

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)
бакалавр
(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ
ІНЖЕНЕРНИХ РОЗРАХУНКІВ РОТОРА
SOFTWARE DEVELOPMENT FOR ENGINEERING
CALCULATIONS OF A ROTOR

Виконав: студент 4 курсу, групи КНТ-217
Спеціальності 122 Комп'ютерні науки
забезпечення
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерні науки

Сітвелієв Е.Е.
(прізвище та ініціали)

Керівник Пархоменко А.В.
(прізвище та ініціали)

Рецензент Морщавка С.В.
(прізвище та ініціали)

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

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 122 Комп'ютерні науки
 (код і найменування)

Освітня програма (спеціалізація) Комп'ютерні науки
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2021 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Сітвелієва Ельвіса Ельдаровича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка програмного забезпечення для інженерних розрахунків ротора. Software Development for Engineering Calculations of a Rotor.
 керівник проєкту (роботи) Пархоменко Анжеліка Володимирівна, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “30” березня 2021 року № 103

2. Строк подання студентом проєкту (роботи) 17 травня 2021 року

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

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Дослідження та практичне використання систем автоматизації інженерних розрахунків. 3. Розробка програмного комплексу автоматизації інженерних розрахунків. 4. Керівництво програміста. 5. Керівництво оператора.

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

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Пархоменко А.В., доцент		
Нормоконтроль	Дейнега Л.Ю., ст. викладач		

7. Дата видачі завдання “ 28 ” лютого 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання
2	Аналіз предметної області	2 тиждень	Розділ 1
3	Вибір інструментарію розробки	3 тиждень	Розділ 2
4	Розробка структури програми	4 тиждень	Розділ 3
5	Розробка програмного забезпечення	5 тиждень	Розділ 3
6	Тестування та експериментальне дослідження програми	6 тиждень	Розділ 3
7	Оформлення пояснювальної записки та документів до неї	7 тиждень	Розділи 4,5, Додатки
8	Нормоконтроль та рецензування	8 тиждень	
9	Захист роботи	9 тиждень	

Студент

(підпис) Сітвелієв Е.Е.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Пархоменко А.В.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 86 с., 26 рис., 9 табл., 2 дод., 26 джерел.

СИСТЕМА АВТОМАТИЗОВАНИХ ІНЖЕНЕРНИХ РОЗРАХУНКІВ, МЕТОД СКІНЧЕНИХ ЕЛЕМЕНТІВ, ТЕХНОЛОГІЧНЕ ОБЛАДНАННЯ, ДИНАМІЧНЕ БАЛАНСУВАННЯ РОТОРА, ПРОГРАМНИЙ КОМПЛЕКС, АЛГОРИТМ РОЗРАХУНКУ МІЦНОСТІ.

Об'єкт дослідження – процес автоматизації інженерних розрахунків на етапі технологічної підготовки виробництва.

Предмет дослідження – методи та засоби розробки програмного забезпечення для автоматизації інженерних розрахунків.

Мета роботи – розробка програмного забезпечення для автоматизації інженерних розрахунків параметрів технологічного обладнання для підвищення точності обробки корпусних деталей.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мова програмування C++, середовище Qt Creator 2.8.1, ПК сумісний з IBM PC, процесор Intel Pentium 1,6 ГГц або вище, оперативна пам'ять з мінімальним об'ємом 1Гб, вільне місце на диску 10 Гб.

Результати. Створено програмний комплекс, який автоматизує інженерні розрахунки параметрів технологічного обладнання для механічної обробки корпусних деталей на верстаті з ЧПУ.

Висновки. Розроблено програмне забезпечення, що дозволяє розрахувати параметри коригуючих мас для динамічного балансування ротору верстата та механічні параметри пристосувань для кріплення заготовки корпусної деталі.

Галузь використання – автоматизоване виробництво корпусних деталей для автомобілебудування, авіабудування та ін.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 86 pages, 26 figures, 9 tables, 2 appendixes, 26 sources.

COMPUTER AIDED ENGINEERING SYSTEM, FINITE ELEMENTS METHOD, TECHNOLOGICAL EQUIPMENT, DYNAMIC BALANCING OF A ROTOR, SOFTWARE COMPLEX, STRENGTH CALCULATION ALGORITHM.

Object of study is a process of automation of engineering calculations at the stage of technological preparation of production.

Subject of study are methods and tools of software development for automation of engineering calculations.

The goal of the work is the development of software for automation of engineering calculations of parameters of technological equipment to increase the accuracy of machining of body parts.

Materials, methods, and tools: object-oriented programming, C++ programming language, IDE Qt Creator 5.1.1, PC compatible with IBM PC 1,6 GHz processor or higher, RAM with minimal 1GB, free disk space 10 GB.

Results. The software package has been created that automates engineering calculations of the parameters of technological equipment for machining body parts on a CNC machine.

Conclusions. A software system has been developed that allows calculating the parameters of corrective masses for dynamic balancing of the rotor of the machine and the mechanical parameters of devices for fastening the workpiece of the body part.

The area of application is automated production of body parts for the automotive industry, aircraft construction, etc..

