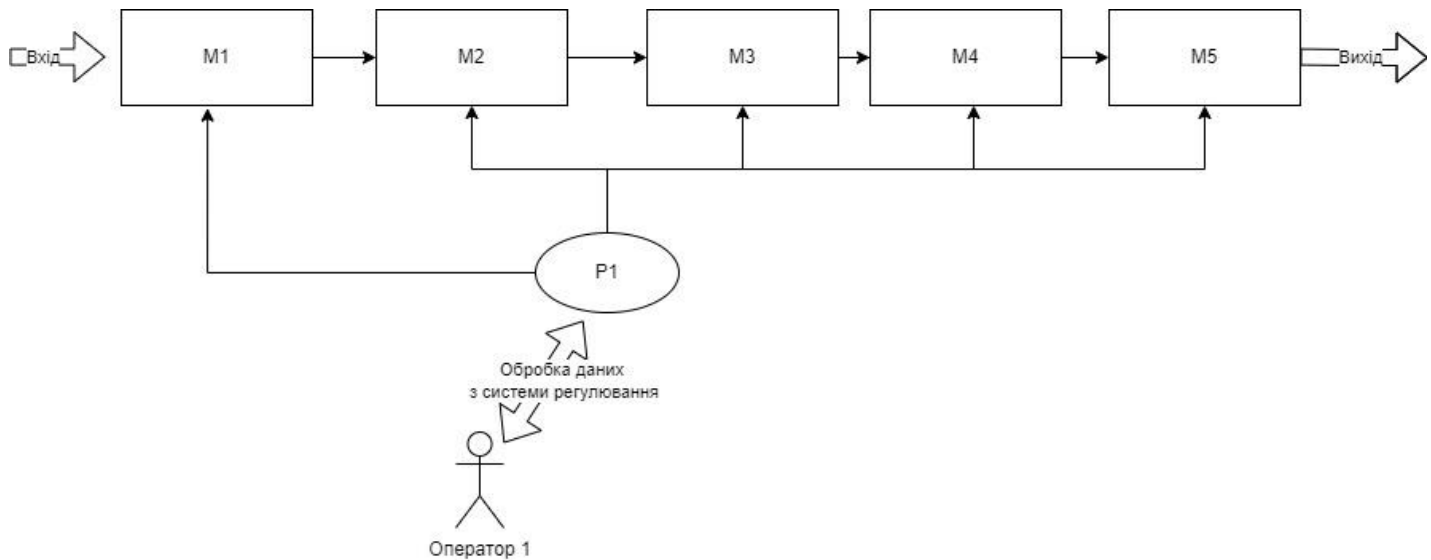


Рисунок 2.2 – Схема з централізованою координацією

Виходячи з рисунку 2.1 можна сказати, що математичний запис теоретичного можливого виникнення браку на даній лінії визначається формулою (2.1):

$$P(I) = M_1 + M_2 + M_3 + M_4 + M_5 + M_{\Pi} + M_B, \quad (2.1)$$

де $P(I)$ – вірогідність випуску браку;

$M_1 \dots M_n$ – вірогідність випуску браку модулем n , яка розраховується за формулою (2.2);

M_{Π} – вірогідність помилки оператора;

M_B – вірогідність використання неякісних матеріалів.

$$M_{\Pi} = P_M + P_{\text{втр}} + P_{\text{відкл}} + P_{\Pi}, \quad (2.2)$$

де P_M – вірогідність неточної роботи модуля, яка розраховується за формулою (2.3);

$P_{\text{втр}}$ – вірогідність механічного втручання в процеси модуля, яка розраховується за формулою (2.4);

$P_{\text{відкл}}$ – вірогідність відключення живлення;

P_{Π} – вірогідність виникнення помилки яка розраховується за формулою (2.5).

$$P_M = \Phi_B + \Phi_3 + \Phi_H, \quad (2.3)$$

де Φ_B – вірогідність відволікання працівника;
 Φ_3 – вірогідність затримки в відображенні інформації;
 Φ_H – вірогідність некоректних даних.

$$P_{\text{втр}} = \frac{K+D}{B_p}, \quad (2.4)$$

де B_p – відстань між модулями;
 K – кількість під модулів;
 D – робочий діапазон під модуля.

$$P_{\Pi} = C \cdot P_c, \quad (2.5)$$

де C – кількість команд в програмі;
 P_c – вірогідність виникнення помилки в певній команді.

Теоретична кількість відмов модулів на виробничій лінії розраховується за формулою (2.6):

$$N_o = N_M \cdot K, \quad (2.6)$$

де N_M – кількість відмов одного модуля.

Кількість своєчасного відключення модуля під час відмов за місяць розраховується за формулою (2.7):

$$N_{\Pi} = \frac{P_{\text{втр}} \cdot 100}{N_o \cdot P_{\Pi}} \cdot P_{\text{від}}, \quad (2.7)$$

де Z_{Π} – загальна кількість відключень;
 $P_{\text{від}}$ – вірогідність вдалого відключення модуля.

Для централізованої моделі яку приведено на рис. 2.2, вірогідність випуску браку буде мати вид формули (2.8):

$$P(l) = M_1 + M_2 + M_3 + M_4 + M_5 + M_B. \quad (2.8)$$

В формулі (2.8) виключається параметр вірогідності помилки оператора тому, що централізована система розроблюваної системи, матиме програмний модуль-автомат вимкнення усієї системи управління підприємства, при виникненні небажаних ситуацій, та потреба в реагуванні оператором зменшується до 0.

Для класичної схеми параметри будуть наступними. Допускаючи що всі 5 модулів є типовими, то їх параметри будуть однаковими. В загальному вигляді для розрахунків буде використано наступні значення параметрів:

- $M_{\Pi}=0,1$;
- $M_B=0,05$;
- $P_{\text{відкл}}=0,0002$;
- $\Phi_B=0,01$;
- $\Phi_3=0,003$;
- $\Phi_H=0,01$;
- $V_p=50, \text{ м}$;
- $K=2$;
- $D=0,8, \text{ м}$;
- $C=5000$;
- $P_c=0,000002$;
- $N_M=4$;
- $Z_{\Pi}=50$;
- $P_{\text{від}}=0,01$.

$$P_M = 0,01 + 0,03 + 0,01 = 0,05,$$

$$P_{\text{втр}} = \frac{2 + 0,8}{75} = 0,037,$$

$$P_{\Pi} = 5000 \cdot 0,000002 = 0,01,$$

$$M_{\Pi} = 0,05 + 0,037 + 0,002 + 0,01 = 0,097,$$

$$P(l) = 0,097 + 0,097 + 0,097 + 0,097 + 0,097 + 0,1 + 0,05 = 0,63,$$

$$N_o = 4 \cdot 2 = 8,$$

$$N_{\Pi} = \frac{0,037 \cdot 100}{8 \cdot 0,01} \cdot 0,1 = 4,25$$

Для централізованої схеми координації параметри дещо зміняться так як система менш залежна від робітника, то параметри M_{Π} та Φ_B будуть відсутні, а $P_{\text{від}}$ збільшено до 0,15.

$$P_M = 0,03 + 0,01 = 0,04,$$

$$M_{\Pi} = 0,04 + 0,037 + 0,002 + 0,01 = 0,089,$$

$$P(1) = 0,089 + 0,089 + 0,089 + 0,089 + 0,089 + 0,05 = 0,495,$$

$$N_{\Pi} = \frac{0,037 \cdot 100}{8 \cdot 0,01} \cdot 0,15 = 6,97$$

Для порівняння результатів розрахунків побудуємо графіки теоретичного виникнення браку та кількості своєчасно відключених модуля під час відмов, за останні 30 днів, результати приведено на рис. 2.3 та 2.4



Рисунок 2.3 – Графіки теоретичного виникнення браку

На графіку позначено 2 лінії: блакитною є лінія класичної координації, а помаранчева – лінія централізованої координації, як видно з графіка, централізована

система координації має менший випуск браку приблизно на 3 одиниці, що в відсотковому значення становить 16,6 %.

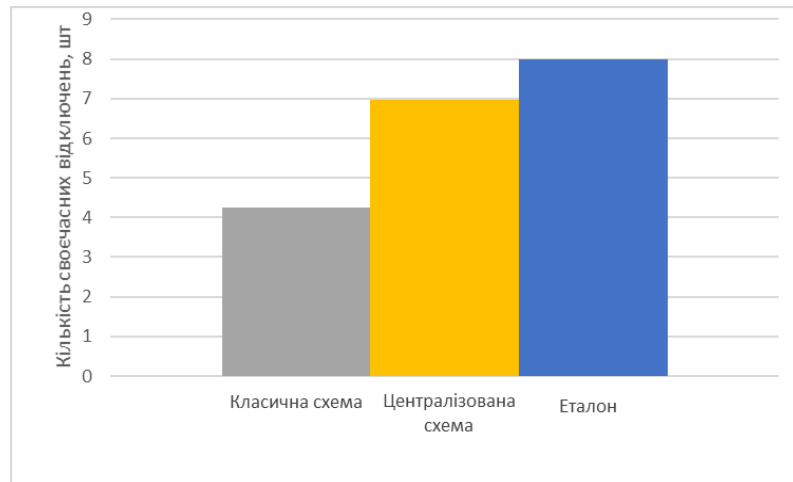


Рисунок 2.4 – Кількість своєчасних відключень модуля під час відмов

На рис. 2.5 приведено графік кількості своєчасно відключених модулів під час відмов, де блакитним кольором позначено загальну кількість відмов за 30 днів, жовтим кольором при роботі за централізованою схемою, а сірим при роботі за класичною схемою виробництва.

Виходячи з отриманих графіків можна сказати, що централізована схема координації підприємством є більш раціональною та має менший брак, що дозволить підприємству мати менші фінансові втрати.

2.2 Розробка математичної моделі управління виробничими процесами складальної автомобільної лінії

В загальнорму вигляді підприємство має 6 модулів процесами яких можна керувати та проводити моніторинг їх параметрів.

Для визначення параметрів математичної моделі всього виробництва необхідно визначити параметри математичної моделі кожного модуля виробництва окремо.

Параметри математичної моделі будь-якої складальної ділянки представляє собою сукупність процесів координації вимкнення/вимкнення модуля, зміни параметрів, моніторингу параметрів кожного модуля, так як схема роботи кожного модуля є типовою, то можна скористатися загальною математичною моделлю, яка підійде кожному окремому модулю.

Загальна математична модель модуль наведено в формулі (2.9).

$$Ma = Mo \vee FN \vee Ch, \quad (2.9)$$

де Mo – процес моніторингу модуля, який описується формулою (2.10);

FN – процес вимкнення/ввімкнення модуля;

Ch – процес зміни параметрів модуля для переналагодження, який описується формулою (2.11).

$$Mo = \sum_0^n Param_1 \wedge \dots \wedge Param_n \vee Log_n \vee S, \quad (2.10)$$

$$Ch = \sum_0^n Ch(Param_n) \wedge S, \quad (2.11)$$

де $Param_n$ – моніторинг n -го параметру модуля;

Log_n – процес логування кожного параметра;

S – процес збереження даних;

$Ch(Param_n)$ – процес зміни n -го параметру модуля.

Приймаючи до уваги той факт, параметри математичної моделі підприємства – це сукупність моделей кожного окремого модуля, моделі процесу логування роботоспроможності кожної ділянки та процесу зупинення/запуску. Тоді узагальнену математичну модель параметрів підприємства можна записати як (2.12):

$$P = \sum_0^n Ma_n \vee M(P) \vee FN_P, \quad (2.12)$$

де Ma_n – математична модель параметрів кожного модуля підприємства;

$M(P)$ – модель процесу моніторинга всього підприємства, яка описується формулою (2.13);

FN_P – модуль процесу зупинки/запуску підприємства.

$$M(P) = \sum_0^n Event_{FN} \vee W(FN_n), \quad (2.13)$$

де $Event_{FN}$ – процес вимкнення/запуск виробничої лінії підприємства;

$W(FN_n)$ – процес запису логу про вимкнення/запуск лінії підприємства.

Беручи до уваги факт, що узагальнена типова складальна лінія автомобілів, як правило, складається із 6 основних модулів, а саме: модулю складання двигуна, мотажу

салону, монтажу кузовних деталей, збірки шасі, з'єднання кузова та шасі, контролю якості, проведемо формалізацію та розроблення математичного забезпечення, яке може складатися з однієї об'єднаної моделі, або декількох окремих моделей. На базі цих розроблених моделей зможемо провести розроблення алгоритму та програмного забезпечення, які реалізують ці моделі.

Отже, узагальнену математичну модель модуля складання двигуна наведено в формулі (2.14):

$$M_e = M_{o_e} \vee FN_e \vee Ch_e, \quad (2.14)$$

$$M_{o_e} = S_{vol} \wedge S_{rot} \wedge S_{valv} \wedge S_{pow} \wedge S_{pist} \wedge log_e, \quad (2.15)$$

де FN_e – набір булевих значень параметрів, який описується формулою (2.16);

Ch_e – набір параметрів двигуна, які можна змінити, який описується формулою (2.17);

S_{vol} – набір параметрів об'єму двигуна, який описується формулою (2.23);

S_{rot} – набір параметрів обертового моменту, який описується формулою (2.24);

S_{valv} – набір параметрів кількості клапанів, який описується формулою (2.25);

S_{pow} – набір параметрів потужності двигуна, який описується формулою (2.26);

S_{pist} – набір параметрів кількості поршнів, який описується формулою (2.27);

log_e – набір залогованих параметрів для кожної одиниці випущеного продукту, який описується формулою (2.28).

$$FN_e = \{Off_{M_e}, On_{M_e}\}, \quad (2.16)$$

де Off_{M_e} – значення для вимкнення модуля збірки двигуна;

On_{M_e} – значення для запуску функціонування модуля двигуна.

$$Ch_e = Ch_{val} \wedge Ch_{rot} \wedge Ch_{valv} \wedge Ch_{pow} \wedge Ch_{pist}, \quad (2.17)$$

де Ch_{val} – набір параметрів для зміни значення об'єму двигуна, який описується формулою (2.18);

Ch_{rot} – набір параметрів для зміни значення обертового моменту двигуна, який описується формулою (2.19);

Ch_{valv} – набір параметрів для зміни значення кількості клапанів двигуна, який описується формулою (2.20);

Ch_{pow} – набір параметрів для зміни значення потужності двигуна, який описується формулою (2.21);

Ch_{pist} – набір параметрів для зміни значення кількості поршнів двигуна, який описується формулою (2.22).

Враховуючи факт, що параметри Ch_{valv} та Ch_{pist} не можуть бути не цілими значеннями – тоді, зміна даних параметрів зводить до вказання значення цих параметрів та описується формулою (2.20) та (2.22), інші параметри мають бути не цілими через, що кожен параметр описаний формулами (2.18 - 2.21) будуть мати набір з 3 параметрів.

$$Ch_{val} = \{val_{val}, valPos_{val}, valNeg_{val}\}, \quad (2.18)$$

де val_{val} – значення еталонного параметра об'єму двигуна;

$valPos_{val}$ – значення позитивного допуску параметра об'єму двигуна;

$valNeg_{val}$ – значення негативного допуску параметра об'єму двигуна.

$$Ch_{rot} = \{val_{rot}, valPos_{rot}, valNeg_{rot}\}, \quad (2.19)$$

де val_{rot} – значення еталонного параметра обертового моменту двигуна;

$valPos_{rot}$ – значення позитивного допуску параметра обертового моменту двигуна;

$valNeg_{rot}$ – значення негативного допуску параметра обертового моменту двигуна.

$$Ch_{valv} = \{val_{valv}\}, \quad (2.20)$$

де val_{valv} – значення еталонного параметра кількості клапанів двигуна.

$$Ch_{pow} = \{val_{pow}, valPos_{pow}, valNeg_{pow}\}, \quad (2.21)$$

де val_{pow} – значення еталонного параметра потужності двигуна;

$valPos_{pow}$ – значення позитивного допуску параметра потужності двигуна;

$valNeg_{pow}$ – значення негативного допуску параметра потужності двигуна.

$$Ch_{pist} = \{val_{pist}\}, \quad (2.22)$$

де val_{pist} – значення еталонного параметра кількості поршнів двигуна.

Кожен параметр з формули (2.15) представляє собою формальний набір параметрів еталонного значення, позитивного та негативного допуску відхилень, та дійсне значення, яке приходить з API даного модуля:

$$S_{vol} = \{E_{vol}, PosE_{vol}, NegE_{vol}, Native_{vol}\}, \quad (2.23)$$

де E_{vol} – еталонне значення параметра об'єму двигуна;

$PosE_{vol}$ – значення позитивної похибки параметра об'єму двигуна;

$NegE_{vol}$ – значення негативної похибки параметра об'єму двигуна;

$Native_{vol}$ – дійсне значення параметра об'єму двигуна.

$$S_{rot} = \{E_{rot}, PosE_{rot}, NegE_{rot}, Native_{rot}\}, \quad (2.24)$$

де E_{rot} – еталонне значення параметра обертового моменту;

$PosE_{rot}$ – значення позитивної похибки параметра обертового моменту;

$NegE_{rot}$ – значення негативної похибки параметра обертового моменту;

$Native_{rot}$ – дійсне значення параметра обертового моменту.

$$S_{volv} = \{E_{volv}, PosE_{volv}, NegE_{volv}, Native_{volv}\}, \quad (2.25)$$

де E_{volv} – еталонне значення параметра кількості клапанів двигуна;

$PosE_{volv}$ – значення позитивної похибки параметра кількості клапанів двигуна;

$NegE_{volv}$ – значення негативної похибки параметра кількості клапанів двигуна;

$Native_{volv}$ – дійсне значення параметра кількості клапанів двигуна.

$$S_{pow} = \{E_{pow}, PosE_{pow}, NegE_{pow}, Native_{pow}\}, \quad (2.26)$$

де E_{pow} – еталонне значення параметра потужності двигуна;

$PosE_{pow}$ – значення позитивної похибки параметра потужності двигуна;

$NegE_{pow}$ – значення негативної похибки параметра потужності двигуна;

$Native_{pow}$ – дійсне значення параметра потужності двигуна.

$$S_{pist} = \{E_{pist}, PosE_{pist}, NegE_{pist}, Native_{pist}\}, \quad (2.27)$$

де E_{pist} – еталонне значення параметра кількості поршнів двигуна;

$PosE_{pist}$ – значення позитивної похибки параметра кількості поршнів двигуна;

$NegE_{pist}$ – значення негативної похибки параметра кількості поршнів двигуна;

$Native_{pist}$ – дійсне значення параметра кількості поршнів двигуна.

$$\log_e = Native_{val} \wedge Native_{rot} \wedge Native_{valv} \wedge Native_{pow} \wedge Native_{pist}. \quad (2.28)$$

Спираючись на параметри, що було описано у формулах (2.15-2.28), отримуємо загальну математичну модель модуля збірки двигуна, яка згідно формулі (2.14), являє собою суму параметрів описаних в формулах (2.15-2.17), загальний вигляд математичної моделі модуля приведено в формулі (2.29)

$$M_e = (S_{vol} \wedge S_{rot} \wedge S_{volv} \wedge S_{pow} \wedge S_{pist} \wedge \log_e) \vee [Off_{M_e}, On_{M_e}] \vee (Ch_{val} \wedge Ch_{rot} \wedge Ch_{valv} \wedge Ch_{pow} \wedge Ch_{pist}). \quad (2.29)$$

Отже, математична модель модуля збірки салону:

$$M_i = Mo_i \vee FN_i \vee Ch_i, \quad (2.30)$$

$$Mo_i = S_{sits} \wedge S_{tor} \wedge S_{metric} \wedge S_{indic} \wedge S_{media} \wedge \log_i, \quad (2.31)$$

де FN_i – набір булевих значень параметрів, який описується формулою (2.32);

Ch_i – набір параметрів, які описують параметр кількості сидінь, які можна змінити, який описується формулою (2.33);

S_{sits} – набір параметрів які описують параметр кількості сидінь, який описується формулою (2.39);

S_{tor} – набір параметрів допуску встановлення торпеди, який описується формулою (2.40);

S_{metric} – набір параметрів кількості вимірювальних приладів, який описується формулою (2.41);

S_{indic} – набір параметрів кількості індикаторів, який описується формулою (2.42);

S_{media} – набір параметрів кількості медіа модулів, який описується формулою (2.43);

\log_i – набір залогованих параметрів для кожної одиниці випущеного продукту, який описується формулою (2.39).

$$FN_i = \{Off_{M_i}, On_{M_i}\}, \quad (2.32)$$

де Off_{M_i} – значення для вимкнення модуля збірки двигуна;

On_{M_i} – значення для запуску функціонування модуля двигуна.

$$Ch_i = \{Ch_{sits} \wedge Ch_{tor} \wedge Ch_{metric} \wedge Ch_{indic} \wedge Ch_{media}\}, \quad (2.33)$$

де Ch_{sits} – набір параметрів для зміни значення кількості сидінь, який описується формулою (2.34);

Ch_{tor} – набір параметрів для зміни значення допуску встановлення торпеди двигуна, який описується формулою (2.35);

Ch_{metric} – набір параметрів для зміни значення кількості вимірювальних приборів, який описується формулою (2.36);

Ch_{indic} – набір параметрів для зміни значення кількості індикаторів, який описується формулою (2.37);

Ch_{media} – набір параметрів для зміни значення кількості медіа модулів, який описується формулою (2.38).