ЗМІСТ

	С.
ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК	8
ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Призначення та особливості виготовлення корпусних деталей	10
1.2 Методи забезпечення точності обробки поверхонь	13
1.3 Методика балансування ротору	15
1.4 Методика розрахунку характеристик міцності пристосувань	18
1.5 Постановка задач дипломної роботи	20
2 ДОСЛІДЖЕННЯ ТА ПРАКТИЧНЕ ВИКОРИСТАННЯ СИСТЕМ АВТОМАТИЗАЦІЇ ІНЖЕНЕРНИХ РОЗРАХУНКІВ	22
2.1 Огляд існуючих систем автоматизації інженерних розрахунків	22
2.2 Структурний аналіз деталі кріплення в Pro/Mechanica	27
2.3 Висновки до розділу 2	36
3 РОЗРОБКА ПРОГРАМНОГО КОМПЛЕКСУ АВТОМАТИЗАЦІЇ ІНЖЕНЕРНИХ РОЗРАХУНКІВ	37
3.1 Призначення та функціональні можливості ПКАІР	37
3.2 Структура ПКАІР	37
3.3 Вибір інструментарію розробки	39
3.4 Розробка алгоритмів функціонування ПКАІР	43
3.5 Особливості програмної реалізації ПКАІР	48
3.6 Тестування ПКАІР	53
4 КЕРІВНИЦТВО ПРОГРАМІСТА	55
4.1 Призначення та умови виконання програми	55
4.2 Характеристики програми	55
4.3 Звернення до програми	57
4.4 Вхідні та вихідні дані	57
4.6 Повідомлення	58
5 КЕРІВНИЦТВО КОРИСТУВАЧА	59

5.1 Призначення програми.....	59
5.2 Умови виконання програми.....	59
5.3 Виконання програми	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	64
Додаток А Текст програми.....	67
Додаток Б Слайди презентації	79

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CAD	– Computer Aided Design;
CAE	– Computer Aided Engineering;
CAM	– Computer Aided Manufacturing;
FEA	– Finite Element Analysis;
IDE	– Integrated Development Environment;
MCE	– метод скінчених елементів;
ОС	– операційна система;
ПЗ	– програмне забезпечення;
ПКАІР	– програмний комплекс автоматизації інженерних розрахунків;
ЧПУ	– числове програмне управління.

ВСТУП

Корпусні деталі є важливими базовими елементами багатьох виробів. В корпусах зазвичай розташовуються різноманітні механізми. Наприклад, до корпусних деталей відносяться блоки циліндрів двигунів та компресорів, корпуса редукторів, насосів та ін. Від точності розмірів корпусних деталей залежить загальна точність виробу [1].

Тому, задача забезпечення якісної механічної обробки корпусних деталей вимагає застосування систем автоматизованих інженерних розрахунків (Computer Aided Engineering, CAE) в комплексі з системами автоматизованої технологічної підготовки виробництва (Computer Aided Manufacturing, CAM).

При цьому важливою проблемою залишається забезпечення правильного положення різального інструмента щодо оброблюваної поверхні, що ґрунтується на розрахунку параметрів та характеристик деталей пристосувань для кріплення заготовки оброблюваної деталі, а також на розрахунку параметрів противаг для динамічного врівноваження ротору – пристрою, що здійснює механічну обробку, зокрема розточування отвору в корпусній деталі.

Динамічне балансування деталей, що обертаються, потрібно для різного типу обладнання. При дисбалансі вал не тільки передає зайве зусилля, але й дестабілізує роботу системи. Динамічне балансування дозволяє усунути вібрацію, тим самим вісь обертання деталі приводиться до відповідності нормам, завдяки чому усуваються биття й вібрація та підвищується точність обробки. Але розрахунок системи противаг для динамічного врівноваження ротора є складною задачею. Тому, тема роботи є актуальною.

Мета роботи – розробка програмного забезпечення для автоматизації інженерних розрахунків параметрів технологічного обладнання для підвищення точності обробки корпусних деталей.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Призначення та особливості виготовлення корпусних деталей

В даний час виготовлення та обробка корпусних деталей – це одна з важливих галузей промисловості. Як відомо, саме корпусні деталі присутні в різних механізмах, машинах, приладах, технічних пристроях. До деталей такого типу відносять корпуси редукторів, насосів, коробки швидкостей і подач верстатів, блоки циліндрів двигунів і компресорів і інші вироби. Якщо врахувати, що в сучасному світі потрібні зовсім різні за своїми розмірами, параметрами, призначенням і конфігурацією деталі, то стає цілком зрозумілою затребуваність у виробництві такого роду [2].

Корпусні деталі відрізняються великим різноманіттям форм і розмірів оброблюваних поверхонь, а також точністю їх обробки. Залежно від цього використовують різні конструкції розточувальних кондукторів для закріплення корпусних заготовок і забезпечення правильного положення інструмента щодо оброблюваної поверхні [1].

Корпусні деталі можуть бути розділені на наступні види:

- деталі складної форми (бабки, кронштейни та ін.);
- деталі коробчатої форми із значною кількістю оброблюваних поверхонь (площині, отвори, пази);
- призматичні деталі з великими приварювальними поверхнями (один з розмірів деталі значно менший двох інших) [2].

Корпусні деталі зазвичай мають отвори, які можна розділити на дві групи:

- точні (основні), поверхні яких слугують опорами для валів, шпінделів та ін.;
- додаткові, наприклад для кріплення деталей (рис.1.1) [1], [3].