Враховуючи факт, що тільки параметр Ch_{tor} може приймати не цілі значення, можемо зробити висновок, що він може мати три різні значення, а інші параметри, що враховані в формулі (2.33), можуть бути лише цілими то значення цих параметрів і буде їх математичним описом.

$$Ch_{sits} = \{val_{sits}\}, \quad (2.34)$$

де val_{sits} – значення еталонного параметра кількості сидінь.

$$Ch_{tor} = \{val_{tor}, valPos_{tor}, valNeg_{tor}\}, \quad (2.35)$$

де val_{tor} – значення еталонного параметра допуску встановлення торпеди;

$valPos_{tor}$ – значення позитивного допуску параметра допуску встановлення торпеди;

$valNeg_{tor}$ – значення негативного допуску параметра допуску встановлення торпеди.

$$Ch_{metric} = \{val_{metric}\}, \quad (2.36)$$

де val_{metric} – значення еталонного параметра кількості вимірювальних приборів.

$$Ch_{indic} = \{val_{indic}\}, \quad (2.37)$$

де val_{indic} – значення еталонного параметра кількості індикаторів.

$$Ch_{media} = \{val_{media}\}, \quad (2.38)$$

де val_{media} – значення еталонного параметра кількості медіа модулів.

Кожен параметр формула (2.31) являє собою формальний набір параметрів еталонного значення, позитивного та негативного допусків відхилень, та дійсне значення яке приходить з API даного модуля.

$$S_{sits} = \{E_{sits}, PosE_{sits}, NegE_{sits}, Native_{sits}\}, \quad (2.39)$$

де E_{sits} – еталонне значення параметра кількості сидінь;

$PosE_{sits}$ – значення позитивної похибки параметра кількості сидінь;

$NegE_{sits}$ – значення негативної похибки параметра кількості сидінь;

$Native_{sits}$ – дійсне значення параметра кількості сидінь.

$$S_{tor} = \{E_{tor}, PosE_{tor}, NegE_{tor}, Native_{tor}\}, \quad (2.40)$$

де E_{tor} – еталонне значення параметра допуску встановлення торпеди;
 $PosE_{tor}$ – значення позитивної похибки параметра допуску встановлення торпеди;
 $NegE_{tor}$ – значення негативної похибки параметра допуску встановлення торпеди;
 $Native_{tor}$ – дійсне значення параметра допуску встановлення торпеди.

$$S_{metric} = \{E_{metric}, PosE_{metric}, NegE_{metric}, Native_{metric}\}, \quad (2.41)$$

де E_{metric} – еталонне значення параметра кількості вимірювальних приборів;
 $PosE_{metric}$ – значення позитивної похибки параметра кількості вимірювальних приборів;
 $NegE_{metric}$ – значення негативної похибки параметра кількості вимірювальних приборів;
 $Native_{metric}$ – дійсне значення параметра кількості вимірювальних приборів.

$$S_{indic} = \{E_{indic}, PosE_{indic}, NegE_{indic}, Native_{indic}\}, \quad (2.42)$$

де E_{indic} – еталонне значення параметра кількості індикаторів;
 $PosE_{indic}$ – значення позитивної похибки параметра кількості індикаторів;
 $NegE_{indic}$ – значення негативної похибки параметра кількості індикаторів;
 $Native_{indic}$ – дійсне значення параметра кількості індикаторів.

$$S_{media} = \{E_{media}, PosE_{media}, NegE_{media}, Native_{media}\}, \quad (2.43)$$

де E_{media} – еталонне значення параметра кількості медіа модулів;
 $PosE_{media}$ – значення позитивної похибки параметра кількості медіа модулів;
 $NegE_{media}$ – значення негативної похибки параметра кількості медіа модулів;
 $Native_{media}$ – дійсне значення параметра кількості медіа модулів.

$$\log_i = \{Native_{sits} \wedge Native_{tor} \wedge Native_{metric} \wedge Native_{indic} \wedge Native_{media}\}. \quad (2.44)$$

Виходячи з формул (2.31-2.45), можна отримати загальну математичну модель модуля збірки салону, яка відповідно до формули (2.30), являє собою суму параметрів

описаних в формулах (2.31-2.33), загальний вигляд математичної моделі модуля наведено в формулі (2.45):

$$M_i = (S_{sits} \wedge S_{tor} \wedge S_{metric} \wedge S_{indic} \wedge S_{media} \wedge log_i) \vee [Off_{M_i}, On_{M_i}] \vee Ch_{sits} \wedge Ch_{tor} \wedge Ch_{metric} \wedge Ch_{indic} \wedge Ch_{media}, \quad (2.45)$$

Математична модель модуля збірки кузова наведено в формулі (2.46):

$$M_b = Mo_b \vee FN_b \vee Ch_b, \quad (2.46)$$

$$Mo_b = S_{hood} \wedge S_{door} \wedge S_{frontbamp} \wedge S_{backbamp} \wedge S_{wind} \wedge S_{paint} \wedge S_{trank} \wedge log_b, \quad (2.47)$$

де FN_b – набір булевих значень параметрів, який описується формулою (2.48);

Ch_b – набір параметрів, модуля збірки кузова, які можна змінити, який описується формулою (2.49);

S_{hood} – набір параметрів які описують параметр допуск встановлення капота, який описується формулою (2.57);

S_{door} – набір параметрів допуску встановлення дверей, який описується формулою (2.58);

$S_{frontbamp}$ – набір параметрів допуску встановлення переднього бампера, який описується формулою (2.59);

$S_{backbamp}$ – набір параметрів допуску встановлення заднього бампера, який описується формулою (2.60);

S_{wind} – набір параметрів допуску встановлення вітрового скла, який описується формулою (2.61);

S_{paint} – набір параметрів допуску фарбування, який описується формулою (2.62);

S_{trank} – набір параметрів допуску встановлення багажника, який описується формулою (2.63);

log_b – набір залогованих параметрів для кожної одиниці випущеного продукту, який описується формулою 2.64.

$$FN_b = \{Off_{M_b}, On_{M_b}\}, \quad (2.48)$$

де Off_{M_b} – значення для вимкнення модуля збірки двигуна;

Op_{M_b} – значення для запуску функціонування модуля двигуна.

$$Ch_b = Ch_{hood} \wedge Ch_{door} \wedge Ch_{frontbamp} \wedge Ch_{backbamp} \wedge Ch_{wind} \wedge Ch_{paint} \wedge Ch_{trank}, \quad (2.49)$$

де Ch_{hood} – набір параметрів для зміни значення допуску встановлення капоту, який описується формулою (2.50);

Ch_{door} – набір параметрів для зміни значення допуску встановлення дверей, який описується формулою (2.51);

$Ch_{frontbamp}$ – набір параметрів для зміни значення допуску встановлення переднього бампера, який описується формулою (2.52);

$Ch_{backbamp}$ – набір параметрів для зміни значення допуску встановлення заднього бампера, який описується формулою (2.53);

Ch_{wind} – набір параметрів для зміни значення допуску встановлення вітрового скла, який описується формулою (2.54);

Ch_{paint} – набір параметрів для зміни значення допуску фарбування, який описується формулою (2.55);

Ch_{trank} – набір параметрів для зміни значення допуску встановлення багажника, який описується формулою (2.56).

$$Ch_{hood} = \{val_{hood}, valPos_{hood}, valNeg_{hood}\}, \quad (2.50)$$

де val_{hood} – значення еталонного параметра допуску встановлення капоту;

$valPos_{hood}$ – значення позитивного допуску параметра допуску встановлення капота;

$valNeg_{hood}$ – значення негативного допуску параметра допуску встановлення капота.

$$Ch_{door} = \{val_{door}, valPos_{door}, valNeg_{door}\}, \quad (2.51)$$

де val_{door} – значення еталонного параметра допуску встановлення дверей;

$valPos_{door}$ – значення позитивного допуску параметра допуску встановлення дверей;

$\text{valNeg}_{\text{door}}$ – значення негативного допуску параметра допуску встановлення дверей.

$$\text{Ch}_{\text{frontbamp}} = \{\text{val}_{\text{frontbamp}}, \text{valPos}_{\text{frontbamp}}, \text{valNeg}_{\text{frontbamp}}\}, \quad (2.52)$$

де $\text{val}_{\text{frontbamp}}$ – значення еталонного параметра допуску встановлення переднього бампера;

$\text{valPos}_{\text{frontbamp}}$ – значення позитивного допуску параметра допуску встановлення переднього бампера;

$\text{valNeg}_{\text{frontbamp}}$ – значення негативного допуску параметра допуску встановлення переднього бампера.

$$\text{Ch}_{\text{backbamp}} = \{\text{val}_{\text{backbamp}}, \text{valPos}_{\text{backbamp}}, \text{valNeg}_{\text{backbamp}}\}, \quad (2.53)$$

де $\text{val}_{\text{backbamp}}$ – значення еталонного параметра допуску встановлення заднього бампера;

$\text{valPos}_{\text{backbamp}}$ – значення позитивного допуску параметра допуску встановлення заднього бампера;

$\text{valNeg}_{\text{backbamp}}$ – значення негативного допуску параметра допуску встановлення заднього бампера.

$$\text{Ch}_{\text{wind}} = \{\text{val}_{\text{wind}}, \text{valPos}_{\text{wind}}, \text{valNeg}_{\text{wind}}\}, \quad (2.54)$$

де val_{wind} – значення еталонного параметра допуску встановлення вітрового скла;

$\text{valPos}_{\text{wind}}$ – значення позитивного допуску параметра допуску встановлення вітрового скла;

$\text{valNeg}_{\text{wind}}$ – значення негативного допуску параметра допуску встановлення вітрового скла.

$$\text{Ch}_{\text{paint}} = \{\text{val}_{\text{paint}}, \text{valPos}_{\text{paint}}, \text{valNeg}_{\text{paint}}\}, \quad (2.55)$$

де $\text{val}_{\text{paint}}$ – значення еталонного параметра допуску фарбування;

$\text{valPos}_{\text{paint}}$ – значення позитивного допуску параметра допуску фарбування;

$valNeg_{paint}$ – значення негативного допуску параметра допуску фарбування.

$$Ch_{trank} = \{val_{trank}, valPos_{trank}, valNeg_{trank}\}, \quad (2.56)$$

де val_{trank} – значення еталонного параметра допуску встановлення багажника;

$valPos_{trank}$ – значення позитивного допуску параметра допуску встановлення багажника;

$valNeg_{trank}$ – значення негативного допуску параметра допуску встановлення багажника.

Кожен параметр формула (2.47) собою формальний набір параметрів еталонного значення, позитивного та негативного допуску відхилення, та дійсне значення яке приходить з АРІ даного модуля.

$$S_{hood} = \{E_{hoodvol}, PosE_{hood}, NegE_{hood}, Native_{hood}\}, \quad (2.57)$$

де E_{hood} – еталонне значення параметра допуску встановлення капота;

$PosE_{hood}$ – значення позитивної похибки параметра допуску встановлення капота;

$NegE_{hood}$ – значення негативної похибки параметра допуску встановлення капота;

$Native_{hood}$ – дійсне значення параметра допуску встановлення капота.

$$S_{door} = \{E_{door}, PosE_{door}, NegE_{rot}, Native_{door}\}, \quad (2.58)$$

де E_{door} – еталонне значення параметра допуску встановлення дверей;

$PosE_{door}$ – значення позитивної похибки параметра допуску встановлення дверей;

$NegE_{door}$ – значення негативної похибки параметра допуску встановлення дверей;

$Native_{door}$ – дійсне значення параметра допуску встановлення дверей.

$$S_{frontbamp} = \{E_{frontbamp}, PosE_{frontbamp}, NegE_{frontbamp}, Native_{frontbamp}\}, \quad (2.59)$$

де $E_{frontbamp}$ – еталонне значення параметра допуску встановлення переднього бампера;

$PosE_{frontbamp}$ – значення позитивної похибки допуску встановлення переднього бампера;

$NegE_{frontbamp}$ – значення негативної похибки параметра допуску встановлення переднього бампера;

$Native_{frontbamp}$ – дійсне значення параметра допуску встановлення переднього бампера.

$$S_{backbamp} = \{E_{backbamp}, PosE_{backbamp}, NegE_{backbamp}, Native_{backbamp}\}, \quad (2.60)$$

де $E_{backbamp}$ – еталонне значення параметра допуску встановлення заднього бампера;

$PosE_{backbamp}$ – значення позитивної похибки параметра допуску встановлення заднього бампера;

$NegE_{backbamp}$ – значення негативної похибки параметра допуску встановлення заднього бампера;

$Native_{backbamp}$ – дійсне значення параметра допуску встановлення заднього бампера.

$$S_{wind} = \{E_{wind}, PosE_{wind}, NegE_{wind}, Native_{wind}\}, \quad (2.61)$$

де E_{wind} – еталонне значення параметра допуску встановлення вітрового скла;

$PosE_{wind}$ – значення позитивної похибки параметра допуску встановлення вітрового скла;

$NegE_{wind}$ – значення негативної похибки параметра допуску встановлення вітрового скла;

$Native_{wind}$ – дійсне значення параметра допуску встановлення вітрового скла.

$$S_{paint} = \{E_{paint}, PosE_{paint}, NegE_{paint}, Native_{paint}\}, \quad (2.62)$$

де E_{paint} – еталонне значення параметра допуск фарбування;

$PosE_{paint}$ – значення позитивної похибки параметра допуск фарбування;

$NegE_{paint}$ – значення негативної похибки параметра допуск фарбування;

$Native_{paint}$ – дійсне значення параметра допуск фарбування.

$$S_{trank} = \{E_{trank}, PosE_{trank}, NegE_{trank}, Native_{trank}\}, \quad (2.63)$$

де E_{trank} – еталонне значення параметра допуск встановлення багажника;

$PosE_{trank}$ – значення позитивної похибки параметра допуск встановлення багажника;

$NegE_{trank}$ – значення негативної похибки параметра допуск встановлення багажника;

$Native_{trank}$ – дійсне значення параметра допуск встановлення багажника.

$$\log_b = Native_{hood} \wedge Native_{door} \wedge Native_{frontbamp} \wedge Native_{backbamp} \wedge Native_{wind} \wedge Native_{paint} \wedge Native_{trank} \quad (2.64)$$

Виходячи з формул (2.47-2.64), можна отримати загальну математичну модель модуля збірки кузова, яка згідно формулі (2.46), являє собою суму параметрів описаних в формулах (2.47-2.49), повний вигляд математичної моделі модуля приведено в формулі (2.65).

$$M_b = (S_{hood} \wedge S_{door} \wedge S_{frontbamp} \wedge S_{backbamp} \wedge S_{wind} \wedge S_{paint} \wedge S_{trank} \wedge \log_b) \vee [Off_{M_b}, On_{M_b}] \vee Ch_{hood} \wedge Ch_{door} \wedge Ch_{frontbamp} \wedge Ch_{backbamp} \wedge Ch_{wind} \wedge Ch_{paint} \wedge Ch_{trank} \quad (2.65)$$

Математична модель модуля збірки шасі наведено в формулі (2.66):

$$M_s = Mo_s \vee FN_s \vee Ch_s \quad , \quad (2.66)$$

$$Mo_s = S_r \wedge S_{stab} \wedge S_{shar} \wedge S_{pr} \wedge S_{red} \wedge \log_s, \quad (2.67)$$

де FN_s – набір бульових значень параметрів, який описується формулою (2.68);

Ch_s – набір параметрів, модуля збірки шасі, які можна змінити, який описується формулою (2.69);

S_r – набір параметрів які описують параметр запасу міцності ричага, який описується формулою (2.75);

S_{stab} – набір параметрів запасу міцності стабілізатора, який описується формулою (2.76);

S_{shar} – набір параметрів запасу міцності шарової опори, який описується формулою (2.77);

S_{pr} – набір параметрів запасу міцності приводів, який описується формулою (2.78);

S_{red} – набір параметрів запасу міцності редуктора, який описується формулою (2.78);

\log_s – набір залогованих параметрів для кожної одиниці випущеного продукту, який описується формулою (2.80).

$$FN_s = \{Off_{M_s}, On_{M_s}\}, \quad (2.68)$$

де Off_{M_s} – значення для вимкнення модуля збірки двигуна;

On_{M_s} – значення для запуску функціонування модуля двигуна.

$$Ch_s = \{Ch_r \wedge Ch_{\text{stab}} \wedge Ch_{\text{shar}} \wedge Ch_{\text{pr}} \wedge Ch_{\text{red}}\}, \quad (2.69)$$

де Ch_r – набір параметрів для зміни значення запасу міцності ричага, який описується формулою (2.70);

Ch_{stab} – набір параметрів для зміни значення запасу міцності стабілізатора, який описується формулою (2.71);

Ch_{shar} – набір параметрів для зміни значення запасу міцності шарової опори, який описується формулою (2.72);

Ch_{pr} – набір параметрів для зміни значення запасу міцності приводів, який описується формулою (2.73);

Ch_{red} – набір параметрів для зміни значення запасу міцності редуктора, який описується формулою (2.74).

$$Ch_r = \{val_r, valPos_r, valNeg_r\}, \quad (2.70)$$

де val_r – значення еталонного параметра запасу міцності ричага;

$valPos_r$ – значення позитивного допуску параметра запасу міцності ричага;

$valNeg_r$ – значення негативного допуску параметра запасу міцності ричага.

$$Ch_{stab} = \{val_{stab}, valPos_{stab}, valNeg_{stab}\}, \quad (2.71)$$

де val_{stab} – значення еталонного параметра запасу міцності стабілізатора;
 $valPos_{stab}$ – значення позитивного допуску параметра запасу міцності стабілізатора;
 $valNeg_{stab}$ – значення негативного допуску параметра запасу міцності стабілізатора.

$$Ch_{shar} = \{val_{shar}, valPos_{shar}, valNeg_{shar}\}, \quad (2.72)$$

де val_{shar} – значення еталонного параметра запасу міцності шарової опори;
 $valPos_{shar}$ – значення позитивного допуску параметра запасу міцності шарової опори;
 $valNeg_{shar}$ – значення негативного допуску параметра запасу міцності шарової опори.

$$Ch_{pr} = \{val_{pr}, valPos_{pr}, valNeg_{pr}\}, \quad (2.73)$$

де val_{pr} – значення еталонного параметра запасу міцності приводів;
 $valPos_{pr}$ – значення позитивного допуску параметра запасу міцності приводів;
 $valNeg_{pr}$ – значення негативного допуску параметра запасу міцності приводів.

$$Ch_{red} = \{val_{red}, valPos_{red}, valNeg_{red}\}, \quad (2.74)$$

де val_{red} – значення еталонного параметра запасу міцності редуктора;
 $valPos_{red}$ – значення позитивного допуску параметра запасу міцності редуктора;
 $valNeg_{red}$ – значення негативного допуску параметра запасу міцності редуктора.

$$S_r = \{E_r, PosE_r, NegE_r, Native_r\}, \quad (2.75)$$

де E_r – еталонне значення параметра запасу міцності ричага;
 $PosE_r$ – значення позитивної похибки параметра запасу міцності ричага;
 $NegE_r$ – значення негативної похибки параметра запасу міцності ричага;

$Native_r$ – дійсне значення параметра запасу міцності ричага.

$$S_{stab} = \{E_{stab}, PosE_{stab}, NegE_{stab}, Native_{stab}\}, \quad (2.76)$$