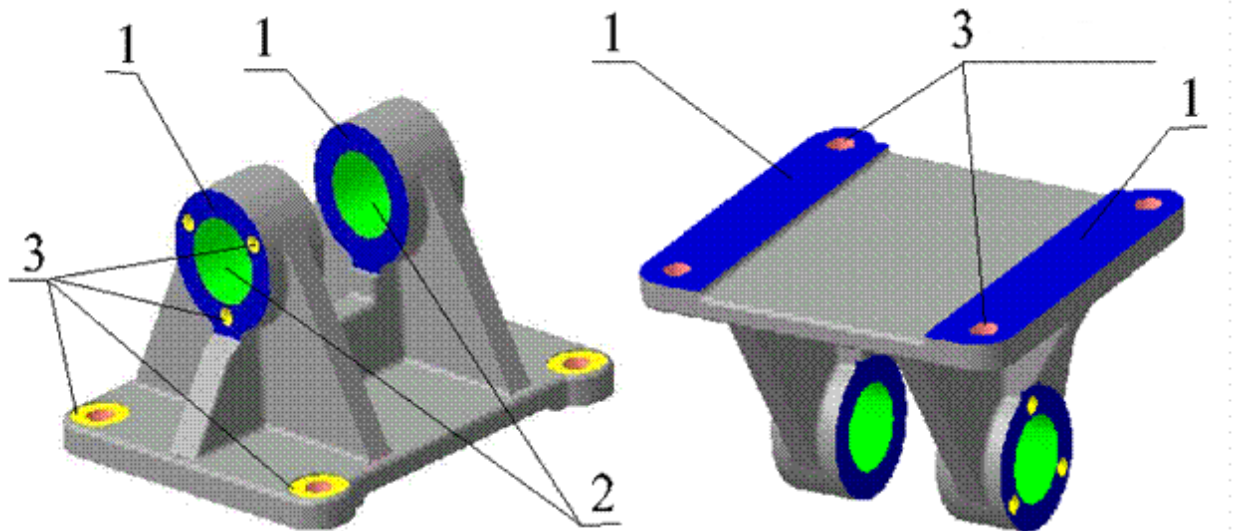


Рисунок 1.1 – Поверхні спряження (1), основні (2) та додаткові (3) отвори корпусної деталі [3]

Діаметри основних отворів в корпусних деталях обмежені, як правило розмірами 20-150 мм. Найбільша кількість отворів припадає на допоміжні, в тому числі кріпильні отвори діаметром до 20 мм [2].

В сучасному виробництві застосовують такі види обробки корпусних деталей:

- метод механічної обробки;
- фрезерування;
- витяжка;
- штампування;
- свердління;
- пробивання;
- зварювання.

Оскільки вироби такого типу найчастіше мають складну форму, несиметричну геометрію, площини обробки з різними шорсткостями, то обробка корпусних деталей перетворюється в трудомісткий процес, коли дуже важливо дотримуватися всіх вимог технічного завдання, безпеки, довговічності, працездатності [2].

В умовах одиничного та дрібносерійного виробництва механічна обробка корпусних деталей починається з розмітки, яку виконують в такій послідовності:

- наносять риски центрових осей;
- від цих осей розмічають інші осі отворів і контури деталі;
- розмічають окружності отворів [1].

У середньо- і крупносерійному виробництві обробка корпусних деталей здійснюється за допомогою спеціальних пристосувань, що повністю виключає їх розмітку. Основні отвори в корпусних деталях зазвичай обробляють на розточувальних, карусельно-токарних, радіально й вертикально-свердлильних і агрегатних верстатах, а іноді і на токарних верстатах з числовим програмним управлінням (ЧПУ) (рис.1.2) [1] - [2].

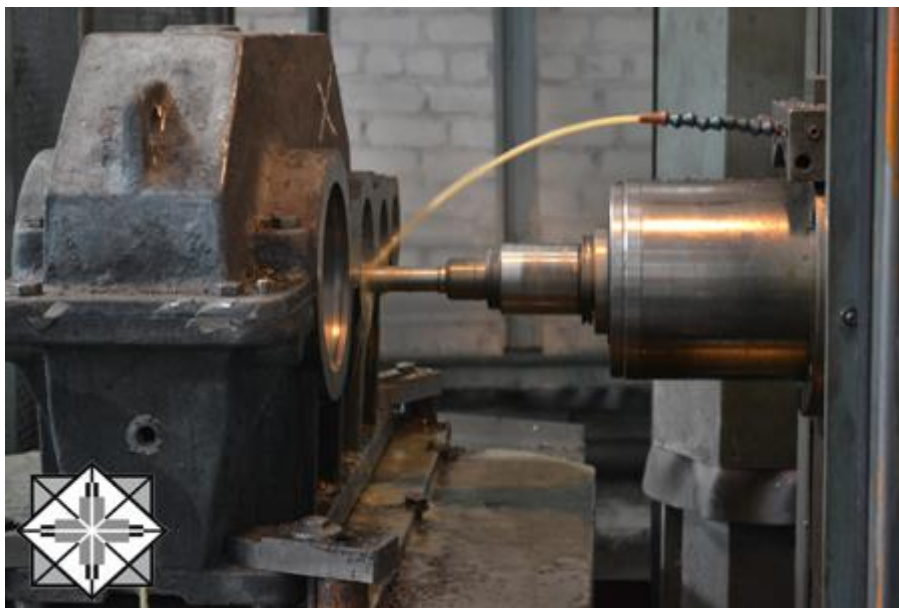


Рисунок 1.2 – Обробка корпусної деталі на верстаті з ЧПУ [2]

В одиничному та дрібносерійному виробництві при обробці отворів корпусні деталі встановлюють на оброблену основну поверхню за розміченими колами отворів. У дрібносерійному виробництві отвори розточують за допомогою накладних шаблонів, що закріплюються на деталі

або на основі пристосування. В цьому випадку шпиндель верстата встановлюється співвісно отвору шаблону [1].

У крупносерійному і масовому виробництві отвори розточують за допомогою спеціальних пристосувань, в яких інструмент має односторонній передній або задній напрямок, або передній і задній одночасно. З переднім або заднім напрямком обробляються зазвичай короткі отвори. Довгі отвори розточуються борштангами, що мають передній і задній напрямки [1].