де E_{stab} – еталонне значення параметра запасу міцності стабілізатора;
 $PosE_{stab}$ – значення позитивної похибки параметра запасу міцності стабілізатора;
 $NegE_{stab}$ – значення негативної похибки параметра запасу міцності стабілізатора;
 $Native_{stab}$ – дійсне значення параметра запасу міцності стабілізатора.

$$S_{shar} = \{E_{shar}, PosE_{shar}, NegE_{shar}, Native_{shar}\}, \quad (2.77)$$

де E_{shar} – еталонне значення параметра запасу міцності шарової опори;
 $PosE_{shar}$ – значення позитивної похибки параметра запасу міцності шарової опори;
 $NegE_{shar}$ – значення негативної похибки параметра запасу міцності шарової опори;
 $Native_{shar}$ – дійсне значення параметра запасу міцності шарової опори.

$$S_{pr} = \{E_{pr}, PosE_{pr}, NegE_{pr}, Native_{pr}\}, \quad (2.78)$$

де E_{pr} – еталонне значення параметра запасу міцності приводів;
 $PosE_{pr}$ – значення позитивної похибки параметра запасу міцності приводів;
 $NegE_{pr}$ – значення негативної похибки параметра запасу міцності приводів;
 $Native_{pr}$ – дійсне значення параметра запасу міцності приводів.

$$S_{red} = \{E_{red}, PosE_{red}, NegE_{red}, Native_{red}\}, \quad (2.79)$$

де E_{red} – еталонне значення параметра запасу міцності редуктора;
 $PosE_{red}$ – значення позитивної похибки параметра запасу міцності редуктора;
 $NegE_{red}$ – значення негативної похибки параметра запасу міцності редуктора;
 $Native_{red}$ – дійсне значення параметра запасу міцності редуктора.

$$\log_s = \{Native_r \wedge Native_{stab} \wedge Native_{shar} \wedge Native_{pr} \wedge Native_{red}\}, \quad (2.80)$$

Виходячи з формул (2.67-2.80), можна отримати загальну математичну модель модуля збірки шасі, яка згідно формулі (2.66) являє собою суму параметрів описаних в формулах (2.67-2.69), повний вигляд математичної моделі модуля приведено в формулі (2.81).

$$M_s = (S_r \wedge S_{stab} \wedge S_{shar} \wedge S_{pr} \wedge S_{red} \wedge \log_s) \vee [Off_{M_s}, On_{M_s}] \vee Ch_r \wedge Ch_{stab} \wedge Ch_{shar} \wedge Ch_{pr} \wedge Ch_{red}, \quad (2.81)$$

Математична модель модуля з'єднання шасі та кузова наведено в формулі (2.82)

$$M_m = Mo_m \vee FN_m \vee Ch_m, \quad (2.82)$$

$$Mo_m = S_{mb} \wedge S_{kb} \wedge S_{ho} \wedge S_{cli} \wedge S_{load} \wedge \log_m, \quad (2.83)$$

де FN_m – набір бульових значень параметрів, який описується формулою (2.84);

Ch_m – набір параметрів, модуля з'єднання шасі та кузова, які можна змінити, який описується формулою (2.85);

S_{mb} – набір параметрів які описують параметр точність встановлення кузова, який описується формулою (2.91);

S_{kb} – набір параметрів, які описують крен кузова, який описується формулою (2.92);

S_{ho} – набір параметрів точності співпадіння опорних отворів, який описується формулою (2.93);

S_{cli} – набір параметрів дорожнього просвіту, який описується формулою (2.94);

S_{load} – набір параметрів навантаження на осі, який описується формулою (2.95);

\log_m – набір залогованих параметрів для кожної одиниці випущеного продукту, який описується формулою (2.96).

$$FN_m = \{Off_{M_m}, On_{M_m}\}, \quad (2.84)$$

де Off_{M_m} – значення для вимкнення модуля збірки двигуна;

Op_{M_m} – значення для запуску функціонування модуля двигуна.

$$Ch_m = \{Ch_{mb} \wedge Ch_{kb} \wedge Ch_{ho} \wedge Ch_{cli} \wedge Ch_{load}\}, \quad (2.85)$$

де Ch_{mb} – набір параметрів для зміни значення точності встановлення кузова, який описується формулою (2.86);

Ch_{kb} – набір параметрів для зміни значення крену кузова, який описується формулою (2.87);

Ch_{ho} – набір параметрів для зміни значення точності співпадіння опорних отворів, який описується формулою (2.88);

Ch_{cli} – набір параметрів для зміни значення дорожнього прояснення, який описується формулою (2.89);

Ch_{load} – набір параметрів для зміни значення навантаження на ось, який описується формулою (2.90).

$$Ch_{mb} = \{val_{mb}, valPos_{mb}, valNeg_{mb}\}, \quad (2.86)$$

де val_{mb} – значення еталонного параметра точності встановлення кузова;

$valPos_{mb}$ – значення позитивного допуску параметра точності встановлення кузова;

$valNeg_{mb}$ – значення негативного допуску параметра точності встановлення кузова.

$$Ch_{kb} = \{val_{kb}, valPos_{kb}, valNeg_{kb}\}, \quad (2.87)$$

де val_{kb} – значення еталонного параметра крену кузова;

$valPos_{kb}$ – значення позитивного допуску параметра крену кузова;

$valNeg_{kb}$ – значення негативного допуску параметра крену кузова.

$$Ch_{ho} = \{val_{ho}, valPos_{ho}, valNeg_{ho}\}, \quad (2.88)$$

де val_{ho} – значення еталонного параметра точності співпадіння опорних отворів;

$valPos_{ho}$ – значення позитивного допуску параметра точності співпадіння опорних отворів;

$valNeg_{ho}$ – значення негативного допуску параметра точності співпадіння опорних отворів.

$$Ch_{cli} = \{val_{cli}, valPos_{cli}, valNeg_{cli}\}, \quad (2.89)$$

де val_{cli} – значення еталонного параметра дорожнього просвіту;
 $valPos_{cli}$ – значення позитивного допуску параметра дорожнього просвіту;
 $valNeg_{cli}$ – значення негативного допуску параметра дорожнього просвіту.

$$Ch_{load} = \{val_{load}, valPos_{load}, valNeg_{load}\}, \quad (2.90)$$

де val_{load} – значення еталонного параметра навантаження на ось;
 $valPos_{load}$ – значення позитивного допуску параметра навантаження на ось;
 $valNeg_{load}$ – значення негативного допуску параметра навантаження на ось.

$$S_{mb} = \{E_{mb}, PosE_{mb}, NegE_{mb}, Native_{mb}\}, \quad (2.91)$$

де E_{mb} – еталонне значення параметра точності встановлення кузова;
 $PosE_{mb}$ – значення позитивної похибки параметра точності встановлення кузова;
 $NegE_{mb}$ – значення негативної похибки параметра точності встановлення кузова;
 $Native_{mb}$ – дійсне значення параметра точності встановлення кузова.

$$S_{kb} = \{E_{kb}, PosE_{kb}, NegE_{kb}, Native_{kb}\}, \quad (2.92)$$

де E_{kb} – еталонне значення параметра крену кузова;
 $PosE_{kb}$ – значення позитивної похибки параметра крену кузова;
 $NegE_{kb}$ – значення негативної похибки параметра крену кузова;
 $Native_{kb}$ – дійсне значення параметра крену кузова.

$$S_{ho} = \{E_{ho}, PosE_{ho}, NegE_{ho}, Native_{ho}\}, \quad (2.93)$$

де E_{ho} – еталонне значення параметра точності співпадіння опорних отворів;
 $PosE_{ho}$ – значення позитивної похибки параметра точності співпадіння опорних отворів;

$NegE_{ho}$ – значення негативної похибки параметра точності співпадіння опорних отворів;

$Native_{ho}$ – дійсне значення параметра точності співпадіння опорних отворів.

$$S_{cli} = \{E_{cli}, PosE_{cli}, NegE_{cli}, Native_{cli}\}, \quad (2.94)$$

де E_{cli} – еталонне значення параметра дорожнього просвіту;

$PosE_{cli}$ – значення позитивної похибки параметра дорожнього просвіту;

$NegE_{cli}$ – значення негативної похибки параметра дорожнього просвіту;

$Native_{cli}$ – дійсне значення параметра дорожнього просвіту.

$$S_{load} = \{E_{load}, PosE_{load}, NegE_{load}, Native_{load}\}, \quad (2.95)$$

де E_{load} – еталонне значення параметра навантаження на ось;

$PosE_{load}$ – значення позитивної похибки параметра навантаження на ось;

$NegE_{load}$ – значення негативної похибки параметра навантаження на ось;

$Native_{load}$ – дійсне значення параметра навантаження на ось.

$$\log_m = Native_{mb} \wedge Native_{kb} \wedge Native_{ho} \wedge Native_{cli} \wedge Native_{load}, \quad (2.96)$$

Виходячи з формул (2.83-2.96), можна отримати повну математичну модель модуля з'єднання шасі та кузова, яка згід загальний вигляд математичної моделі модуля приведено в формулі (2.97):

$$M_m = (S_{mb} \wedge S_{kb} \wedge S_{ho} \wedge S_{cli} \wedge S_{load} \wedge \log_m) \vee [Off_{M_m}, On_{M_m}] \vee Ch_{mb} \wedge Ch_{kb} \wedge Ch_{ho} \wedge Ch_{cli} \wedge Ch_{load}, \quad (2.97)$$

Математична модель модуля контролю якості наведена в формулі (2.98):

$$M_q = Mo_q \vee FN_q, \quad (2.98)$$

$$Mo_q = S_{bq} \wedge S_{eq} \wedge S_{iq} \wedge S_{mq} \wedge S_{sq} \wedge \log_q, \quad (2.99)$$

де FN_q – набір бульових значень параметрів, який описується формулою (2.100);

S_{bq} – набір параметрів які описують параметр точність встановлення кузова, який описується формулою (2.101);

S_{eq} – набір параметрів, які описують крен кузова, який описується формулою (2.102);

S_{iq} – набір параметрів точності співпадіння опорних отворів, який описується формулою (2.103);

S_{mq} – набір параметрів дорожнього просвіту, який описується формулою (2.104);

S_{sq} – набір параметрів навантаження на осі, який описується формулою (2.105);

\log_q – набір залогованих параметрів для кожної одиниці випущеного продукту, який описується формулою (2.106).

$$FN_q = \{Off_{M_q}, On_{M_q}\}, \quad (2.100)$$

де Off_{M_q} – значення для вимкнення модуля збірки двигуна;

On_{M_q} – значення для запуску функціонування модуля двигуна.

Враховуючи те, що параметри даного модуля не можуть мати похобок, а мають строго визначені стани, а саме «так» чи «ні», то набори параметрів будуть складатися лише з параметрів еталону та дійсного значення.

$$S_{bq} = \{E_{bq}, Native_{bq}\}, \quad (2.101)$$

де E_{bq} – еталонне значення параметра допустимості значень модуля збірки кузова;

$Native_{bq}$ – дійсне значення параметра допустимості значень модуля збірки кузова.

$$S_{eq} = \{E_{eq}, Native_{eq}\}, \quad (2.102)$$

де E_{eq} – еталонне значення параметра допустимості значень модуля збірки двигуна;

$Native_{eq}$ – дійсне значення параметра допустимості значень модуля збірки двигуна.

$$S_{iq} = \{E_{iq}, \text{Native}_{iq}\}, \quad (2.103)$$

де E_{iq} – еталонне значення параметра допустимості значень модуля збірки салону;
 Native_{iq} – дійсне значення параметра допустимості значень модуля збірки салону.

$$S_{mq} = \{E_{mq}, \text{Native}_{mq}\}, \quad (2.104)$$

де E_{mq} – еталонне значення параметра допустимості значень модуля з'єднання шасі та кузова;
 Native_{mq} – дійсне значення параметра допустимості значень модуля з'єднання шасі та кузова.

$$S_{sq} = \{E_{sq}, \text{Native}_{sq}\}, \quad (2.105)$$

де E_{sq} – еталонне значення параметра допустимості значень модуля збірки шасі;
 Native_{sq} – дійсне значення параметра допустимості значень модуля збірки шасі.

$$\log_q = \text{Native}_{bq} \wedge \text{Native}_{eq} \wedge \text{Native}_{iq} \wedge \text{Native}_{mq} \wedge \text{Native}_{sq}, \quad (2.106)$$

Виходячи з формул (2.99-2.106), можна отримати загальну математичну модель модуля контролю якості, яка згідно формулі (2.98), являє собою суму параметрів описаних в формулах (2.99-2.101), повний вигляд математичної моделі модуля приведено в формулі (2.107):

$$M_q = (S_{bq} \wedge S_{eq} \wedge S_{iq} \wedge S_{mq} \wedge S_{sq} \wedge \log_q) \vee [\text{Off}_{M_q}, \text{On}_{M_q}] \vee \text{Ch}_{bq} \wedge \text{Ch}_{eq} \wedge \text{Ch}_{iq} \wedge \text{Ch}_{mq} \wedge \text{Ch}_{sq}, \quad (2.107)$$

Провівши формалізацію та розроблення математичного забезпечення, яке складатється з окремих математичних моделей кожного процесу, отримаємо загальну математичну модель всього виробництва, яка на наступних етапах буде слугувати для розробки алгоритму та програмного забезпечення, які реалізують цю моделі формула (2.108):

$$P = Ma_b \wedge Ma_e \wedge Ma_i \wedge Ma_s \wedge Ma_m \wedge Ma_q \vee M(P) \vee FN_p. \quad (2.108)$$

2.3 Висновки до розділу 2

Використовуючи теоретичні та емпіричні дослідження, системний аналіз, формалізацію проведено обґрунтований вибір етапів розробки системи.

Обґрунтовані факти, які доводять, що використання централізованої системи координації підприємством є більш оптимальним за рахунок зменшення вірогідності випуску браку, та збільшення кількості своєчасних відключень модулів під час відмов.

Централізована система має перевагу над класичною схемою за випущеним браком на 16,6%, що дає змогу значно збільшити прибуток підприємства та може запобігти пошкодженню техніки, так як теоретична кількість вчасних відключень модулів при виникненні небажаних ситуацій збільшена на більш ніж 15 %, що свідчить про достатньо гарну реакцію засобу на виникнення аварійних ситуацій.

Було розроблено модель підприємства для системи, яка розробляється, описано її складові частини та приведено загальну математичну модель модуля підприємства на основі якої буде розроблятися програмний засіб використаний в системі.

Використавши декомпозиції модулів, було сформовано опис процесів, які відбуваються кожному окремому модулю та розроблено математичну модель кожного модулю.

На основі, вище описаного, було проведено розроблення математичної моделі всього виробництва, яка на наступних етапах буде слугувати для розробки алгоритму та програмного забезпечення, які реалізують цю моделі. Складовою узагальненої моделі є математичні описи кожного окремого модуля та його параметри та залежності між ними.

РОЗДІЛ 3

РОЗРОБКА КОМП'ЮТЕРНО-ІНТЕГРОВАНОЇ СИСТЕМИ УПРАВЛІННЯ
ДЛЯ КООРДИНАЦІЇ ФУНКЦІОНУВАННЯ ПІДСИСТЕМ ПІДПРИЄМСТВА

3.1 Розробка схеми комп'ютерно-інтегрованої системи управління для координації функціонування підсистем підприємства

Комп'ютерно-інтегрована система для координації функціонування підприємства, є досить складною. Будь-яка комп'ютерно-інтегрована система має структуру яка включає в себе базові рівні системи такі як: визначені функції системи (ФЗ), інформаційне (ІЗ), математичне (МЗ), алгоритмічне (АЗ), програмне (ПЗ) та технічне (ТЗ) забезпечення, данна схема представлена на рис. 3.1.

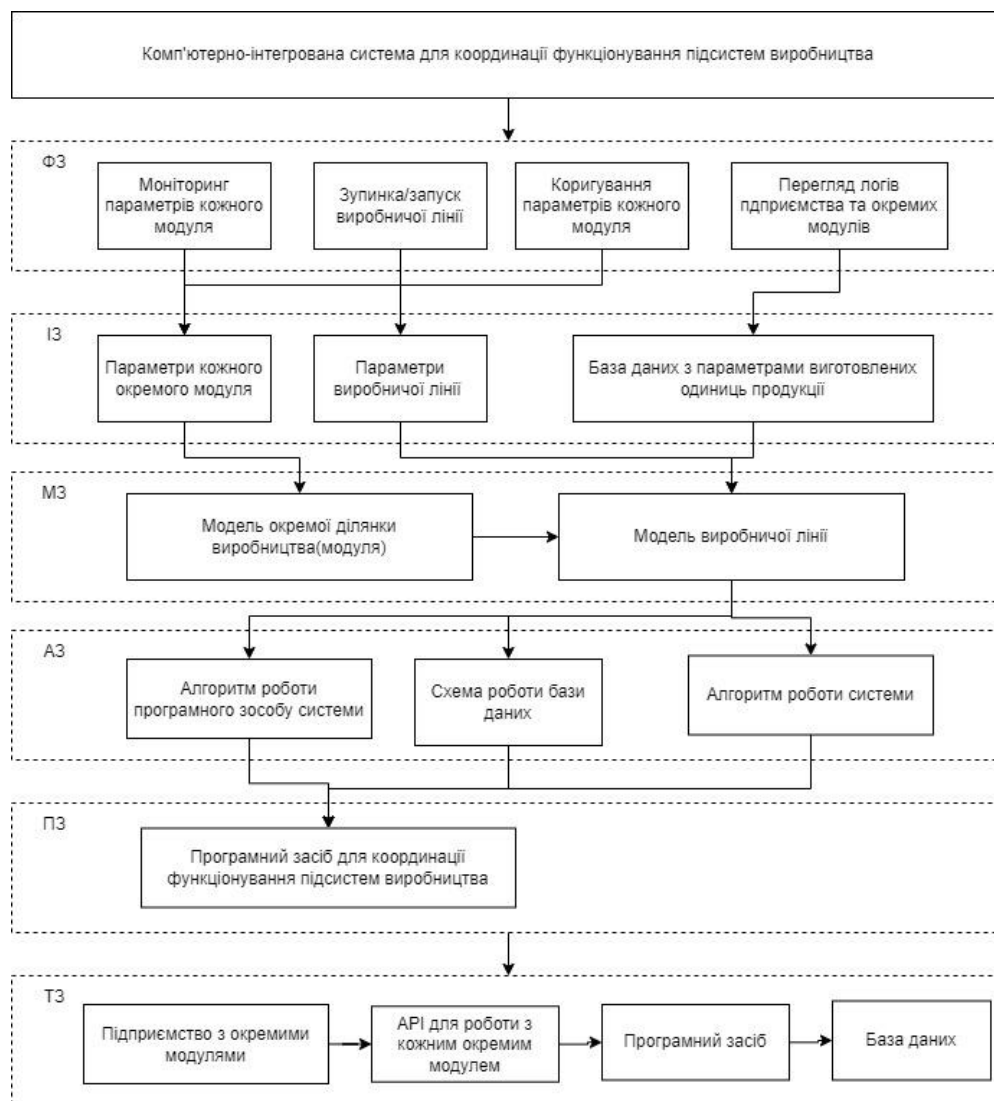


Рисунок 3.1 – Структурна схема комп'ютерно-інтегрованої системи для координації функціонування підсистем

В загальному вигляді систему такого типу можна представити у вигляді модулю підприємства, на який поступає інформація зі значеннями параметрів на основі даних яких – проводиться координація.

API – підпрограма яка перетворює бінарні сигнали модуля в числові значення параметру, а саме в координати знаходження підмодуля (наприклад, виконавчого автоматизованого пристрою, роботу, маніпулятору тощо) та відповідає за передачу даних в програму, яка в свою чергу, поєднана з базою даних, де вноситься та зберігаються дані параметрів кожного окремого модуля – ділянки (технологічної операції). На рисунку 3.2 наведено узагальнену схему, що була покладена в основу розроблення для комп'ютерно-інтегрованої системи координації функціонування підприємства.

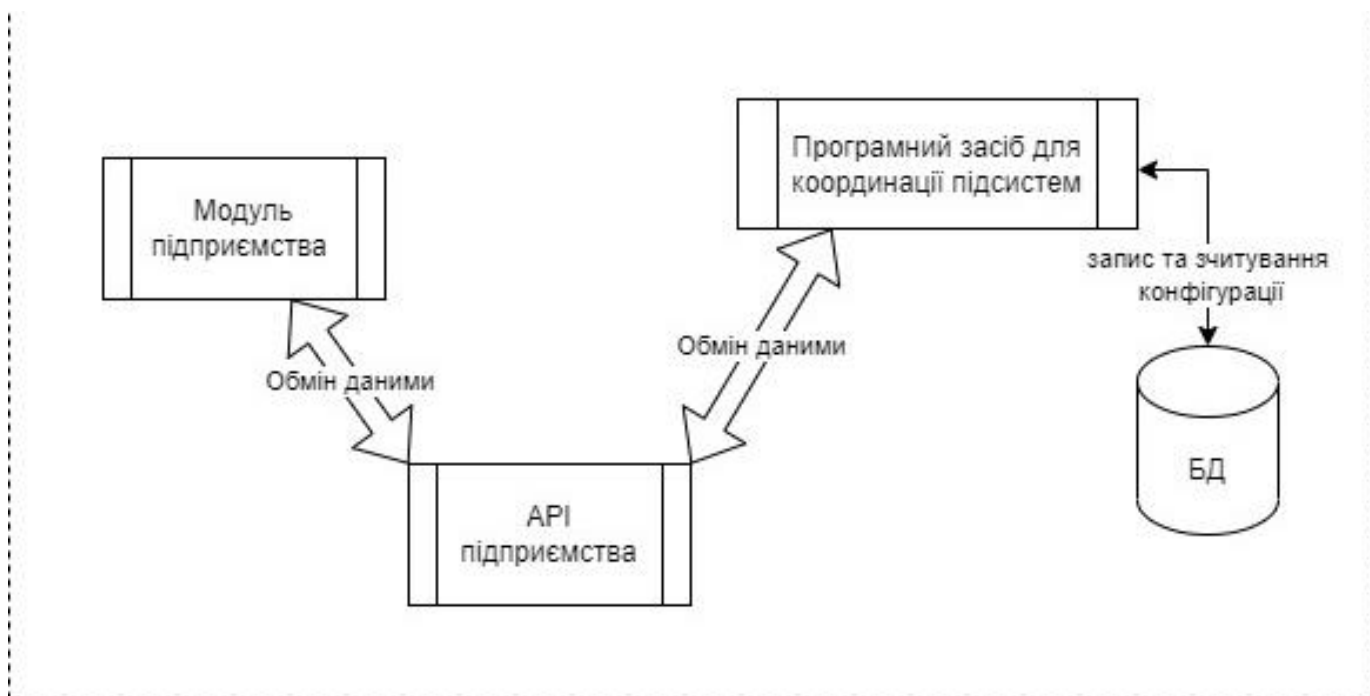


Рисунок 3.2 – Узагальнена схема роботи системи

API підприємства також має змогу перераховувати змінені одиниці параметрів в певні координати, за допомогою яких під модулі орієнтуються в просторі, що дає змогу змінити точність встановлення деталей або контроль якості цих деталей.

На рис. 3.2 зображено зв'язок бази даних з програмним засобом, який показує, що програмний засіб матиме контекст зав'язаний на базі даних, а не статичний контекст як в більшості програм.

Програмний засіб представляє собою певний інтерфейс на якому необхідно відобразити інформацію, щодо параметрів певного модуля, на основі яких користувачи (оператор) може приймати рішення щодо працездатності та коректності роботи кожного

окремого модуля чи технологічної операції (ділянки). У випадку виникнення аварійних (нештатних) ситуацій програмний засіб запрограмований на вимкнення лінії, або у випадку програмного збою оператор може зробити це власноруч.

За допомогою програмного засобу можна збільшити гнучкість виробництва за рахунок програмного переналагоджування в меню змінити параметри модуля. Система має змогу відстежити кожну одиницю випущеного продукту за рахунок системи логування параметрів на кожному етапі виробництва.

Так як написання API є складним та ресурсозатратним для кожного підприємства API відрізняється так як кожне підприємство використовує різне обладнання і для підключення до програмного засобу необхідна адаптація, тобто зробити універсальну під програму дуже складно, виходячи з цього розробка програмного засобу для координації є актуальною задачею. Нижче приведено розробка програмного засобу його алгоритм роботи та схема БД.

3.2 Вибір середовища розробки

Для розробки програмного засобу для координації функціонування підсистемам підприємства, з точки зору використання ПЗ на настільному персональному комп'ютері або на лептопі, доцільно вибирати поміж середовищ призначених для розробки програмних засобів для настільних комп'ютерів.

Для стабільної роботи програми та можливого розширення функціональних можливостей в майбутньому та підтримки, доцільно вибрати мову програмування, яку будуть підтримувати найближчий час та відповідну платформу.

З точки зору підтримки та довговічності програми найбільш актуальними на даний момент для розробки ПЗ для настільних комп'ютерів це платформа .NET та мова програмування C# так як версія платформи .NET 5.0 буде підтримуватися до 2025 року, в додачу до цього зовсім недавно від дати публікації кваліфікаційної роботи вийшов реліз версії .NET 5.0 та C# 10.0 термін завершення підтримки досі не було вказано, що свідчить про те що найближчі 5 – 7 років програма буде легко розширюватися та матиме велику кількість спеціалістів для цього.

На платформі .NET є декілька найбільш популярних інструментів для розробки засобів для настільних комп'ютерів а саме:

- Xamarin.Forms;

- Windows Forms;
- Unity engine + .NET;
- WPF;
- MAUI [65].

Xamarin.Forms — це фреймворк інтерфейсу користувача з відкритим вихідним кодом. Xamarin.Forms дозволяє розробникам створювати програми Xamarin.Android, Xamarin.iOS і Windows з єдиної спільної кодової бази [65].

Xamarin.Forms дозволяє розробникам створювати користувацькі інтерфейси в XAML із кодом на C#.

Ці інтерфейси відображаються як ефективні власні елементи керування на кожній платформі[65].

Xamarin.Forms надає послідовний API для створення елементів інтерфейсу користувача на різних платформах.

Цей API може бути реалізований в XAML або C# і підтримує прив'язування даних для таких шаблонів, як Model-View-ViewModel (MVVM) [65].

Під час виконання Xamarin.Forms використовує засоби візуалізації платформи для перетворення між платформних елементів інтерфейсу користувача у власні елементи керування на Xamarin.Android, Xamarin.iOS та UWP.

Це дозволяє розробникам отримати оригінальний вигляд, відчуття та продуктивність, усвідомлюючи переваги спільного використання коду між платформами [55].

Програми Xamarin.Forms зазвичай складаються із спільної бібліотеки .NET Standard та окремих проектів платформи.

Спільна бібліотека містить представлення XAML або C# і будь-яку бізнес-логіку, наприклад послуги, моделі або інший код.

Проекти платформи містять будь-яку специфічну для платформи логіку або пакети, які потрібні додатку.

Xamarin.Forms має велику екосистему бібліотек, які додають різноманітні функції до програм [65].

Xamarin.Forms надає загальний API, який надає власні елементи керування на різних платформах, але конкретна платформа може мати функціональні можливості, які не існують на інших платформах.

Наприклад, платформа Android має вбудовану функціональність для швидкого прокручування в ListView, а iOS – ні.

Специфіка платформи Xamarin.Forms дозволяє використовувати функціональні можливості, які доступні лише на певній платформі, без створення користувацьких засобів візуалізації або ефектів [62].

Windows Forms – це структура інтерфейсу користувача для створення настільних програм Windows.

Він забезпечує один з найпродуктивніших способів створення настільних програм на основі візуального дизайнера, наданого у Visual Studio.

Такі функціональні можливості, як розміщення візуальних елементів керування перетягуванням, дозволяють легко створювати настільні програми.

За допомогою Windows Forms розробляються графічні програми, які легко розгортати, оновлювати та працювати в автономному режимі або підключені до Інтернету.

Програми Windows Forms можуть отримати доступ до локального обладнання та файлової системи комп'ютера, на якому працює програма [66].

Windows Forms – це технологія інтерфейсу користувача для .NET, набору керованих бібліотек, які спрощують звичайні завдання програми, такі як читання та запис у файлову систему.

Коли ви використовуєте середовище розробки, як наприклад Visual Studio, можна створювати інтелектуальні клієнтські програми Windows Forms, які відображають інформацію, запитують введення від користувачів і спілкуються з віддаленими комп'ютерами через мережу.

У Windows Forms форма – це візуальна поверхня, на якій відображається інформацію для користувача. Зазвичай створюються програми Windows Forms, додаючи елементи керування до форм і розробляючи відповіді на дії користувача, наприклад клацання мишею або натискання клавіш.

Елемент керування – це дискретний елемент інтерфейсу користувача, який відображає дані або приймає введення даних.

Коли користувач щось робить з вашою формою або одним із елементів керування, ця дія створює подію. Додаток реагує на ці події за допомогою коду та обробляє події, коли вони відбуваються.

Windows Forms містить різноманітні елементи керування, які можна додати до форм а саме [66]:

- елементи керування, які відображають текстові поля;
- кнопки;

- спадні вікна;
- перемикач;
- веб-сторінки і т.д. [67]

Якщо наявний елемент керування не відповідає вашим потребам, Windows Forms також підтримує створення власних спеціальних елементів керування за допомогою класу `UserControl` [65].

Windows Forms має багаті елементи керування інтерфейсом користувача, які імітують функції високоякісних програм, таких як Microsoft Office.

Коли використовуються елементи керування `ToolStrip` і `MenuStrip`, можна створювати панелі інструментів і меню, які містять текст і зображення, відображати підменю та розміщувати інші елементи керування, наприклад текстові поля та поля зі списком [66].

За допомогою конструктора Windows Forms з перетягуванням у Visual Studio можна легко створювати програми Windows Forms.

Просто необхідно обрати елементи керування за допомогою курсора та розташуйте їх у потрібному місці на формі.

Конструктор надає такі інструменти, як лінії сітки та лінії прив'язки, щоб уникнути клопоту з вирівнюванням елементів керування.

Також можна використовувати елементи керування `FlowLayoutPanel`, `TableLayoutPanel` і `SplitContainer` для створення розширених макетів форм за менший час.

Нарешті, якщо потрібно створити власні елементи інтерфейсу користувача, простір імен `System.Drawing` містить великий вибір класів для відтворення ліній, кіл та інших фігур безпосередньо на формі [65].

Багато програм повинні відображати дані з бази даних, файлів XML або JSON, веб - служби чи іншого джерела даних.

Windows Forms надає гнучкий елемент керування, який називається `DataGridView`, для відображення таких табличних даних у традиційному форматі рядків і стовпців, так що кожна частина даних займає власну клітинку.

Використовуючи `DataGridView`, ви можете налаштувати зовнішній вигляд окремих комірок, блокувати довільні рядки і стовпці на місці, а також відображати складні елементи керування всередині клітинок, серед інших функцій [67].

Підключення до джерел даних через мережу – це просте завдання з Windows Forms.

Компонент `BindingSource` представляє з'єднання з джерелом даних і надає методи прив'язки даних до елементів керування, переходу до попереднього та наступного записів, редагування записів і збереження змін до вихідного джерела.

Елемент керування `BindingNavigator` забезпечує простий інтерфейс через компонент `BindingSource`, щоб користувачі могли переміщатися між записами.

Можна легко створювати елементи керування, пов'язані з даними, використовуючи вікно «Джерела даних» у `Visual Studio`.

У вікні відображаються такі джерела даних, як бази даних, веб-сервіси та об'єкти у проєкті. Також можна створювати елементи керування, пов'язані з даними, перетягуючи елементи з цього вікна на форми у проєкті.

Також можна зв'язати наявні елементи керування з даними, перетягуючи об'єкти з вікна джерел даних на наявні елементи керування [67].

Іншим типом прив'язки даних, яким можна керувати в `Windows Forms`, є налаштування. Більшість програм повинні зберігати деяку інформацію про стан під час виконання, наприклад останній відомий розмір форм, і зберігати дані про налаштування користувача, наприклад розташування за умовчанням для збережених файлів.

Функція, параметри програми відповідають цим вимогам, забезпечуючи простий спосіб збереження обох типів налаштувань на клієнтському комп'ютері. Після визначення цих параметрів за допомогою `Visual Studio` або редактора коду вони зберігаються як XML і автоматично зчитуються в пам'ять під час виконання [66].

`Unity` – це 2D/3D-движок і фреймворк, який представляє систему для проєктування ігрових сцен або сцен програми для 2D, 2.5D і 3D.

На `Unity` можна розробляти не лише ігри, а й навчальні симулятори, додатки першого реагування та інші бізнес-додатки, розроблені за допомогою `Unity`, які потребують взаємодії з 2D/3D простором.

`Unity` дозволяє взаємодіяти з ними за допомогою не лише коду, а й візуальних компонентів, а також експортувати їх на будь-яку велику мобільну платформу та багато іншого – безкоштовно. Також є професійна версія, яка дуже гарна, але вона не безкоштовна. З безкоштовною версією можна зробити вражаючу кількість. `Unity` підтримує всі основні 3D-додатки та багато аудіо форматів і навіть розуміє формат `Photoshop .psd`, щоб можна просто перемістити файл `.psd` в проєкт `Unity`. `Unity` дозволяє імпортувати та збирати активи, писати код для взаємодії з об'єктами, створювати або імпортувати анімації для використання з розширеною системою анімації та багато іншого [68].

Unity виконала роботу, щоб забезпечити між платформну підтримку, і можна змінювати платформи буквально одним клацанням миші, хоча чесно кажучи, зазвичай потрібні мінімальні зусилля, наприклад, інтеграція з кожним магазином для покупок у програмі [66].

Мабуть, найпотужнішою частиною Unity є Unity Asset Store, можливо, найкращий ринок активів на ринку ігор. У ньому можна знайти всі ігрові компоненти, такі як ілюстрації, 3D-моделі, файли анімації для 3D-моделей, аудіо ефекти та повні треки, плагіни, у тому числі такі, як набір інструментів MultiPlatform, який може допомогти з підтримкою кількох платформ – системи візуальних сценаріїв, такі як PlayMaker і Behave, розширені шейдери, текстури, ефекти частинок тощо.

Інтерфейс Unity повністю підтримує сценарії, що дозволяє багатьом стороннім плагінам інтегруватися прямо в графічний інтерфейс Unity.

Більшість, якщо не всі, професійні розробники ігор використовують ряд пакетів із магазину активів, і якщо у вас є щось гідне запропонувати, ви також можете опублікувати це там [69].

Однак Unity за замовчуванням не є системою для розробки 2D-об'єктів і 3D-моделей (за винятком рельєфу). Існують плагіни конструктора 2D-світу, такі як 2D-редактор ландшафту, для створення 2D-моделей, а також можна створювати ландшафти всередині Unity, використовуючи їхні інструменти ландшафту, щоб створювати дивовижні ландшафти з деревами, травою, горами тощо [68].

WPF розшифровується як Windows Presentation Foundation. Це потужний фреймворк для створення додатків Windows.

WPF спочатку був представлений у версії .NET Framework 3.0, а потім багато інших функцій було додано в наступних версіях .NET Framework [67].

До WPF інші фреймворки користувальницького інтерфейсу, пропонувані Microsoft, такі як форми MFC та Windows, були лише обгортками навколо DLL User32 та GDI32, але WPF лише мінімально використовує User32. WPF – це більше, ніж просто обгортка це частина платформи .NET. Він містить суміш керованого та некерованого коду. Основні компоненти архітектури WPF представлені нижче. Найважливішою частиною коду WPF є:

- структура презентації;
- ядро презентації;
- мілкор [68].

Структура презентації та ядро презентації були написані в керованому коді. *Milcore* – це частина некерованого коду, яка забезпечує тісну інтеграцію з *DirectX* (відповідає за відображення та візуалізацію). CLR робить процес розробки більш продуктивним, пропонуючи безліч функцій, таких як керування пам'яттю, обробка помилок тощо [67].

У попередніх фреймворках графічного інтерфейсу не було реального поділу між тим, як виглядає програма і як вона веде себе. І графічний інтерфейс, і поведінка були створені однією мовою, напр. *C#* або *VB.Net*, що вимагало б від розробника більше зусиль для реалізації інтерфейсу користувача та поведінки, пов'язаної з ним.

У WPF елементи інтерфейсу користувача розроблені в XAML, тоді як поведінка може бути реалізована на процедурних мовах, таких як *C#* і *VB.Net*. Тому дуже легко відокремити поведінку від коду конструктора.

З XAML програмісти можуть працювати паралельно з дизайнерами. Розділення між графічним інтерфейсом користувача та його поведінкою може дозволити легко змінити зовнішній вигляд елемента керування за допомогою стилів і шаблонів.

Хоча даний фреймворк і має певні великі переваги проте підтримка даного фреймворку а ІТ спільності зменшується та в найближчі часи на ринку може з'явитися дефіцит спеціалістів даного напрямку.

.NET Multi-platform App UI (.NET MAUI) – це крос платформний фреймворк для створення нативних мобільних і настільних додатків із *C#* і XAML.

Використовуючи *.NET MAUI*, ви можете розробляти програми, які можуть працювати на *Android*, *iOS*, *macOS* та *Windows* з єдиної спільної бази коду.

.NET MAUI є відкритим вихідним кодом і є еволюцією *Xamarin.Forms*, розширеною від мобільних до настільних сценаріїв, з елементами керування інтерфейсом, перебудованими з нуля для підвищення продуктивності та розширюваності. Якщо раніше використовували *Xamarin.Forms* для створення між платформних інтерфейсів користувача, можна помітите багато подібності з *.NET MAUI*. Однак є й деякі відмінності. Використовуючи *.NET MAUI*, можна створювати багато платформні програми за допомогою одного проекту, але за потреби можна додати вихідний код і ресурси для певної платформи. Однією з ключових цілей *.NET MAUI* є можливість реалізувати якомога більшу частину логіки програми та макета інтерфейсу користувача в одній базі коду.

.NET MAUI об'єднує API Android, iOS, macOS і Windows в єдиний API, що дозволяє розробнику виконувати одноразовий запуск і де завгодно, а також забезпечує глибокий доступ до кожного аспекту кожної рідної платформи.

.NET 6 надає ряд платформ для створення програм: .NET для Android, .NET для iOS, .NET для macOS і бібліотека Windows UI 3 (WinUI 3). Усі ці фреймворки мають доступ до однієї бібліотеки базових класів .NET 6 (BCL). Ця бібліотека абстрагує деталі базової платформи від коду. BCL залежить від середовища виконання .NET, щоб забезпечити середовище виконання для коду. Для Android, iOS і macOS середовище реалізовано Mono, реалізацією середовища виконання .NET. У Windows Win32 забезпечує середовище виконання [69].

Хоча BCL дозволяє програмам, що працюють на різних платформах, спільно використовувати спільну бізнес-логіку, різні платформи мають різні способи визначення інтерфейсу користувача для програми, і вони надають різні моделі для визначення того, як елементи інтерфейсу користувача взаємодіють і взаємодіяли. Також можна створити інтерфейс користувача для кожної платформи окремо, використовуючи відповідну платформу (.NET для Android, .NET для iOS, .NET для macOS або WinUI 3), але цей підхід вимагає від вас підтримувати базу коду для кожного окремого сімейства пристроїв.

.NET MAUI надає єдину структуру для створення інтерфейсу для мобільних і настільних програм. У програмі .NET MAUI пишеться код, який переважно взаємодіє з API .NET MAUI. Потім .NET MAUI безпосередньо споживає власні API платформи. Крім того, код програми може безпосередньо використовувати API платформи, якщо потрібно.

Програми .NET MAUI можна писати на ПК або Mac і компілювати у власні пакети програм [67-69]:

Програми Android, створені з використанням .NET MAUI, компілюються з C# на проміжну мову (IL), яка потім компілюється вчасно (JIT) у рідну збірку під час запуску програми [69]. Програми iOS, створені з використанням .NET MAUI, повністю завчасно (AOT) компілюються з C# у рідний код збірки ARM.

Програми macOS, створені за допомогою .NET MAUI, використовують Mac Catalyst, рішення від Apple, яке переносить програму iOS, створену за допомогою UIKit, на робочий стіл і доповнює її додатковими API-інтерфейсами AppKit і платформи, якщо потрібно. Програми Windows, створені з використанням .NET MAUI, використовують

бібліотеку Windows UI 3 (WinUI 3) для створення рідних програм, які націлені на робочий стіл Windows [68].