При контролі корпусних деталей контролюють точність розмірів і відносного положення плоских поверхонь і головних отворів, точність геометричної форми та шорсткість базуючих поверхонь деталі, правильність відносного положення різьбових та інших дрібних отворів [4].

1.2 Методи забезпечення точності обробки поверхонь

При виробництві корпусних деталей пред'являються підвищені вимоги до обробки та шорсткості різних поверхонь і отворів, які прийняті за бази. Точність обробки базових поверхонь впливає на працездатність і надійність, а також на точність складання деталей і вузлів, що в кінцевому підсумку веде до злагодженої роботи всього агрегату [2].

Виходячи з призначення деталей, умов, в яких будуть експлуатуватися механізми, а також вимог до їх роботи й функціональності, визначають конструктивне виконання корпусних деталей, матеріал, з якого вони повинні бути виготовлені, а також параметри точності. Обов'язково повинні враховуватися й технологічні чинники отримання заготовок, можливість обробки різанням і зручність складання [2].

Аналіз процесу досягнення заданої точності при обробці заготовок корпусних деталей на верстатах з ЧПУ показує, що всі параметри можна розділити на дві групи:

- параметри, які не пов'язані з точністю відліку координатних переміщень робочих органів верстата (точність діаметральних розмірів і геометричної форми отворів та ін.);

- параметри, які пов'язані з точністю відліку і координатних переміщень робочих органів верстата (точність відстаней між поверхнями, точність лінійних розмірів і ін.) [1].

У більшості корпусних деталей перевіряють:

- прямолінійність і правильність розташування основних (базових) площин;
- розміри і форми основних отворів;
- співвісність осей отворів;
- міжосьові відстані, паралельність і перекіс осей;
- правильність розташування осей отвору щодо основних поверхонь;
- неперпендикулярність осей основних отворів;
- неперпендикулярність торцевої площині щодо осі отвору;
- шорсткість обробки поверхонь основних отворів, основних та інших площин [1].

Для отримання потрібних розмірів деталей потрібно не лише забезпечити точні переміщення робочих органів верстата, але винятково точно узгодити положення заготовки та інструмента в системі координат верстата. Відхилення від заданого положення призводить до зміщення всього оброблюваного контуру відносно технологічних баз або раніше оброблених поверхонь. Відхилення від потрібного положення заготовки відбувається через похибки базування, похибки пристосувань та ін. Похибка закріплення виникає при зміщенні заготовки в пристосуванні або зміщенні пристосування з заготовкою в робочій позиції під впливом сил різання або затиску. Зменшення цієї похибки на верстатах з ЧПУ відбувається раціональним підбором конструкцій та розмірів опорно-установочних елементів пристосувань, а також іншими шляхами [1].

У промисловості зазвичай застосовують три способи зменшення похибок обробки, що виникають в результаті пластичного деформування заготовки під дією сили різання:

- підвищення жорсткості закріплення заготовки за допомогою спеціальних опор;
- зменшення сили різання шляхом управління режимом різання;
- введення додаткових робочих ходів [5].

1.3 Методика балансування ротору

Ротором називають ланки механізмів, які здійснюють обертальний рух та утримуються при цьому своїми несучими поверхнями в опорах ходів [6].

Якщо маса ротора розподілена щодо вісі обертання рівномірно, то головна центральна вісь інерції $X-X$ збігається з віссю обертання та ротор є врівноваженим або ідеальним. При розбіжності вісі обертання з віссю $X-X$, ротор буде неврівноваженим і в його опорах при обертанні виникнуть змінні реакції, викликані дією інерційних сил і моментів (точніше, рухом центру мас з прискоренням). Залежно від взаємного розташування вісі обертання і головної центральної вісі інерції $X-X$, розрізняють наступні види неврівноваженості роторів:

- статичну, коли ці осі паралельні;
- моментну, коли осі перетинаються в центрі мас ротора;
- динамічну, коли осі або перетинаються поза центром мас, або не перетинаються, а перехрещуються в просторі [6].

При балансуванні оперують дисбалансами, які представляють собою міру статичної неврівноваженості ротора. Це векторна величина, що дорівнює добутку неврівноваженої маси m на її ексцентриситет e , який представляє собою радіус-вектор центру цієї маси щодо вісі ротора [6].

Балансування – це технологічний процес суміщення головної вісі центральної інерції з віссю ротора. Балансуванням називають процес визначення значень і кутових координат дисбалансів ротора і їх зменшення за допомогою коригування розміщення його мас.

Жорсткий ротор можна врівноважити двома корегуючими масами, розташованими в двох довільно обраних площинах, перпендикулярних вісі його обертання. Ці площини називають площинами корекції [6].

Найбільш широке застосування на сьогоднішній день отримав метод балансування коригуванням мас, при якому коригувальні маси переміщують по ротору, встановлюють на ньому або видаляють таким чином, щоб головна центральна вісь інерції наближалася до вісі ротора. Переміщення, додавання або видалення коригувальної маси виконують в одній або декількох точках однієї площини корекції, перпендикулярної осі ротора, або в декількох паралельних площинах корекції одночасно або послідовно в кожній площині.

Процес балансування може бути послідовним, коли вимір дисбалансу та його зменшення складають самостійні операції, і суміщений, коли вимірювання та коригування мас відбувається одночасно [6].

Точність балансування характеризується величиною залишкового дисбалансу D_0 ротора в кожній з площин корекції. Величина D_0 не повинна перевищувати допустимих для даного класу точності значень, регламентованих [7].