.NET MAUI надає набір елементів керування, які можна використовувати для відображення даних, ініціювання дій, індикації активності, відображення колекцій, вибору даних тощо. На додаток до набору елементів керування, .NET MAUI також надає:

- складний механізм макетування для розробки сторінок;
- кілька типів сторінок для створення різноманітних типів навігації, наприклад ящиків;
- підтримка зв'язування даних, для більш елегантних і зручних шаблонів розробки;
- можливість налаштовувати обробники для покращення способу представлення елементів інтерфейсу користувача. Основні між платформні API для доступу до властивих функцій пристрою. Ці API дозволяють програмам отримувати доступ до таких речей, як GPS, акселерометр, а також стан акумулятора та мережі.

Між платформна графічна бібліотека, яка надає загальний API для націлювання на кілька платформ, що дає змогу ділитися своїм кодом 2D-малювання між платформами або змішувати та поєднувати графічні реалізації в одній програмі.

Система одного проекту, яка використовує багатоцільове націлювання для націлювання на Android, iOS, macOS та Windows.

.NET гаряче перезавантаження, для змінити як XAML, так і керований вихідний код під час роботи програми, а потім спостерігати за результатами змін, не перебудовуючи програму [66]. Підсумовуючи вище згадані факти можна зробити висновок про те, що Xamarin.Forms має певні переваги для розробки між платформових програмних засобів, але не для вузько профільної програми яка буде використовуватися тільки на настільних комп'ютерах.

Windows Forms є достатньо простим у використанні та багатофункціональним не є фреймворком для розробки кросплатформових програмних засобів, але має багатий функціонал та достатньо багатий набір графічних елементів для використання їх при створенні графічного інтерфейсу. Unity engine + .NET гарний фреймворк який заточений більше для розробки ігор або проектів які містять 3-D графіку таких як симулятори та інші програми для симуляції будь-яких об'єктів, що не дуже підходить темі даної програми.

WPF має переваги в паралельній розробці для створення одночасно інтерфейсу та логічної частини програми, так як паралельна розробка в даному контексті не важлива то більшість переваг зводяться до нуля. .NET MAUI має значні переваги над всіма вище перерахованими фреймворками, але має велику долю ризику по причині раннього релізу та можливих ризиків недоробок фреймворку [68].

Прийнявши всю аргументацію згадану в попередніх п'яти абзацах можна сказати, що для вузько направленою програмного засобу найбільш підходящим фреймворком являється Windows Forms на якому і буде написаний даний програмний засіб [69].

3.3 Розробка алгоритмічного забезпечення комп'ютерно-інтегрованої системи

Алгоритм даного продукту представляє собою послідовність дій які користувач потенційно буде виконувати, а саме:

- логування;
- вибір модуля для перегляду;
- моніторинг параметрів модуля;
- координація модулем.

В свою чергу існує 6 модулів які користувач може обрати а саме: модуль збірки кузова, модуль збірки шасі, модуль збірки салону, модуль збірки двигуна, модуль з'єднання кузова та шасі, модуль контролю якості.

В кожному із модулів існує декілька параметрів як правило від 5-8 в демонстраційних цілях, за потреби модулі можна декомпонувати на під модулі та збільшити кількість параметрів від 5 до 20-40, більша кількість параметрів потребує виділення під модулів, що призведе до розширення алгоритму. Алгоритм приведено на рис. 3.3.

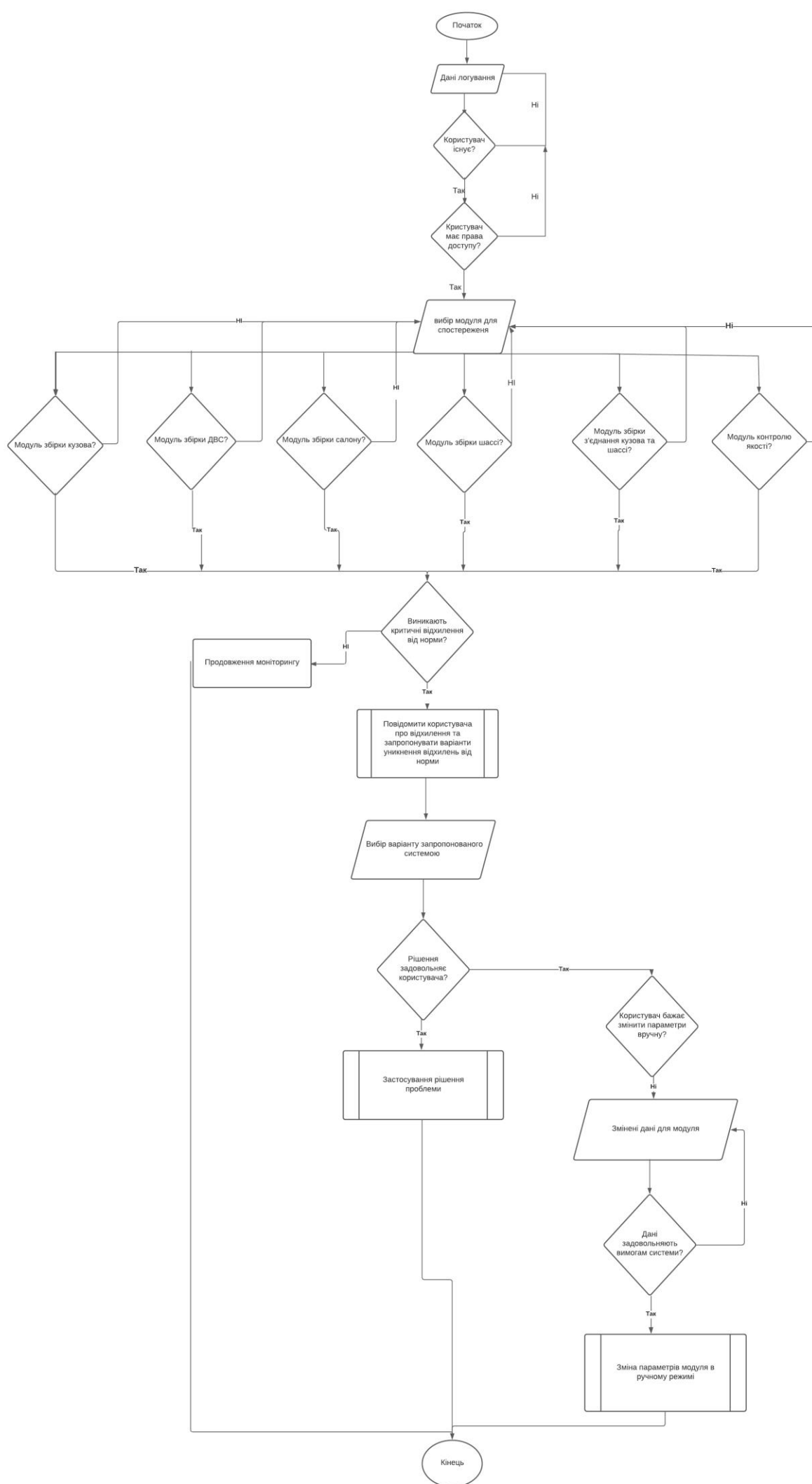


Рисунок 3.3 – Алгоритм роботи програми

3.4 Розробка схеми бази даних комп'ютерно-інтегрованої системи

Для програмного продукту було розроблено схему бази даних, яку приведено на рис. 3.4. Данна схема бази даних розроблена для використання в реляційних базах даних. Для системи координації буде використана база даних Microsoft SQL Server так як дана база даних має повну сумісність з Windows Forms так як дані продукти були розроблені однією компанією та було передбачено їх інтеграцію.

Для роботи з базою даних було використано Entity Framework та підхід Code First для створення таблиць.

Entity Framework представив підхід Code-First з Entity Framework 4.1. Code-First в основному корисний у Domain Driven Design. У підході Code-First зосереджуєтесь на домені програми(моделі) і початку створення класів для сутності домену, а не спочатку проектуєте базу даних, а потім створюєте класи, які відповідають дизайну бази даних.

Робочий процес розробки в підході з використанням коду буде таким:

- створити або змінити класи домену;
- налаштувати ці класи домену за допомогою Fluent-API або атрибутів анотації даних;
- створити або оновити схему бази даних за допомогою автоматичної міграції або міграції на основі коду.

Схема бази даних містить наступні таблиці:

- користувачі;
- модулі;
- модуль кузова;
- модуль шасі;
- модуль збірки салону;
- модуль двигуна;
- модуль з'єднання кузова та шасі;
- модуль контролю якості.

Для кожної із вище згаданих таблиць крім користувачів та модулі існують дублюючі таблиці для логування параметрів кожного модуля, що дозволить переглядати історію кожного модуля та оцінювати якість випущеної продукції.

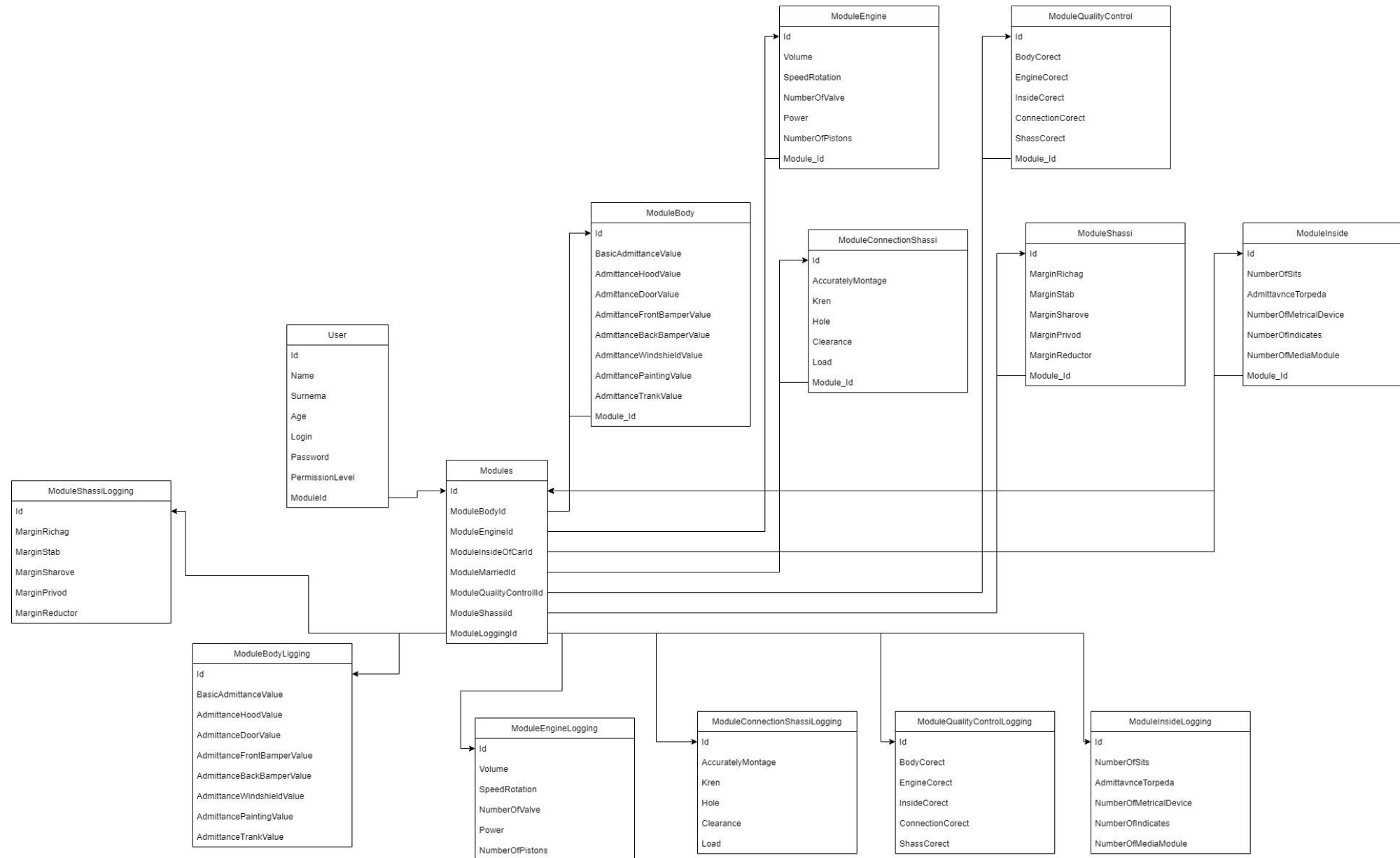


Рисунок 3.4 – Схема бази даних

3.5 Розробка програмного забезпечення комп'ютерно-інтегрованої системи

Для даного продукту було обрано монолітну форму архітектури так як виділяти кожен модуль в мікросервіс являється недоцільним через виникнення великої кількості залежностей та абстрактних рівнів які не мають практичної користі для програми. Архітектура програми приведена на рис. 3.5.

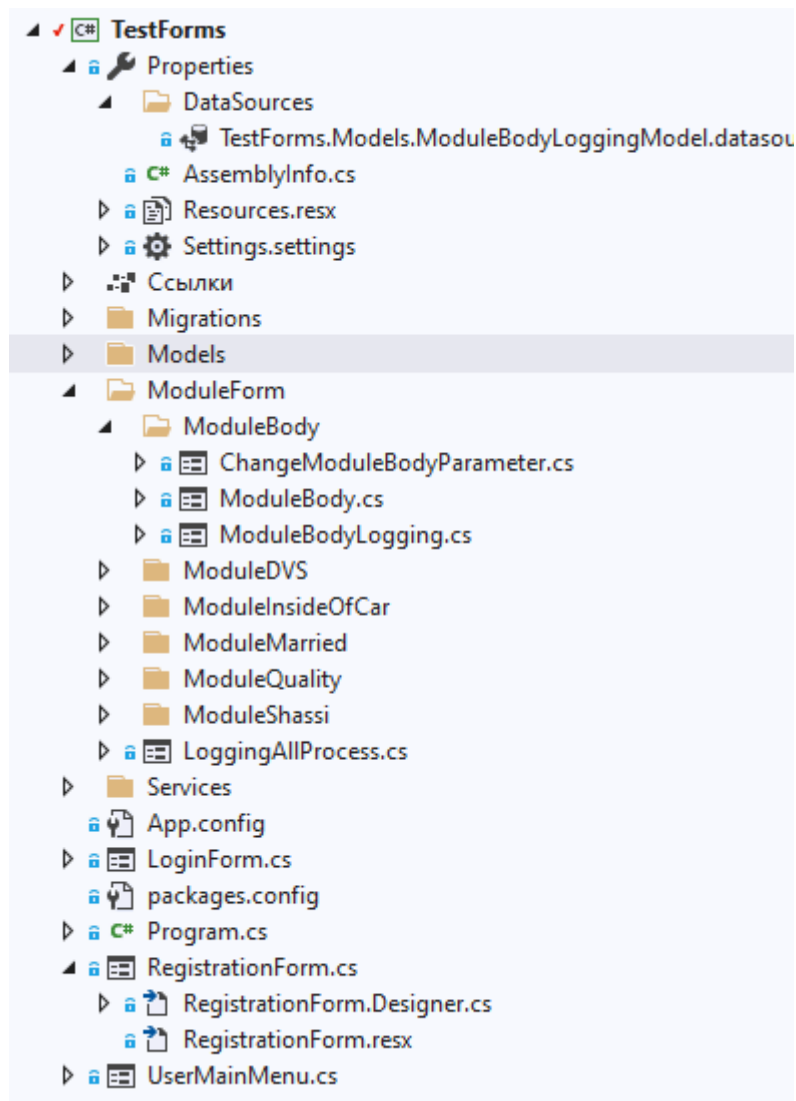


Рисунок 3.5 – Архітектура рішення

Як показано на рис. 3.5 архітектура програми вміщає в себе базові файли такі як:

- settings.settings;
- resources.resx;
- assemblyInfo.cs;
- dataSources.

Які знаходяться в папці properties. Крім базових файлів існують наступні папки:

- migrations;
- models;

- moduleForm;
- services.

Папка ModuleForm містить в собі наступні папки:

- moduleBody;
- moduleDVS;
- moduleInsideOfCar;
- moduleMarried;
- moduleQuality;
- moduleShassi.

В кожній з цих папок лежить по 3 файли відповідного модуля це файли форм логування, зміни параметрів та головної форми модул.

Крім цього окремо лежать загальні форми такі як:

- loginForm.cs;
- registrationForm.cs;
- userMainMenu.cs.

Кожен модуль має шаблону форму код програми буде приведений в додатку а.

Даний програмний продукт має систему входу приведену на рис 3.6 на якій працює валідація вхідних даних це перевірка наявності користувача в базі даних, перевірка правильності вводу логіна та пароля та перевірка рівня доступу, рівень доступу має бути не менш ніж 4 рівень, що свідчить про досвідченість робітника. Крім системи входу програма також має свою систему реєстрації(рис.3.7), кожен користувач може зареєструватися в даній програмі, в даному контексті мається на увазі що є відповідальний працівник який не допустить дискретне видання рівнів доступу користувачам.

Рисунок 3.6 – Форма входу в ПЗ

The image shows a web browser window titled "Реєстрація" (Registration). The main heading is "Реєстрація користувача" (User Registration). The form contains the following fields and elements:

- Имя (Name): text input field
- Прізвище (Surname): text input field
- Вік (Age): text input field
- Рівень доступу (Access level): text input field
- Логін (Login): text input field
- Пароль (Password): text input field
- Пароль повторно (Repeat password): text input field
- Below the password fields is a blue link: [Переглянути пароль](#) (View password)
- Below the link is a checkbox: Я згоден з правами використання програмного забезпечення (I agree with the software license terms)
- At the bottom center is a button: **Зареєструватися** (Register)
- At the bottom right is a button: **Назад** (Back)

Рисунок 3.7 – Форма реєстрації

Для реєстрації в програмі необхідно заповнити всі поля форми та коректно ввести пароль 2 рази, в разі якщо користувач допустив опомилку в паролі, то користувач може переглянути обидва паролі натиснувши на посилання «переглянути пароль».

Також, обов'язковою є заповнення поля «я згоден з правами використання програмного забезпечення» так як, компанія та/або юридична особа яка є надавачем ліцензії не несе відповідальність за втрату даних. На рис. 3.8 приведено головне меню програми.

Головне меню представляє собою – меню навігації з якого можна перейти в будь-який модуль виробництва, зупинити або запустити виробництво що супроводжується червоною приміткою в лівому верхньому вуглі(рис.3.9).

На рисунку 3.9 приведено картину після зупинки підприємства, якщо порівняти рисунки 3.10 та 3.8 то можна помітити, що кнопка «Зупинити підприємство» змінюється на «Запустити підприємство». Переглянувши логи приведені на рис 3.10 можна побачити що, статус кожного модуля змінився на «Зупинено» що свідчить про те що всі модулі були зупинені по натисканню кнопки.

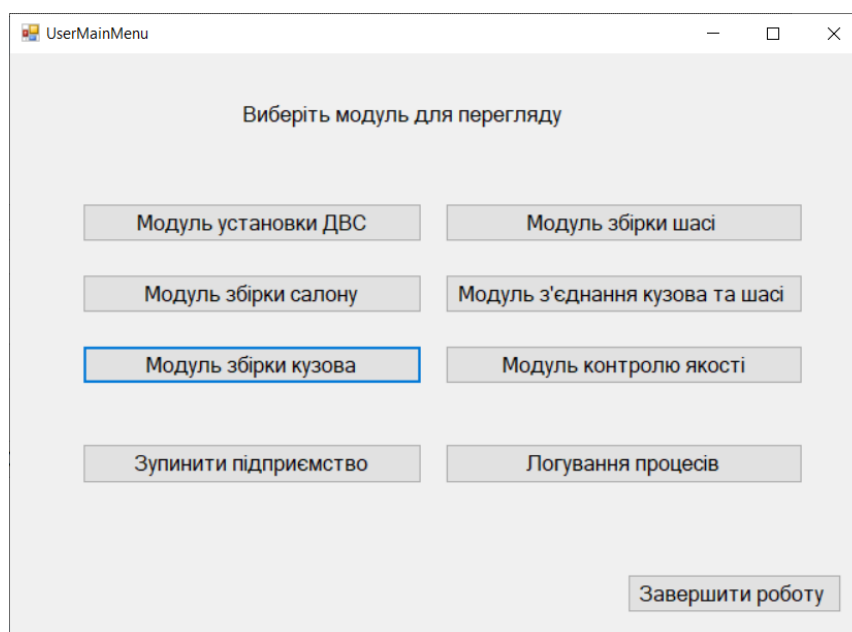


Рисунок 3.8 – Головне меню програми

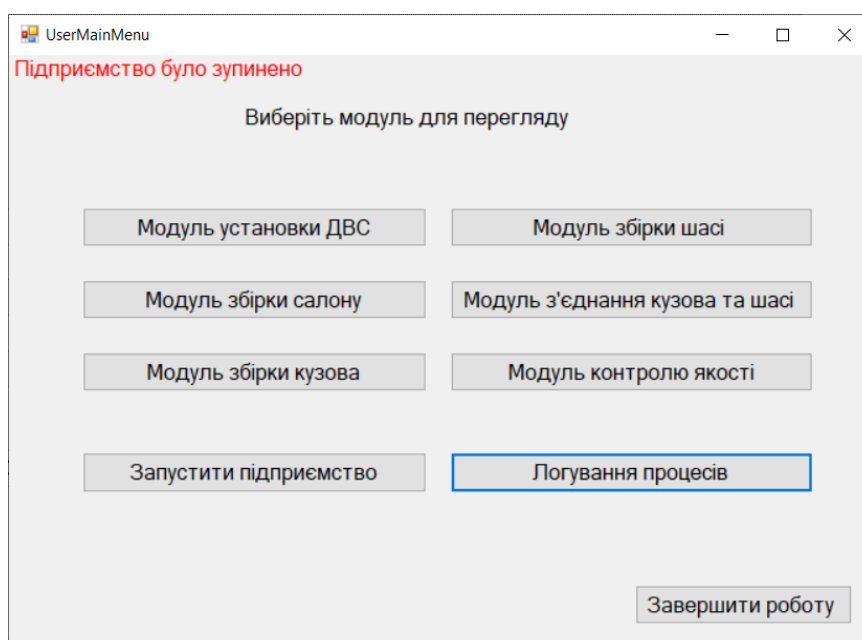


Рисунок 3.9 – Головне меню

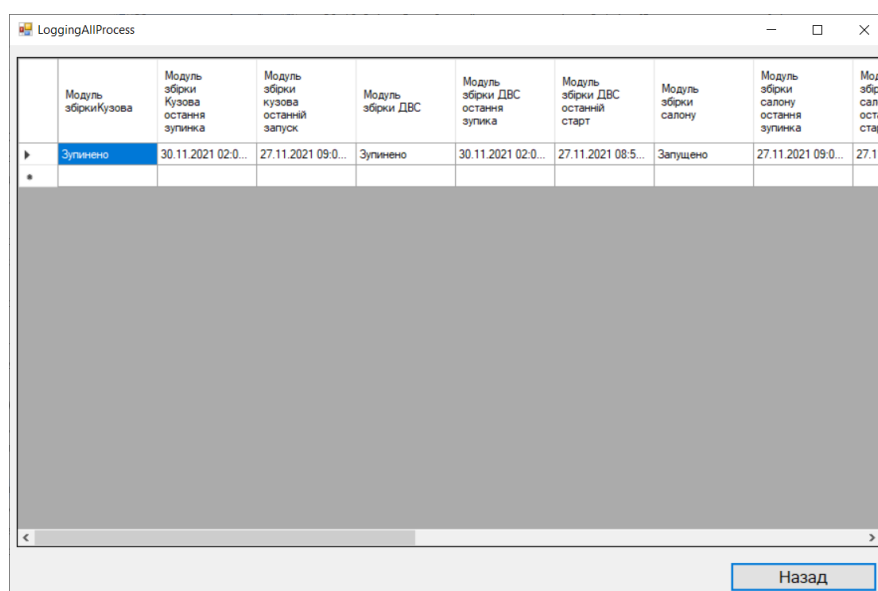


Рисунок 3.10 – Меню перегляду логів по підприємству

Крім цього в кожного модуля є ще два поля це остання зупинка модуля та останній запис, в обидвох випадках при зупинці будь-якого модуля це буде відображено в загальному логуванні програми. Після запуску підприємства з'являється червона помітка як на рис. 3.11 та кнопка знову змінюється на «Зупинити підприємство» та результат відображається як показано на рис. 3.12.

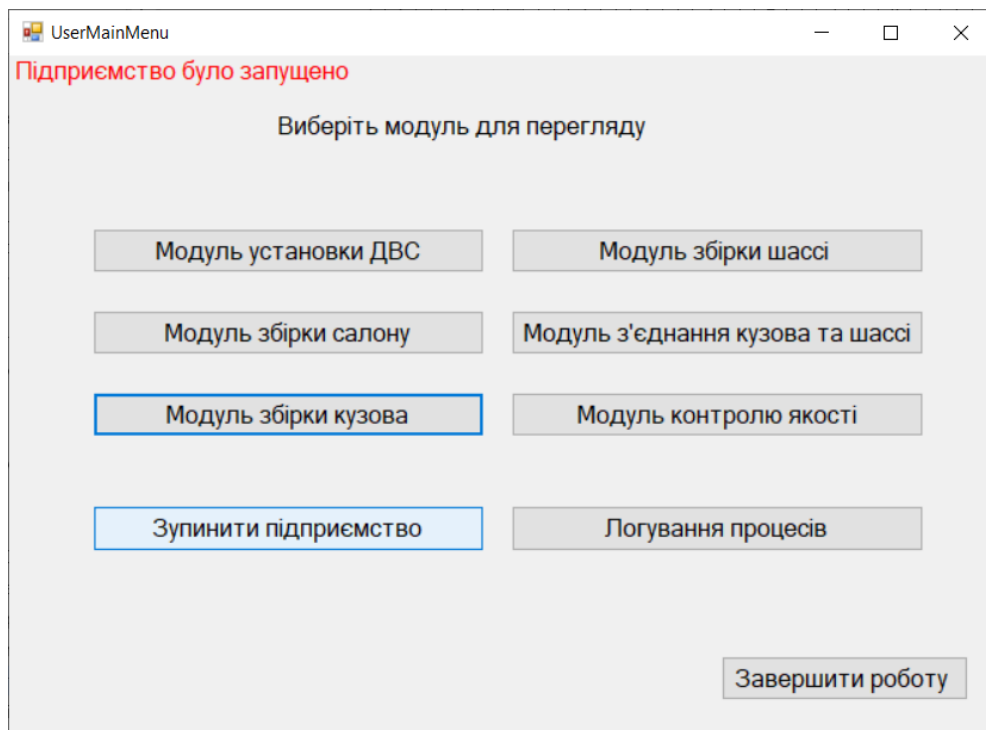


Рисунок 3.11 – Головне меню після запуску підприємства

Модуль збірки Кузова	Модуль збірки Кузова остання зупинка	Модуль збірки кузова останній запуск	Модуль збірки ДВС	Модуль збірки ДВС остання зупинка	Модуль збірки ДВС останній старт	Модуль збірки салону	Модуль збірки салону остання зупинка	Мод збір сал остан старт
Запущено	30.11.2021 02:0...	30.11.2021 02:0...	Запущено	30.11.2021 02:0...	30.11.2021 02:0...	Запущено	27.11.2021 09:0...	27.11.2021 09:0...

Рисунок 3.12 – Логування запуску підприємства

При відкритті вкладки модуля двигуна можна спостерігати наступну форму як приведено на рис. 3.13. В даному модулю для демонстрації концепції приведено 5

параметрів кожен з яких має «Еталон» бажане значення та допуски та «Дійсне значення» під останній розуміється що значення приходять з робочого модуля який з'єднаний з програмою за допомогою спеціального API. Також можна зупинити модуль, змінити параметри в разі якщо допуски необхідно буде збільшити або зменшити, та переглянути логи, для який існує спеціальна модель за допомогою якої формуються логи. В меню зміни параметрів у користувача є можливість змінити еталон та встановити нові допуски. Зовнішній вигляд модуля до та після зміни параметрів, модуля логування та форми зміни параметрів приведено на рис. 3.13-3.16.

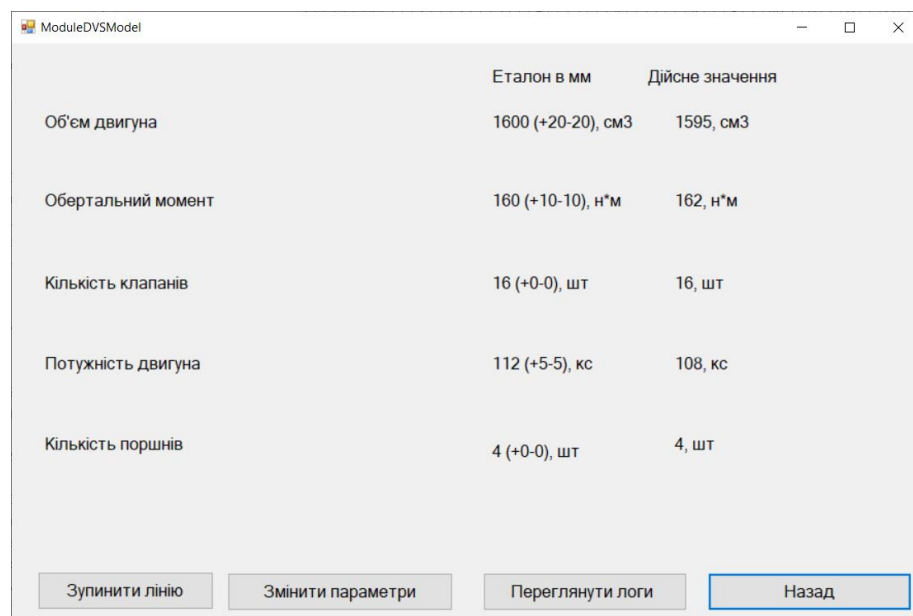


Рисунок 3.13 – Меню модуля двигуна

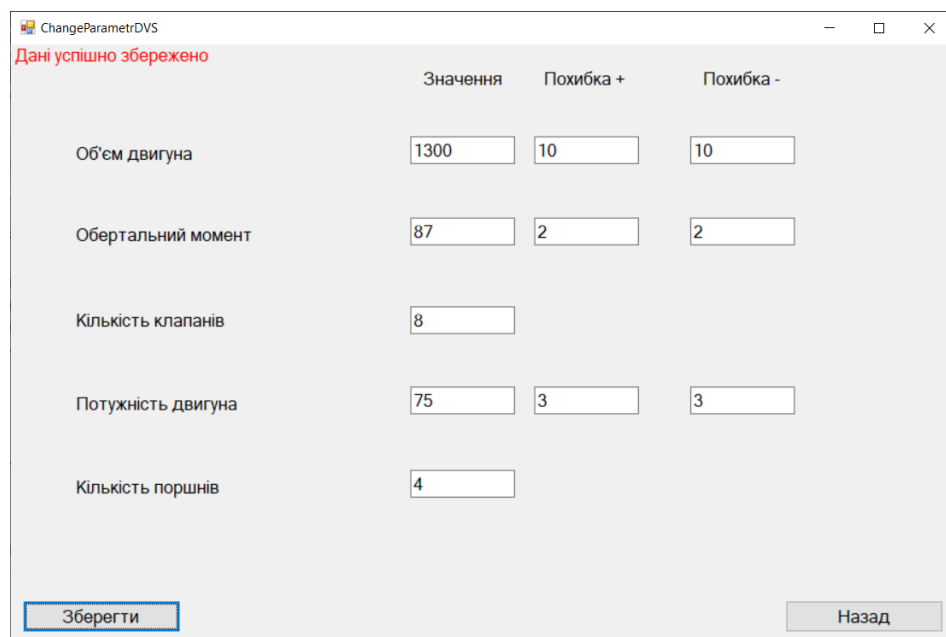


Рисунок 3.14 – Меню зміни параметрів двигуна

ChangeParametrDVS

Дані успішно збережено

	Значення	Похибка +	Похибка -
Об'єм двигуна	1300	10	10
Обертальний момент	87	2	2
Кількість клапанів	8		
Потужність двигуна	75	3	3
Кількість поршнів	4		

Зберегти Назад

Рисунок 3.15 – Меню модуля двигуна після зміни параметрів

LoggingDvsForm

	Об'єм двигуна	Обертальний момент	кількість клапанів	Потужність двигуна	Кількість поршнів
	1285, см3	93, н'м	8, шт	75, кс	4, шт
	1310, см3	91, н'м	8, шт	76, кс	4, шт
	1295, см3	89, н'м	8, шт	74, кс	4, шт
	1284, см3	94, н'м	8, шт	76, кс	4, шт
	1309, см3	93, н'м	8, шт	74, кс	4, шт
	1282, см3	94, н'м	8, шт	74, кс	4, шт
	1308, см3	92, н'м	8, шт	75, кс	4, шт
	1293, см3	90, н'м	8, шт	77, кс	4, шт
	1306, см3	91, н'м	8, шт	76, кс	4, шт
	1585, см3	168, н'м	16, шт	116, кс	4, шт
	1601, см3	162, н'м	16, шт	115, кс	4, шт
	1590, см3	152, н'м	16, шт	111, кс	4, шт
	1615, см3	169, н'м	16, шт	115, кс	4, шт
	1588, см3	169, н'м	16, шт	113, кс	4, шт
	1589, см3	155, н'м	16, шт	115, кс	4, шт
	1602, см3	155, н'м	16, шт	113, кс	4, шт
	1604, см3	162, н'м	16, шт	115, кс	4, шт
	1592, см3	151, н'м	16, шт	111, кс	4, шт
	1581, см3	161, н'м	16, шт	107, кс	4, шт
	1594, см3	162, н'м	16, шт	115, кс	4, шт
	1611, см3	155, н'м	16, шт	115, кс	4, шт

Назад

Рисунок 3.16 – Логування модуля двигуна

Концепція координації для модуля управління збіркою салона також як і концепція модуля збірки двигуна, за виключенням параметрів, які в даному випадку являються кількісною характеристикою панелі приборів(торпеди) в автомобілі, в залежності від оцінки характеристики деякі поля можуть не допускати допуски так як

параметр описується цілим числом такі як кількість сидінь, або кількість індикаторів. Меню управління модулем збірки салону та зміни параметрів приведені на рис. 3.17 - 3.19.

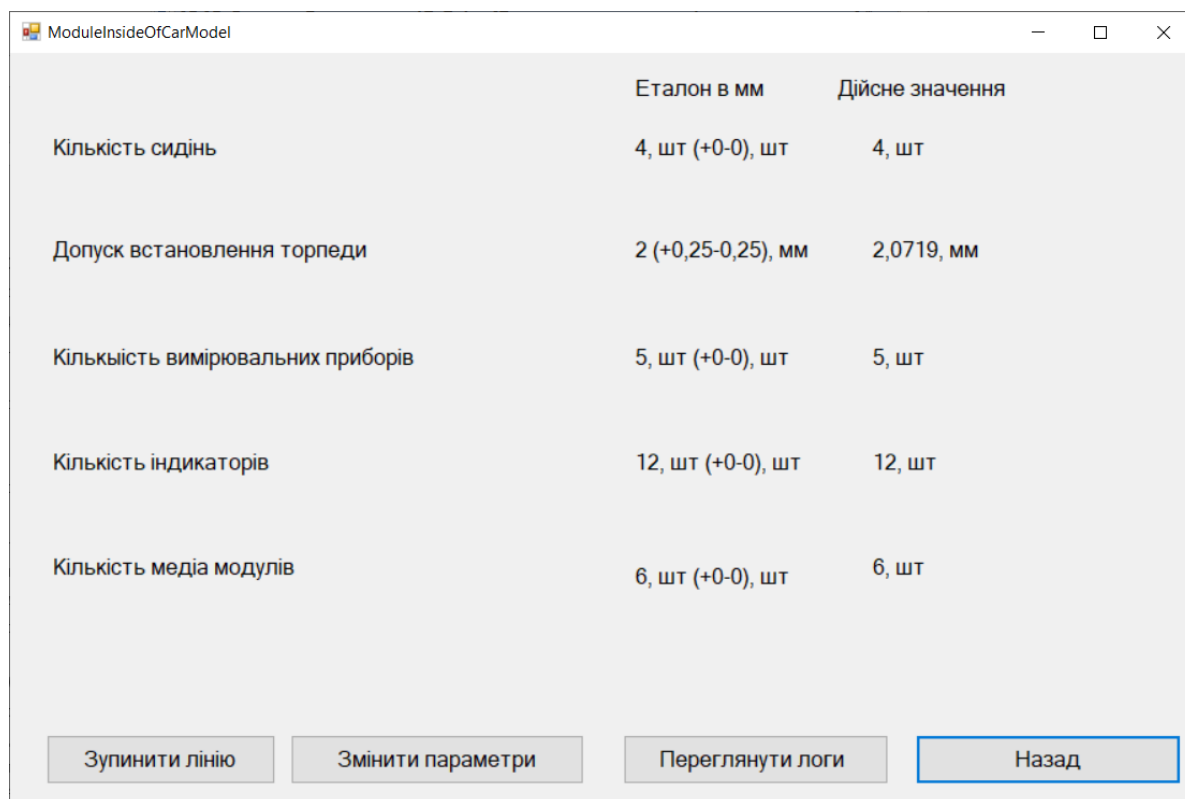


Рисунок 3.17 – Меню керування модулем збірки салона

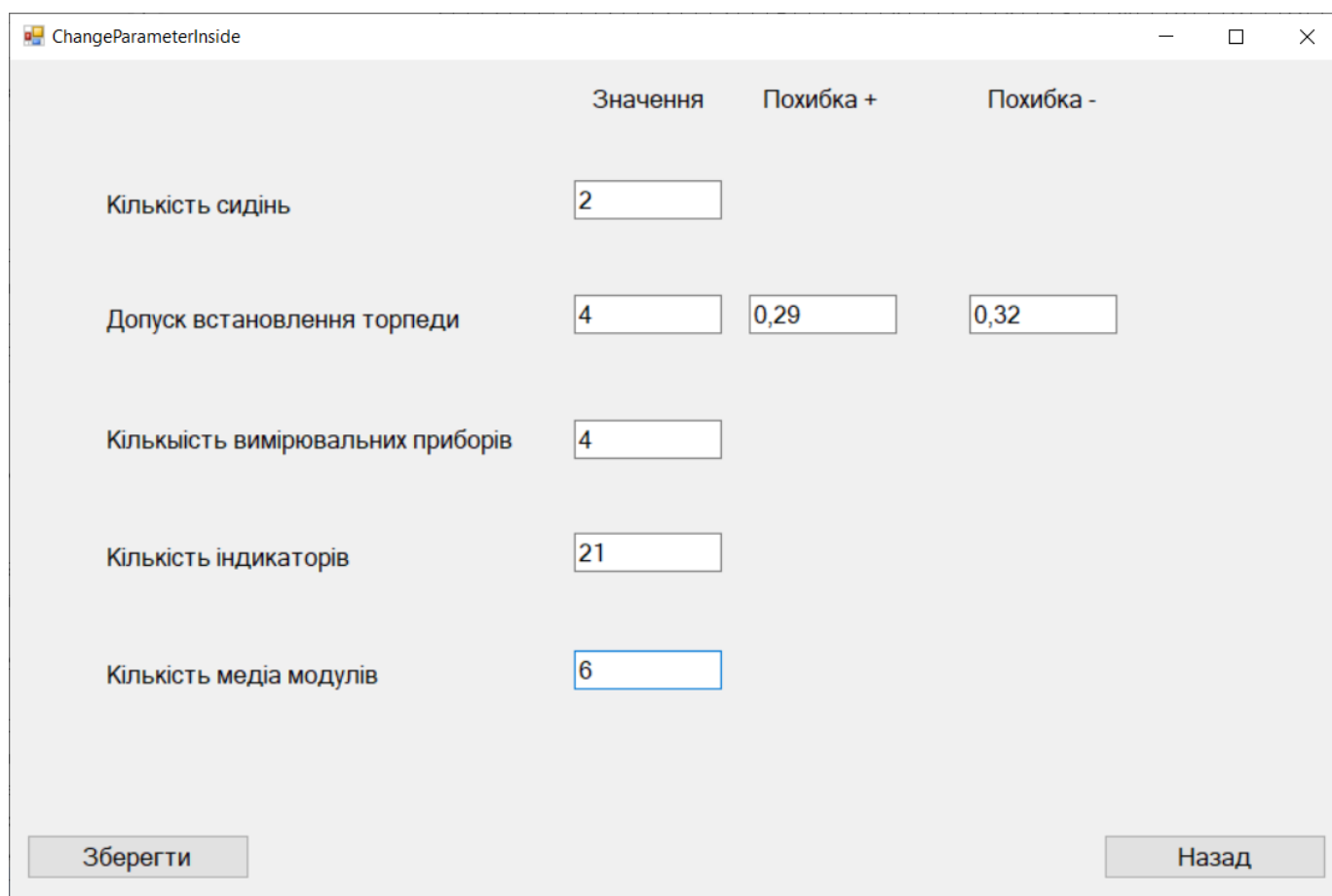


Рисунок 3.18 – Меню зміни параметрів модуля збірки самона

	Значення	Похибка +	Похибка -
Кількість сидінь	2		
Допуск встановлення торпеди	4	0,29	0,32
Кількість вимірювальних приборів	4		
Кількість індикаторів	21		
Кількість медіа модулів	6		

Buttons: Зберегти, Назад

Рисунок 3.19 – Меню після зміни параметрів

Наступний модуль збірки кузова, даний модуль показує те, що данні модулі можуть бути багатфункціональними і кожен модуль може використовуватися для моніторингу самих різних ситуацій, в даному модулю можна переглянути параметри допусків(зазорів) кузова та аналогічно попереднім встановити нові значення для параметрів, точність даного модуля 4 знаки після коми. Даний модуль приведено на рис. 3.20-3.21.