У даній роботі в якості ротора виступає пристрій для розточування отвору в корпусній деталі. Це збірна одиниця, при обертанні якої на кожну i -ту деталь буде діяти відцентрова сила інерції P_i , що визначається за формулою (1.1):

$$\vec{P}_i = m_i \cdot \vec{r}_i \cdot \omega^2 = \vec{D}_i \cdot \omega^2, \quad (1.1)$$

де m_i – маса i -ої деталі, г;

ω - кутова швидкість обертання ротора, рад/с;

r_i – відстань центра i -ої маси від центра мас ротора, мм.

Для забезпечення стану динамічної рівноваги необхідно забезпечити умову рівності нулю суми всіх невіднованих сил і суми всіх моментів M_i цих сил згідно системи рівнянь (1.2):

$$\begin{cases} \sum_i^n \vec{P}_i = 0 \\ \sum_i^n \vec{M}_i = \sum_i^n h_i \cdot \vec{P}_i = 0 \end{cases}, \quad (1.2)$$

де h_i - відстань від центру координат до центру мас кожної деталі.

Сумарний момент дисбалансів ротора можна врівноважити парою рівних по модулю дисбалансів, розташованих у двох довільних площинах L і R , перпендикулярних осі ротора (рис. 1.3). У такому випадку повне (динамічне) врівноваження ротора вимагає використання тільки двох протипаг, а завдання балансування зводиться до знаходження чотирьох невідомих.

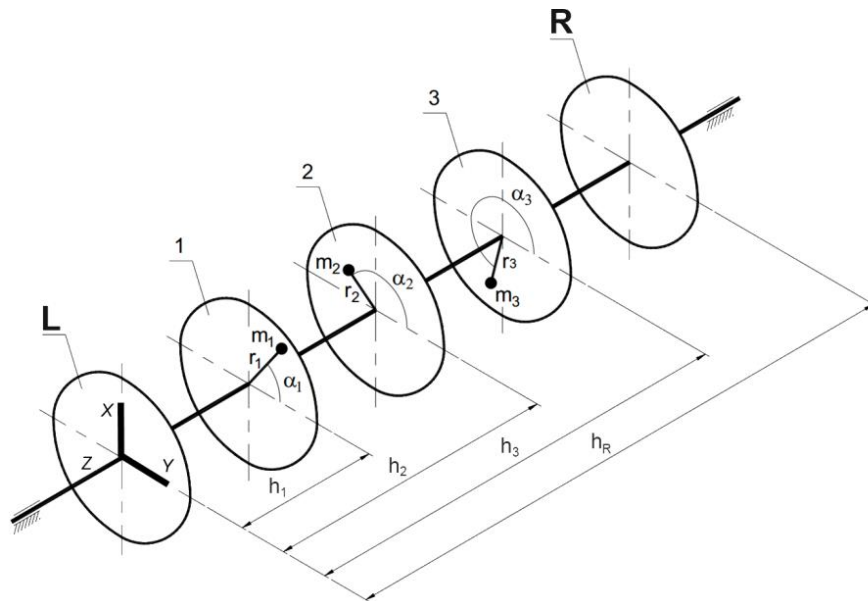


Рисунок 1.3 – Схема для вирішення завдання балансування ротору

Для вирішення цього завдання досить побудувати систему координат XYZ, початок якої належить одній з площин корекції, а вісь Z збігається з віссю обертання. Записавши окремо рівняння проєкцій дисбалансів на вісь X і вісь Y і моментів цих проєкцій дисбалансів, отримаємо систему рівнянь (1.3):

$$\begin{cases} \sum_i^n (m_i \cdot \vec{r}_i) + m_L \cdot \vec{r}_L + m_R \cdot \vec{r}_R = 0 \\ \sum_i^n (h_i \cdot m_i \cdot \vec{r}_i) + h_L \cdot m_L \cdot \vec{r}_L + h_R \cdot m_R \cdot \vec{r}_R = 0 \end{cases} . \quad (1.3)$$

Розв'язок цієї системи рівнянь дозволить знайти параметри коригувальних мас для балансування ротору.

1.4 Методика розрахунку характеристик міцності пристосувань

При проведенні розрахунку розглядаються такі випадки навантаження вузлів кріплень пристосувань:

а) досягнення максимального (мінімального) експлуатаційного перевантаження у вертикальній площині;

б) аварійна зупинка з виникненням інерційних перевантажень по осях пов'язаної з верстатом системи координат.

У центрі мас обраної деталі пристосування вагою G в розглянутих випадках навантаження діють окремо прикладені розрахункові навантаження за формулами (1.4):

$$\begin{aligned} P_X &= G \times n_x \times f, \\ P_Y &= G \times n_y \times f, \\ P_Z &= G \times n_z \times f, \end{aligned} \quad (1.4)$$

де n_x, n_y, n_z – максимально можливі складові перевантаження;

f – нормований коефіцієнт безпеки.

Для розрахунку в вертикальній площині приймаємо такі значення вертикального перевантаження та коефіцієнта безпеки: $n_{y\max} = 9,0$; $n_{y\min} = -2$; $f = 1,5$. Для випадку аварійної зупинки приймаємо найбільш складний варіант навантаження, передбачений нормами міцності при $f = 1,0$: поздовжнє перевантаження $n = 0 \dots 9$ при дії вперед і $n = 0 \dots 1,5$ при зворотному напрямку дії; вертикальне перевантаження $n = 0 \dots 4$ при направленні перевантаження вниз і $n = 0 \dots 2$ при направленні вгору; бічне перевантаження $n = 1,5$ при дії перевантаження вліво і вправо. Виникаючі при цьому інерційні навантаження вважаються чинними окремо.

У процесі розрахунку визначаємо запаси міцності конструкції в критичних (найбільш слабких) перетинах вузлів кріплень. Під запасом міцності в даному випадку розуміється відношення межі міцності матеріалу деталі (на розтягнення, зріз чи зминання) до розрахункового напруження. При проведенні розрахунку визначаються наступні параметри:

- межа міцності на розтяг;
- межа міцності на зріз;
- межа міцності на зминання;
- запас міцності матеріалу конструкції;
- межа плинності матеріалу.

Додатково виконується розрахунок кріплення пристосування на зминання. Площа критичного перерізу при роботі матеріалу деталі на зминання визначається за формулою (1.5):

$$F_{CM} = n \times d \times h, \quad (1.5)$$

де n – кількість отворів в деталі;

d – діаметр отвору;

h – товщина матеріалу деталі.

Чинне напруження зминання визначається за формулою (1.6):

$$s_{CM} = P_{(x, y, z)} / F_{CM} \quad , \quad (1.6)$$

Отриманому напруженню відповідає запас міцності, що визначається за виразом (1.7):

$$\eta = s_{пц\,CM} / s_{CM} \quad , \quad (1.7)$$

де $s_{пц\,CM} = 26 \text{ кгс/мм}^2$ – для матеріалу Д-16;

$s_{пц\,CM} = 27 \text{ кгс/мм}^2$ – для матеріалу 12Х18Н10Т 10 [8] - [9].

1.5 Постановка задач дипломної роботи

Як показали проведені дослідження, задача забезпечення точності обробки корпусних деталей потребує комплексного підходу. Похибки обробки можуть бути зменшені шляхом забезпечення стабільного положення деталі та інструмента в системі координат верстата:

- раціональним підбором конструкцій та параметрів опорно-установочних елементів пристосувань;
- підбором коригувальних мас для балансування ротору, тобто пристрою для розточування корпусної деталі.

Метою роботи є розробка програмного забезпечення для автоматизації інженерних розрахунків параметрів технологічного обладнання для підвищення точності обробки корпусних деталей.

Задачі дипломної роботи:

- дослідити функціональні можливості існуючих систем автоматизації інженерних розрахунків;

- виконати проєктування програмного комплексу автоматизації інженерних розрахунків (ПКАІР);
- виконати конструювання ПКАІР;
- практично застосувати ПКАІР для визначення параметрів міцності деталі пристосування та балансування ротору при обробці корпусних деталей;
- розробити програмну документацію.

2 ДОСЛІДЖЕННЯ ТА ПРАКТИЧНЕ ВИКОРИСТАННЯ СИСТЕМ АВТОМАТИЗАЦІЇ ІНЖЕНЕРНИХ РОЗРАХУНКІВ

2.1 Огляд існуючих систем автоматизації інженерних розрахунків

Протягом багатьох років ключовими гравцями на ринку програмних систем комп'ютерного інжинірингу (Computer-Aided Engineering, CAE) залишаються компанії ANSYS, Inc (ANSYS), Dassault Systèmes (SIMULIA/Abaqus), Altair Engineering (Altair HyperWorks), Siemens (NX Nastran) та PTC (Creo Simulate).

ANSYS – це відома система, що працює на основі геометричного ядра Parasolid та методу скінчених елементів (MCE). Дозволяє здійснювати розв'язок різноманітних задач механіки деформованого твердого тіла та механіки конструкцій, механіки рідини й газу, теплопередачі й теплообміну, електродинаміки, акустики, а також зв'язаних полів [10].

Компанія розробник Ansys Inc. також пропонує інші системи моделювання, в тому числі DesignSpace, AI Solutions (NASTRAN, ICEM CFD); призначені для використання в більш специфічних галузях виробництва. Пропоновані компанією засоби чисельного моделювання та аналізу сумісні з деякими іншими пакетами, працюють на різних ОС. Програмна система ANSYS сполучається з відомими CAD-системами Unigraphics, CATIA, Pro / ENGINEER, SolidEdge, SolidWorks, Autodesk Inventor і деякими іншими [11].

Оновлена версія Ansys 2020 R2 надає покращені функції чисельного моделювання, ресурси для суперкомп'ютерів і платформні рішення для забезпечення глобальної взаємодії команд і обміну інформацією [11].

Оновлення хмарного сервісу Ansys Cloud, такі як підтримка віртуальних робочих столів, дозволяють об'єднати можливості флагманських програмних продуктів для комп'ютерного моделювання Ansys з високим

ступенем масштабованості обчислювальної потужності, що надається хмарними технологіями. Розрахункові платформи були доповнені потужними інтуїтивно зрозумілими робочими процесами для проведення параметричної оптимізації, управління даними й конфігураціями (в тому числі даними про матеріали), візуалізації залежностей (рис. 2.1) і підтримки прийняття рішень. Технології створення цифрових двійників Ansys забезпечують віддалений контроль виробів і є критично важливим компонентом для проведення обслуговування за технічним станом [12].