	Еталон в мм	Дійсне значення
Базовий допуск встановлення деталей	(+3-0,6), мм	+3-0,6, мм
Допуск капота	3 (+0,6-0,6), мм	3,1886, мм
Допуск дверей	4 (+0,6-0,6), мм	3,7719, мм
Допуск переднього бампера	3 (+0,6-0,6), мм	2,4674, мм
Допуск заднього бампера	2 (+0,6-0,6), мм	2,1667, мм
Допуск вітрового скла	4 (+0,6-0,6), мм	3,5162, мм
Допуск окрашення	2 (+0,6-0,6), мм	2,0454, мм
Допуск багажника	2 (+0,6-0,6), мм	2,1259, мм

Buttons: Зупинити лінію, Змінити параметри, Переглянути логи, Назад

Рисунок 3.20 – Меню модуля збірки кузова

Зміна параметрів модуля кузова

Дані успішно збережено

	Значення	Похибка +	Похибка -
Базовий допуск встановлення деталей		0,4	0,8
Допуск капота	2	0,2	0,6
Допуск дверей	3	0,26	0,3
Допуск переднього бампера	4	0,1	0,2
Допуск заднього бампера	4	0,3	0,3
Допуск вітрового скла	0,2	0,02	0,06
Допуск окрашення	85	4	4
Допуск багажника	3,8	0,3	0,26

Зберегти Назад

Рисунок 3.21 – Меню зміни параметрів модуля кузова

Модуль шасі розрахований на запланований(очікуваний) запас міцності тобто кожен з параметрів представляє собою показник який було отримано в результаті випробовувань певної деталі на стенді, після чого дані заходять в головне меню та логуються. Загальний вигляд меню приведено на рис. 3.22.

ModuleShassiModel

	Еталон в мм	Дійсне значення
Запас міцності ричагів	52 (+0,8-0,3), кН	52,5048, кН
Запас міцності стабілізатора	235 (+1,2-1), кН	234,7188, кН
Запас міцності шарової опори	336 (+1,3-1,3), кН	336,4847, кН
Запас міцності приводів	212 (+2,2-1,8), кН	210,7701, кН
Запас міцності редуктора	400 (+1,4-1,3), кН	398,7377, кН

Зупинити лінію Змінити параметри Переглянути логи Назад

Рисунок 3.22 – Меню модуля шасі

Меню зміни параметрів містить 5 параметрів так як при випробуваннях точність дуже важлива то кожна з характеристик має похибки та може мати точність до 4 знака після коми логування відповідно буде аналогічної точності, ілюстрація даних модулів приведено на рис. 3.23-3.24.

	Значення	Похибка +	Похибка -
Запас міцності ричагів	11	0,8	0,8
Запас міцності стабілізатора	232	0,8	0,8
Запас міцності шарової опори	3415	0,8	0,8
Запас міцності приводів	455	0,8	0,8
Запас міцності редуктора	562	0,8	0,8

Зберегти Назад

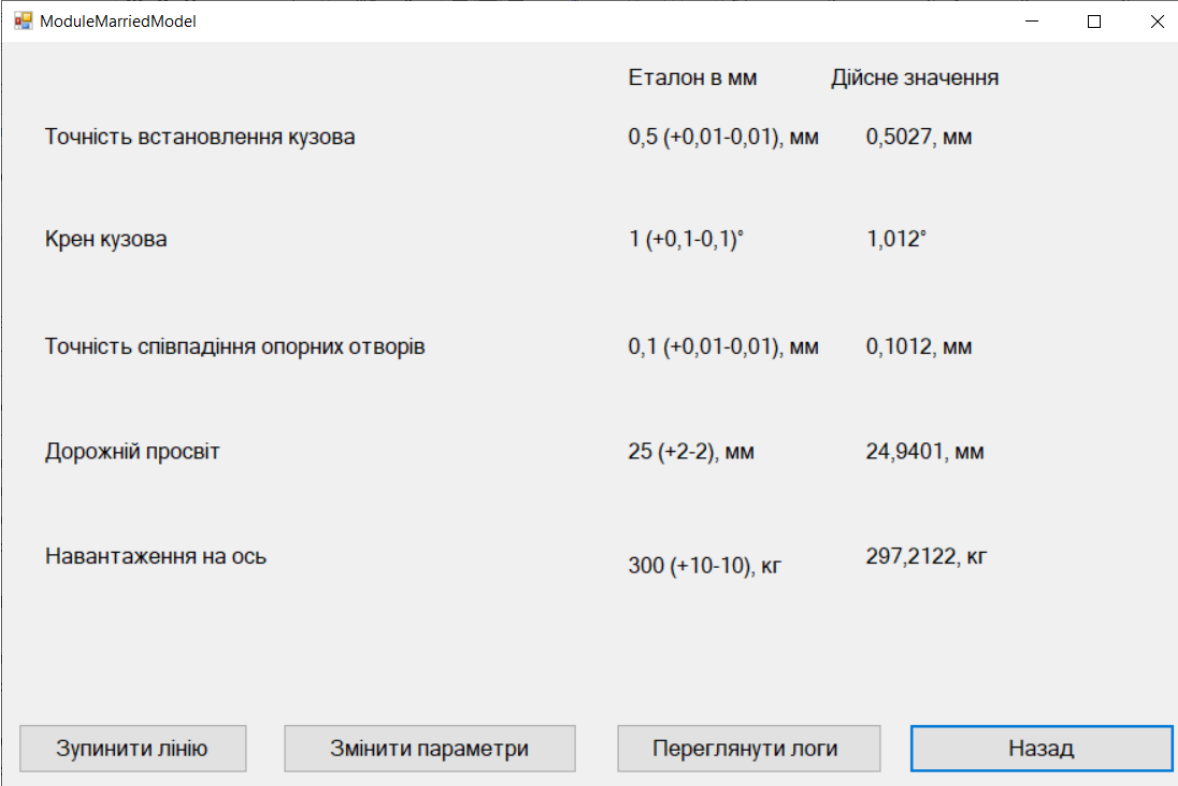
Рисунок 3.23 – Меню зміни параметрів шасі

	Запас міцності ричага	Запас міцності стабілізатора	Запас міцності шарової опори	Запас міцності приводів	Запас міцності редуктора
	12,6837, кН	23,2138, кН	34,0363, кН	44,5923, кН	56,519, кН
	11,4188, кН	22,4649, кН	34,292, кН	45,0738, кН	55,2314, кН
	11,4921, кН	22,3581, кН	33,7233, кН	44,2779, кН	56,7795, кН
	11,8272, кН	23,2092, кН	33,979, кН	44,7594, кН	55,4919, кН
	12,2425, кН	22,4747, кН	33,2563, кН	45,192, кН	55,4161, кН
	11,2899, кН	23,4793, кН	34,5818, кН	45,0603, кН	55,7222, кН
	12,1991, кН	22,2419, кН	33,5315, кН	44,6059, кН	56,3926, кН
	11,6146, кН	23,3212, кН	34,4375, кН	44,2197, кН	56,5401, кН
	12,0106, кН	22,7771, кН	33,7386, кН	45,7327, кН	55,9763, кН
	11,9497, кН	22,5723, кН	34,6932, кН	44,7012, кН	55,2525, кН
	12,3457, кН	23,6282, кН	33,9943, кН	44,6142, кН	56,2887, кН
	12,7157, кН	22,3819, кН	33,5608, кН	44,4102, кН	55,8598, кН
	12,2639, кН	23,0049, кН	34,149, кН	45,7385, кН	56,5759, кН
	12,3372, кН	22,898, кН	33,5803, кН	44,9426, кН	56,5239, кН
	12,0754, кН	23,5401, кН	34,3561, кН	45,2652, кН	56,1596, кН
	12,4105, кН	22,7912, кН	34,6117, кН	45,7468, кН	56,472, кН
	11,3846, кН	22,3027, кН	33,3058, кН	44,8109, кН	55,23, кН
	12,4838, кН	22,6843, кН	34,0431, кН	44,9509, кН	56,42, кН
	12,032, кН	23,3074, кН	34,6313, кН	44,6792, кН	55,5361, кН
	12,1053, кН	23,2005, кН	34,0627, кН	45,4833, кН	55,4842, кН
	11,8435, кН	22,2425, кН	33,2384, кН	44,206, кН	56,7198, кН

Назад

Рисунок 3.24 – Меню логування модуля шасі

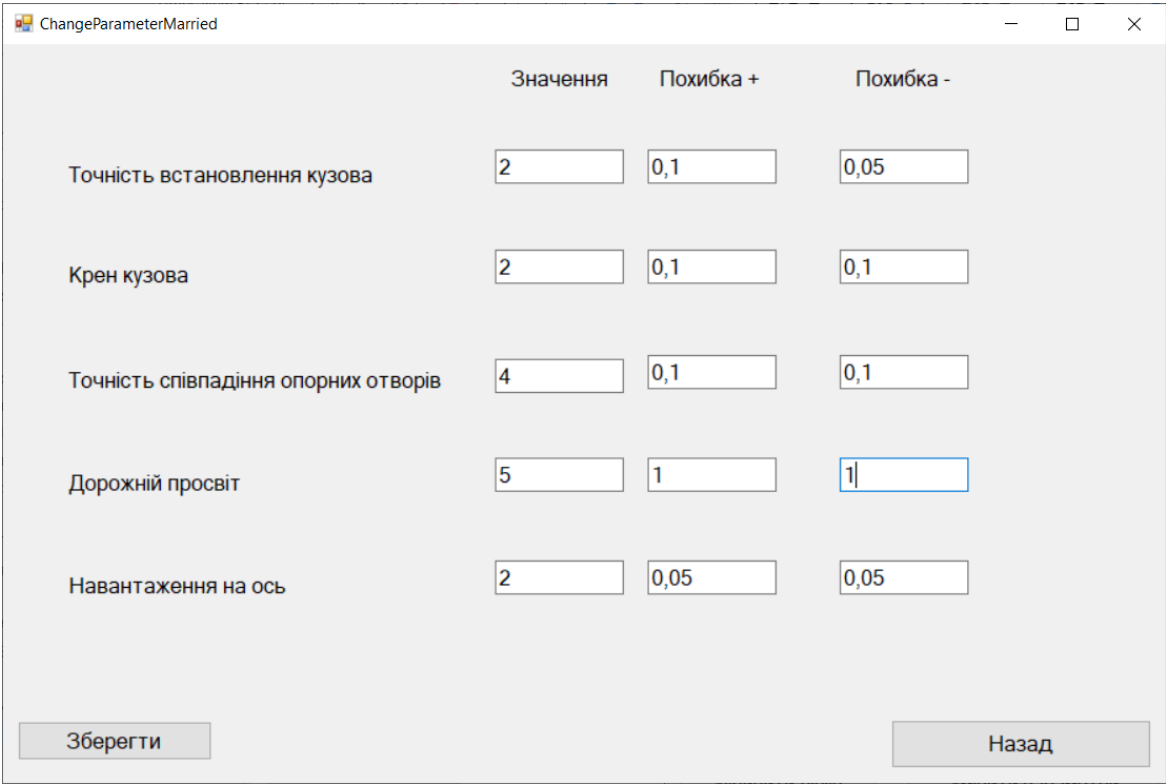
Модуль з'єднання кузова та шасі також має точність 4 знаки після коми, кожен параметр описує точність роботи підмодуля кожна з хаарктеристик може бути виведена в окремий під модуль та мати більш детальну інформацію при параметри. Меню модуля та зміни параметрами приведено на рис. 3.25 - 3.26



	Еталон в мм	Дійсне значення
Точність встановлення кузова	0,5 (+0,01-0,01), мм	0,5027, мм
Крен кузова	1 (+0,1-0,1)°	1,012°
Точність співпадіння опорних отворів	0,1 (+0,01-0,01), мм	0,1012, мм
Дорожній просвіт	25 (+2-2), мм	24,9401, мм
Навантаження на ось	300 (+10-10), кг	297,2122, кг

Buttons: Зупинити лінію, Змінити параметри, Переглянути логи, Назад

Рисунок 3.25 – Меню модуля з'єднання шасі та кузова



	Значення	Похибка +	Похибка -
Точність встановлення кузова	<input type="text" value="2"/>	<input type="text" value="0,1"/>	<input type="text" value="0,05"/>
Крен кузова	<input type="text" value="2"/>	<input type="text" value="0,1"/>	<input type="text" value="0,1"/>
Точність співпадіння опорних отворів	<input type="text" value="4"/>	<input type="text" value="0,1"/>	<input type="text" value="0,1"/>
Дорожній просвіт	<input type="text" value="5"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
Навантаження на ось	<input type="text" value="2"/>	<input type="text" value="0,05"/>	<input type="text" value="0,05"/>

Buttons: Зберегти, Назад

Рисунок 3.26 – Меню зміни параметрів модуля з'єднання шасі та кузова

Останній модуль це самий важливий модуль – модуль контролю якості автомобіля, так як параметр являється логічним значенням так або ні то в меню зміни параметр не має сенсу. Кожен параметр відповідає за окремий модуль який вище був згаданий, та формується висновок по результатам модуля, при не спів падінні хоча б одного параметра в будь якому модулі, продукт(автомобіль) знімається з лінії та відправляється на усунення недоліків. Вид модуля приведено на рис. 3.27.

	Еталон в мм	Дійсне значення
Чи допустимі допуски по кузову	Так	Так
Чи допустимі допуски по двигуну	Так	Так
Чи допустимі допуски по салону	Так	Так
Чи допустимі допуски по з'єднанню кузова та шасі	Так	Так
Чи допустимі допуски по шасі	Так	Так

Buttons at the bottom:

Рисунок 3.27 – Меню модуля контролю якості

3.6 Розрахунок технічних затрат на програмний засіб

Умовне число команд Q визначається за формулою:

$$G = f \cdot p \quad (3.1)$$

де f – коефіцієнт умовного числа команд який приймається як 5000;
 p – коефіцієнт новизни та складності завдання, який приймається 0,8.

$$G = 5000 \cdot 0,8 = 4000, \text{ команд.}$$

Для розрахунку часу затраченого на створення програмного засобу необхідно обчислити формулу (3.8).

$$R = R_{\text{по}} + R_o + R_a + R_{\text{бс}} + R_{\text{н}} + R_{\text{п}} + R_{\text{от}} + R_{\text{д}}, \quad (3.2)$$

де $R_{\text{по}}$ – час на підготовку та опис ТЗ, частіше за все цк 40 люд/год;

R_o – час на опис завдання, який визначається за формулою (3.3);

R_a – час затрачений при розробці алгоритму роботи ПЗ, який визначається за формулою (3.4);

$R_{\text{бс}}$ – час затрачений на розробку схеми, який визначається за формулою (3.5);

$R_{\text{н}}$ – час затрачений на написання програмного коду, який визначається за формулою (3.6);

$R_{\text{п}}$ – час затрачений на відладку програми, який визначається за формулою (3.38);

$R_{\text{от}}$ – час затрачений на тестування працездатності програми, який визначається за формулою (3.7);

$R_{\text{д}}$ – час на оформлення документації, та самодокументування коду, який приймається як 24 люд/год.

$$R_o = G \cdot L / (50 \cdot K), \quad (3.3)$$

де L – коефіцієнт вірогідної зміни завдання або коригування, який приймає значення 1,5;

K – коефіцієнт кваліфікації розробника, який приймає значення 0,8.

$$R_a = G / (50 \cdot K), \quad (3.4)$$

$$R_{\text{н}} = G \cdot 1,5 / (50 \cdot K), \quad (3.5)$$

$$R_{\text{п}} = G / 50, \quad (3.6)$$

$$R_{\text{от}} = G \cdot 4 / (50 \cdot K), \quad (3.7)$$

$$R_o = 4000 \cdot 1,5 / 50 \cdot 0,8 = 150 \text{ люд/год,}$$

$$R_a = 4000 / 50 \cdot 0,8 = 100 \text{ люд/год,}$$

$$R_H = 4000 \cdot 1,5 / 50 \cdot 0,8 = 150 \text{ люд/год},$$

$$R_{\Pi} = 4000 / 50 = 80 \text{ люд/год},$$

$$R_{\text{от}} = 4000 \cdot 4 / 50 \cdot 0,8 = 400 \text{ люд/год},$$

$$K = 40 + 150 + 100 + 100 + 150 + 80 + 400 + 24 = 1044 \text{ люд/год}.$$

Виходячи з вище приведених розрахунків, які показують що процес розробки даного програмного засобу є об'ємним та важким, а саме проце свідчить час затрачений на розробку програмного продукту, який складає приблизно **1044** люд/год.

3.7 Висновки до розділу 3

Розробили комп'ютерно-інтегровану систему для координації функціонування підсистем на виробництві, алгоритм роботи та схему бази даних, описали вибір середовища програмування, платформу та мову програмування.

3.8 Охорона праці

Сучасний розвиток техніко-технологічного стану інфраструктури підприємств забезпечує постійну автоматизацію та оптимізацію виробничих процесів при роботі з комп'ютером.

Будівля, де планується встановлення та подальша робота з комп'ютером, має відповідати проектній документації, погодженій з уповноваженими державними органами. Крім того, необхідно враховувати стандарти санітарного освітлення, вимоги до параметрів мікроклімату (температура, відносна вологість), ступінь та інтенсивність вібрації, шуму і вогнестійкість приміщення, а також характеристики електромагнітного, ультрафіолетового та інфрачервоного випромінювання. поля [71].

У кожному приміщенні, де будуть обладнані робочі місця працівників, які працюють за комп'ютером, мають бути елементи природного та штучного освітлення. У той же час на вікнах необхідно встановити легко регульовані жалюзі або штори, які дозволять персоналу регулювати рівень освітлення в кімнаті.

Комп'ютери бажано розташувати в кімнаті так, щоб світло потрапляло на екрани моніторів з півдня або північного сходу. Для досягнення максимального рівня безпеки та здоров'я при роботі з комп'ютером виробничі приміщення повинні бути оснащені аптечками, автоматичними системами пожежної сигналізації та вогнегасниками

Вимоги до особистого робочого місця працівника

Розмір одного робочого місця повинен бути не менше 6 кв м. При необхідності суміжні робочі місця, які працюють з комп'ютером, слід розділити перегородками висотою до 2 метрів. При визначенні відповідного розміру кімнати та робочого простору на одну людину слід також враховувати шафи, сейфи, шафи чи інші меблі чи техніку, які знаходяться в кімнаті.

На столі працівника можна розміщувати допоміжні пристрої (принтери, колонки, сканери), наприклад місця для зберігання документів, за умови, що це не обмежує видимість екрану та не заважає. Робочий стілець повинен бути поворотним, легко регулюватися по висоті та забезпечувати достатню підтримку та зручне положення спини та хребта. Щодня проводити вологе прибирання, очищати робоче місце та монітор комп'ютера від пилу [70].

Забороняється:

- ремонт і обслуговування комп'ютера на робочому місці;
- спонтанний ремонт або спроба виконати технічне налаштування комп'ютера без залучення компетентних спеціалістів;
- зберігати непотрібні документи, деталі та предмети, які не потрібні для роботи;
- використовувати монітори з розмитим зображенням і монітори розриву екрана;
- працювати на матричному принтері без антивібраційного покриття та зі знятою кришкою.

До роботи з комп'ютером не допускаються особи, які не пройшли затверджений підприємством курс охорони зайнятості.