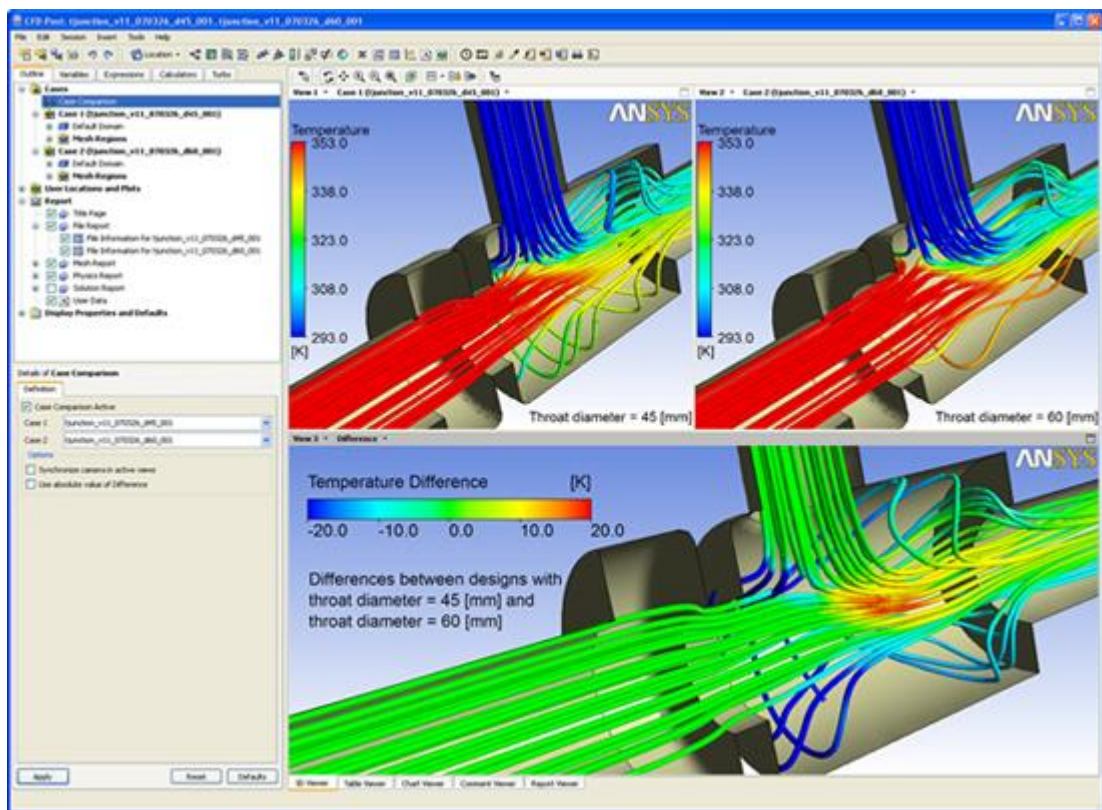


Рисунок 2.1 – Приклад реалізації проєкту в Ansys [13]

SIMULIA – це масштабований набір інструментів для вирішення безлічі інженерних задач, від типових до унікальних за рівнем складності. SIMULIA надає пакет уніфікованих рішень для аналізу, за допомогою яких користувачі можуть спільно працювати, безперешкодно обмінюватися даними моделювання та застосовувати перевірені методи без шкоди для достовірності інформації [14].

Додатки SIMULIA прискорюють процес оцінки робочих характеристик, надійності й безпеки матеріалів та продуктів перед виготовленням фізичних прототипів. Використання SIMULIA дозволяє порівнювати, оцінювати та оптимізувати проекти на всіх етапах розробки, щоб підвищувати ефективність кінцевого продукту при зниженні витрат.

Виробники та постачальники в авіакосмічній та оборонній промисловості використовують рішення SIMULIA як частину інтегрованого середовища розробки для оцінки варіантів проектного рішення, спільної роботи над проектами та залучення обчислювальних ресурсів для більш ефективного аналізу [14].

Пакет продуктів Abaqus Unified FEA (Finite Element Analysis) пропонує потужні комплексні рішення не тільки для роботи з поточними завданнями, а й для усунення складних інженерних проблем в самих різних галузях. Так, наприклад, в автомобільній промисловості робочі технічні групи на основі структури даних моделювання та інтегрованої вирішальною технології можуть здійснювати розрахунок повного навантаження автомобіля, динамічних вібрацій, багатомодульних систем, наслідків удару / аварії, проводити нелінійний статичний аналіз і визначати теплову і акустико-структурну зв'язку. Провідні галузеві компанії використовують пакет Abaqus Unified FEA для консолідації процесів та інструментів, скорочення витрат, підвищення ефективності та отримання конкурентних переваг [15].

Платформа HyperWorks, розроблена компанією Altair Engineering, Inc., є світовим лідером в області CAE-систем. Вона об'єднує кращі на сьогоднішній день інструменти моделювання, розрахунку, аналізу та структурної оптимізації. Altair HyperWorks – це ціле сімейство програмних продуктів різної спрямованості: HyperMesh, HyperCrash, HyperView, OptiStruct, AcuSolve, RADIOSS (рис. 2.2). Протягом останніх десятиліть їх використовують лідери різних галузей промисловості по всьому світу. Програмні продукти HyperWorks націлені на вирішення найважливіших і

актуальних проблем – це підвищення якості моделювання та точності отриманих результатів, зниження часових витрат інженерів і комп'ютерного часу на обчислення, розв'язання найскладніших завдань нелінійної механіки і структурної оптимізації [16].