Організація робочого місця повинна передбачати:

- правильне розміщення робочого місця у виробничому приміщенні;
- хороші умови освітлення приміщення та робочого місця, відсутність відблисків;
- хороші ергономічні характеристики найважливіших елементів робочої підлоги;
- характер і особливості трудової діяльності.

Щоб зменшити втому, користувачам комп'ютера повинно бути комфортно [70].

Конструкція робочого місця користувача ПК (сидіння під час роботи) повинна забезпечувати оптимальну робочу позу з наступними ергономічними характеристиками:

- ноги - на підлозі або на підставці для ніг;
- стегна - в горизонтальній площині;
- передпліччя – вертикальні;
- лікті - під кутом $70^{\circ} - 90^{\circ}$ до вертикальної площини;
- зап'ястя зігнуті під кутом не більше 20° щодо горизонталі
- літаки;
- нахил голови – $15^{\circ} - 20^{\circ}$ відносно вертикальної площини [71].



Рисунок 3.28 – Еталон робочого місця

Робоче місце користувача ПК забезпечено письмовим столом, стільцем і підставкою для ніг. Висота робочого столу повинна бути в межах від 0,65 м до 0,8 м, а ширина повинна забезпечувати можливість виконання операцій у межах дії моторного поля.

Клавіатуру слід розташувати на стільниці на відстані 200 мм від краю до робочого. Конструкція клавіатури повинна передбачати опорний пристрій (виготовлений з матеріалу з високим коефіцієнтом тертя, що перешкоджає її переміщенню), що дозволяє змінювати кут нахилу поверхні клавіатури в межах $5^{\circ} \dots 15^{\circ}$.

ВИСНОВКИ

У кваліфікаційній роботі біло розроблено комп'ютерно-інтегровану систему для координації функціонування підсистемами підприємства, яка дає змогу підвищення техніко-економічних показників виробництва, за рахунок зменшення: витрат на обслуговування та відновлення окремих модулів систем управління координації, передбачення поломок у роботі системи, вартості продукції та проценту браку, витрат на відновлення модулів.

Було вирішено наступні завдання:

- досліджено засоби та алгоритми координації модулів виробництва виробничої лінії;
- обрано та обґрунтовано варіанти координації підсистем на виробництві;
- розроблено програмний засіб для координації підсистем на виробництві.

Розроблена система має значення для збільшення техніко-економічних показників виробництва шляхом передбачення пошкодження обладнання за рахунок моніторингу параметрів модулів та своєчасного реагування системи на відключення, завдяки чому виробничі лінії буде працювати мінімальний час в резонансі.

Також, логування всього процесу, дозволить зробити простішим сертифікацію товару та перевірки партії товарів так як, можна переглянути логи по кожній одиниці окремо.

Упровадження розробленого програмного засобу можливе в комп'ютерно-інтегрованих автоматизованих системах на виробництві для складальних ліній, так і в навчальних лабораторіях у освітньому процесі на лабораторних практикумах та практичних роботах в межах спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології та можуть цікавим для фахівців, що пов'язані з промисловою автоматизацією.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. ДСТУ 3008: 2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – К.: ДП “УкрНДНЦ”, 2016. – 30 с.
2. ДСТУ ГОСТ 7.1:2006. Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання / Нац. стандарт України. – Вид. офіц. – [Чинний від 2007–07–01]. – Київ : Держспоживстандарт України, 2007. – 47 с.
3. Положення про кваліфікаційну роботу здобувача вищої освіти на другому (магістерському) рівні [Електронний ресурс] : Наказ ХНУРЕ від 06 травня 2021 р. No 143. – Режим доступу : https://nure.ua/wp-content/uploads/Main_Docs_NURE/143-vid-06.05.2021-pro-vvedennja-v-diju-rishennja-vchenoi-radi -universitetu.pdf.
4. Стандарт вищої освіти магістра за спеціальністю 151 «Автоматизація та комп’ютерно-інтегровані технології» галузі знань 15 «Автоматизація та приладобудування» затверджено і введено в дію Наказом Міністерства освіти і науки України від 10.08.2020 р. No 1022. Режим доступу : <https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/2020/08/10/151-avtomatizatsiya- ta-kit-magistr.pdf>
5. Бондаренко Ю.В Перспективы применения микроэлектромеханических систем для промышленного оборудования /Ю.В. Бондаренко // Радиоэлектроника и молодежь в XXI веке : материалы 21-го Междунар. молодеж. форума, 2017 г. – Харьков : ХНУРЭ, 2017. – Т. 2. – С. 15 – 16.
6. Бондаренко Ю.В Інтерференційні методи контролю якості функціональних оптичних поверхонь /Ю.В. Бондаренко // Радиоэлектроника та молодь в XXI столітті : матеріали 22 – го Міждунар. молодіж. форуму, 2018 м. – Харкова : ХНУРЕ, 2018. – Т. 2. – С. 25 – 26.
7. Бондаренко Ю.В Примеры использования МЭМС устройств /Ю.В. Бондаренко // Радиоэлектроника и молодежь в XXI веке: материалы 23-го Междунар. молодеж. форума, 2019 г. – Харьков : ХНУРЭ, 2019. – Т. 2. – С. 25 – 26.
8. Невлюдова В. В. Основные причины отказа гибких комутационных плат / В. В. Невлюдова, Н. П. Демская Ю.В. Бондаренко // XVI Международная научно-техническая конференция: материалы XVI международной научно-технической

конференція «Физические процессы и поля технических и биологических объектов», 2017 г. – Кременчуг, 2017. – С. 33-34.

9. Бондаренко Ю.В. Цифрова економіка в бізнесі / Ю.В. Бондаренко, С. В. Хрустальова// Міжнародна науково-практична конференція: матеріали, Міжнародної науково-практичної конференція «Математичне моделювання процесів в економіці та управлінні проектами і програмами» 2019 г. – Харків – Миколаїв, 2019. – С. 24 – 25.

10. Бондаренко Ю.В. Мехатроника и мехатронные устройства /Ю.В. Бондаренко, И. О. Яшков // Актуальні проблеми сучасної науки: матеріали II міжнародної науково-практичної конференції, 2019 р. – Київ, 2019. – С. 51 – 53.

11. Бондаренко Ю.В. Автоматизированная система управления (АСУ) /Ю.В. Бондаренко, И. О. Яшков // Актуальні проблеми сучасної науки: матеріали II міжнародної науково-практичної конференції, 2019 р. – Київ, 2019. – С. 53 – 55.

12. Бондаренко Ю.В. Порівняння продуктивності бункерно - завантажувальних пристроїв /Ю.В. Бондаренко, И. О. Яшков // Теоретичні та практичні аспекти розвитку науки: матеріали III міжнародної науково-практичної конференції, 2017 р. – Київ, 2017. – С. 9 – 11.

13. Бондаренко Ю.В. Особенности автоматизации банковских систем /Ю.В. Бондаренко, И. О. Яшков // Теорія і практика розвитку наукових знань: матеріали II міжнародної науково-практичної конференції, 2017 р. – Київ, 2017. – С. 22 – 24.

14. Бондаренко Ю. В. Аналіз методів контролю параметрів функціональних поверхонь підкладок / Ю. В. Бондаренко // ЗБІРНИК студентських наукових статей «Автоматизація та приладобудування» / Ю. В. Бондаренко. – Харків, 2017. – С. 96–101..

15. Бондаренко Ю. В. Микрооптические устройства на основе отражающих элементов – микрозеркал / Ю. В. Бондаренко // ЗБІРНИК студентських наукових статей «Автоматизація та приладобудування» / Ю. В. Бондаренко. – Харків, 2018. – С. 16–20.

16. Бондаренко Ю. В. Светофильтры / Ю. В. Бондаренко // ЗБІРНИК студентських наукових статей «Автоматизація та приладобудування» / Ю. В. Бондаренко. – Харків, 2018. – С. 208–211.

17. Филипенко О. І. Методи контролю структур топології поверхонь матеріалів виробів електронної техніки, МЕМС та МОЕМС / О. І. Филипенко, О. О. Чала, Ю. В. Бондаренко // Технологія приладобудування / О. І. Філіпенко, О. О. Чала, Ю. В. Бондаренко. – Харків, 2018. – С. 3–7.

18. Невлюдов, І.Ш. Основи наукових досліджень [Текст]: навчальний посібник/ І.Ш. Невлюдов, Олександров Ю.М, Андрусевич І.Ш., Чала О.О.– КК НАУ м. Кривий ріг, 2017. – 344 с
19. Невлюдов І.Ш. Технологии микросистемной техники [Текст]/ И. Ш. Невлюдов, В. А. Палагин, Е. А. Чалая/– НТЖ «Технология приборостроения», Харьков. – 2014.– № 3, С.7-10.
20. Невлюдов І.Ш., Палагин В.А., Чалая Е.А. «Технологии микросистемной техники (часть II)», НТЖ «Технология приборостроения». – Х., 2015. №2.
21. Невлюдов І. Ш.Трансфер технологій у сучасній науці, освіті та виробництві в умовах четвертої промислової революції «ІНДУСТРІЯ 4.0» / Невлюдов І. Ш., Чала О. О., Олександров Ю. М. // Сучасний рух науки: тези доп. VIII міжнародної науково-практичної інтернетконференції, 3-4 жовтня 2019 р. – Дніпро, 2019. – Т.2 С.: 604-608
22. O. Filipenko, O. Chala, V. Bortnikova, O. Sychova and I. Botsman, «Impact of Technological Operations Parameters on Moems Components Formation,» 2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers (CAOL), 2019, pp. 371-374, doi: 10.1109/CAOL46282.2019.9019570.
23. Филипенко О. І. Технологічні фактори виробництва, що впливають на якість покриттів дзеркальних поверхонь МОЕМС-перемикачів / О. І. Филипенко, О. О. Чала, М. І. Відешин // Наукові нотатки. - 2017. - Вип. 57. - С. 178-183. - Режим доступу: http://nbuv.gov.ua/UJRN/Nn_2017_57_29.
24. Методи та моделі кібер-фізичного керування процесами в організаційно-технічних виробничих об'єктах: дис. докт. техн. наук : 05.13.07 / . – Харків, 2021. – 406 с.
25. Невлюдов, І. Ш., Демська, Н. П., Чала, О. О., & Демська, А. І. Групове управління гнучкими виробничими системами у виготовленні мемс виробів. ББК: У 290-21, 101.
26. Nevliudov, I. S., Pysmenetskyi, V.O., Frolov, A. V., Chala, O. O., Yemelianov, M. A. (2017). Аналіз ефективності використання кремнієвих концентраторних сонячних фотоелементів. Система управління, навігації та зв'язку. Збірник наукових праць, 6(46), 85-88.
27. Чала О. О. Технологічне забезпечення якості підкладок функціональних компонентів мікрооптоелектромеханічних систем. – 2021.
28. V. Bortnikova, V. Yevsieiev, S. Maksymova, I. Nevliudov, O. Chala and K. Kolesnyk, «Mathematical Model of Equivalent Stress Value Dependence from Displacement

of RF MEMS Membrane,» 2019 IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), 2019, pp. 83-86, doi: 10.1109/MEMSTECH.2019.8817394.

29. Левченко Є. О. Machine-to-mobile (M2M) в автотранспортних мережах / Є. О. Левченко, О. О. Чала // Стан, досягнення та перспективи інформаційних систем і технологій : матеріали XXXI Всеукраїнск. наук.–техн. конф. молодих вчених, аспірантів та студентів, 22–23 квітня 2021 р. – Одеса, 2021. – С. 162–163.

30. Чала О. О. Вплив параметрів технологічних операцій шліфування та полірування на формоутворення компонентів MOEMS / О. О. Чала, О. І. Филипенко, І. В. Боцман // III-я Міжнародна конференція «Виробництво & Мехатронні Системи 2019», м. Харків, 24-25 жовтня 2019 р. – Х. : ХНУРЕ, 2019. – С. 69 – 72.

31. Невлюдов І. Ш. Основи виробництва електронних апаратів //Харків: Сміт. – 2005.

32. V. Palagin, I. Razumov-Fryziuk, I. Botsman and V. Nevliudova, «Development of multi-probe connecting devices on flexible polyimide base for MEMS components testing,» 2018 XIV-th International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), 2018, pp. 232-235, doi: 10.1109/MEMSTECH.2018.8365740.

33. Nevliudov I. S. et al. MEMS Multiprobe Contact Devices for Electrical Testing of Printed Circuit Boards and BGA Components //Engineering Science, Taylor & Francis. – 2017. – Т. 9. – №. 3 (2). – С. 462-468.

34. Картенко І. В. Механізми координації у складних ієрархічних системах / І. В. Картенко, І. В. Савка – Львів, 2008. – С. 156–166.

35. 1. Месарович М. Теория иерархических многоуровневых систем / М. Месарович, Д. Мако, И. Такахара. – М.: Мир, 1973. – 344 с.

36. Алиев Р.А. М.И. Методы и алгоритмы координации в промышленных системах управления / Алиев Р.А., Либерзон М.И. – М.: Радио и связь, 1987. – 208 с.

37. Алиев Р.А. Безытеративные алгоритмы координации в двухуровневых системах / Р.А. Алиев, М.И. Либерзон // Изв. АН СССР. Техническая кибернетика, 1986. – № 3. – С.163–166.

38. Алтунин А.Е. Исследование и разработка методов принятия решений в многоуровневых иерархических системах газовой промышленности: автореф. канд. техн.. наук. / А.Е. Алтунин; МИНХ и ГП им. И.М. Губкин. – М., 1979. – 24 с.

39. Погостинский Ю.А. Координационный механизм стратегического управления. – СПб., 2006. – 248 с.

40. Ahmed S. Coordination in International Manufacturing: The Role of Competitive Priorities and the Focus of Globally Dispersed Facilities / S. Ahmed, F. Andreas, O. Miguel. // MDPI. – 2018. – №10. – С. Sustainability.
41. Автомобілебудування України [Електронний ресурс]. – 2007. – Режим доступу до ресурсу: <http://ukrexport.gov.ua/ukr/prom/ukr/29.html>.
42. Месарович М. Общая теория систем / М. Месарович, Я. Такахара. – М: Мир, 1978. – 312 с.
43. MES Оперативне управління виробництвом [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: 41. <https://rarus.ua/ru/catalog/industry-solutions/pdm-itil-mes/mes-operativne-upravlinnya-virobnitstvom/>.
44. Desktop Guide (Windows Forms .NET) [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/overview/?view=netdesktop-6.0>.
45. Nachane D.M. Optimization methods in multilevel systems: a methodological survey / Nachane D.M. // «Eur. J. Oper. Res.». – 1985 – N1. – P. 25– 38.
46. Findeisen W. Two-level control and coordination for dinamisal systems / Findeisen W., Malinowski K. // Archiwum automatiki i telemechaniki. – Т. XXIV, N . – P. 3– 27.
47. Wilson I.D. Foundations of hierarhical control / Wilson I.D. // «International Journal of Control» – 1979. N 6. – P. 899–933.
48. Lee K.Y. Coordinated control of distributed-parameter systems / Lee K.Y. // «Distrib. Parameter Contr. Syst.», Oxford. – 1982. – P. 213–238.
49. Michalska H. Joint coordination method for the steady-state control of large-scale systems / Michalska H., Ellis J.E., Roberts P.D. // Int. J. Syst. Sci. – 1985. – N 5. – P. 605– 618.
50. Бернштейн Н.А. Архив биологических наук: Проблема взаимоотношений координации и локализации. Т. 38, вып. 1. – 1935.
51. Васильев Ю.В. Теория управления / Ю.В. Васильев, В.Н. Парахина, Л.И. Ушвицкий (ред.) – М.: Финансы и статистика, 2008. – 608 с.
52. Ансофф И. Новая корпоративная стратегия / И. Ансофф ; пер. с англ. под ред. Ю. Н. Контуревского // Спб. : Питер, 1999. – 416 с.
53. Коломієць О. В. Побудова динамічної моделі об'єднання підприємств із синергічними зв'язками / О. В. Коломієць // Вісник Державного Університету «Львівська політехніка». Логістика. – 2002. – № 446. – С. 328–338.

54. Ладанюк А. П. Координація функціонування технологічних дільниць цукрового заводу з урахуванням задач прогнозування / А. П. Ладанюк, Н. А. Заєць, Л. О. Власенко, М. Н. Луцька // Вісник Вінницького політехнічного інституту. – 2006. – № 6. – С. 112–115.

55. Математична модель [Електронний ресурс]. – 2021. – Режим доступу до ресурсу:

https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%BD%D0%B0_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C.

56. Безручко Б. П. Математическое моделирование и хаотические временные ряды / Б. П. Безручко, Д. А. Смирнов. – Саратов, 2005. – 343 с. – (ГосУНЦ "Колледж").

57. Методичні вказівки до практичних занять з дисципліни „Теорія систем та системний аналіз” (для студентів 3 курсу ФПО та ЗН галузі знань 0306 "Менеджмент і адміністрування" напряму підготовки 6.030601 "Менеджмент") / Харк. нац. акад. міськ. госп-ва; уклад.: В. М. Охріменко, Т. Б. Воронкова. – Х.: ХНАМГ, 2012. – 24 с.

58. Хасанов М. М. О методах анализа и управления самоорганизующимися процессами нефтегазодобычи / М. М. Хасанов // Нефтяное хозяйство. – 1994. – № 2. – С. 74–77.

59. Балакиров Ю. А. Оптимальное управление процессами нефтедобычи / Ю. А. Балакиров, Л. В. Капушак, Е. А. Слепян. – К. : Техника, 1987. – 148 с.

60. Yazenin A. V. Possibilistic optimization. Amasure-based approach / A. V. Yazenin, M. Wagenknecht. – Brandenburg : Technische Universitet Cottbus, 1996. – 133 p.

61. Заєць Н. А. Автоматизоване управління колонною дифузійною установкою з використанням принципів координації та адаптації / Н. А. Заєць // Восточноевропейский журнал передовых технологий. – 2008. – № 2(32). – С. 34–37.

62. Ладанюк А. П. Системний аналіз складного об'єкта в задачах діагностики та координації / А. П. Ладанюк, Л. О. Власенко, Н. А. Заєць // Автоматизація виробничих процесів. – 2006. – № 2. – С. 44–47.

63. Ладанюк О. А. Автоматизоване управління взаємозв'язаними підсистемами технологічних комплексів харчових виробництв : автореф. дис. на здобуття наук. ступеня канд. техн. наук : спец. 05.13.07 «Автоматизація технологічних процесів і виробництв» / О. А. Ладанюк ; Укр. держ. універ. харчових технологій. – К., 1996. – 16 с.

64. Плоткін Я.Д. Виробничий менеджмент: Навч. посіб. / Плоткін Я.Д., Пащенко І.Н. – Львів: «Львівська політехніка», 1999. – 258 с.
65. Introduction to Xamarin.Forms [Електронний ресурс] // Pluralsign. – 2015. – Режим доступу до ресурсу: https://www.pluralsight.com/courses/introduction-xamarin-forms?aid=7010a000002LUv2AAG&promo=&utm_source=non_branded&utm_medium=digital_paid_search_google&utm_campaign=XYZ_EMEA_Dynamic&utm_content=&cq_cmp=1576650371&gclid=CjwKCAiAwKyNBhBfEiwA_mrUMobUrZkY772OZ8XwBjfyt9pLzV0SNWOGzBJ0jYcbH-cy6AqRVNPsKxoCkUAQAvD_BwE.
66. Windows Forms overview (Общие сведения о Windows Forms) [Електронний ресурс] // Microsoft. – 2021. – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/windows-forms-overview?view=netframeworkdesktop-4.8>.
67. Unity (ігрової движок) [Електронний ресурс]. – 2021. – Режим доступу до ресурсу:
[https://ru.wikipedia.org/wiki/Unity_\(%D0%B8%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA\)](https://ru.wikipedia.org/wiki/Unity_(%D0%B8%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA)).
68. What is Windows Presentation Foundation (WPF) [Електронний ресурс] // Microsoft. – 2021. – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/visualstudio/designers/getting-started-with-wpf?view=vs-2022>.
69. What is .NET MAUI? [Електронний ресурс] // Microsoft. – 2021. – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/maui/what-is-maui>.
70. Охорона праці при роботі з комп'ютером [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://nkpk.org.ua/article/420>.
71. Інструкція з охорони праці при роботі з комп'ютером, принтером, ксероксом та іншою оргтехнікою [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://osvita-docs.com/node/41>.