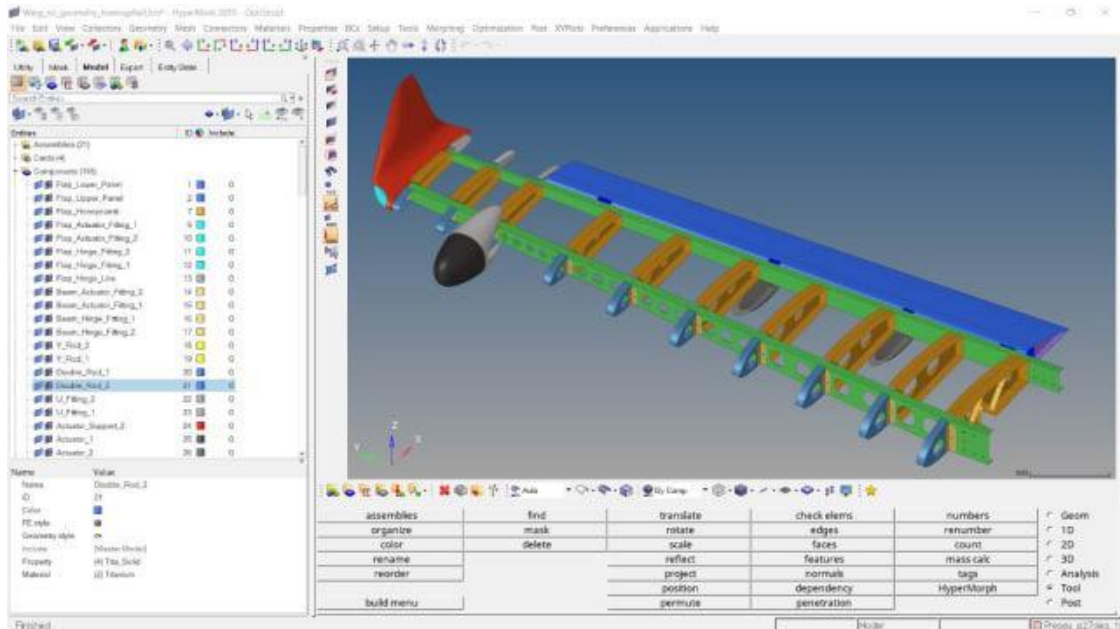


Рисунок 2.2 – Приклад реалізації проекту в Altair HyperWorks [17]

Програмне забезпечення (ПЗ) NX Nastran компанії Siemens на сьогоднішній день доступно в хмарному середовищі, що динамічно розвивається, заснованому на технології SaaS (Software as a Service). Нове спільне рішення об'єднує високопродуктивні комп'ютерні обчислення, що надаються на вимогу, з провідним ПЗ для моделювання на основі МСЕ, що надаються Rescale. Інженери тепер можуть легко отримувати доступ до необхідних саме для них обчислювальних потужностей для виконання комп'ютерного моделювання. Вебплатформа Rescale дозволяє використовувати переваги NX Nastran, що позитивно впливає на вартість великомасштабного моделювання. Це призводить до підвищення ефективності роботи, якості продукції та скорочення часу виведення продукту на ринок.

Використовуючи платформу Rescale, користувачі NX Nastran можуть одночасно виконувати розрахунки для сотень моделей. Оплата користування ресурсами при цьому буде розраховуватися в залежності від фактичного використання. Платформа Rescale підвищує значущість NX Nastran, надаючи користувачам можливість ефективно виконувати масштабні розрахунки, забезпечуючи надійний захищений доступ до ПЗ [18].

Розширення PTC Creo Simulation і PTC Creo Advanced Simulation, взаємодіючи з PTC Creo Parametric, Pro/ENGINEER і PTC Creo Elements / Pro, забезпечують проведення структурного, частотного і термічного аналізу для вивчення робочих характеристик виробів і оптимізації цифрових конструкцій. Види можливого аналізу:

- структурний – використовується для встановлення структурної цілісності компонентів в умовах реальних обмежень і постійного навантаження;
- тепловий – використовується для вивчення наслідків сильного нагрівання або охолодження в умовах реальних обмежень і постійного навантаження;
- частотний – оцінка й розрахунок власної частоти коливань;
- аналіз критичного навантаження – розрахунок коефіцієнтів критичного навантаження [19].

Система Creo (Pro/ENGINEER) компанії PTC, дозволяє в повній мірі охопити всі стадії підготовки виробництва – від розробки концептуальної ідеї нового виробу до створення керуючих програм для верстатів з ЧПУ. Наприклад, крім традиційних модулів для конструкторського і технологічного проектування до системи входить аналітичний модуль Pro/Mechanica. Призначення цього модуля – оптимізація конструкції виробу за масовими, міцнісними та експлуатаційними критеріями Pro/MECHANICA включає в себе три продукта: Pro/MECHANICA STRUCTURE, Pro/MECHANICA THERMAL, Pro/MECHANICA MOTION [20].

Результати порівняльного аналізу рішень компанії PTC для інженерних розрахунків та аналізу наведено в табл. 2.1. Система Pro/Mechanica є найбільш простою у використанні та надає базовий функціонал.

Таблиця 2.1 – Порівняльний аналіз CAE-систем від PTC

Функції	Pro/ Mechanica	Creo Simulation	Creo Advanced Simulation
Статичний структурний аналіз	+	+	+
Симуляція роботи деталей та складань	+	+	+
Аналіз критичного навантаження	-	+	+
Аналіз твердотільних елементів	-	+	+
Оптимізація конструкції	-	+	+
Нелінійність великих деформацій	-	-	+
Динамічний аналіз	-	-	+

2.2 Структурний аналіз деталі кріплення в Pro/Mechanica

Для проведення інженерного аналізу в Pro/Mechanica потрібно виконати наступні етапи (рис. 2.3) [21]:

- розробка моделі;
- аналіз моделі;
- визначення змін моделі;
- оптимізація моделі.

Для того, щоб провести аналіз міцності за допомогою MCE спочатку потрібно створити деталь. Запускаємо Pro/Engineering і створюємо новий проєкт. Результат розробки 3D-моделі деталі показано на рис. 2.4.

Для моделювання навантажень потрібно вибрати в меню Applications - Mechanica. Після чого ми потрапимо у вбудований режим роботи Pro/Mechanica середовища Pro/Engineering.



Рисунок 2.3 – Етапи інженерного аналізу проєкту

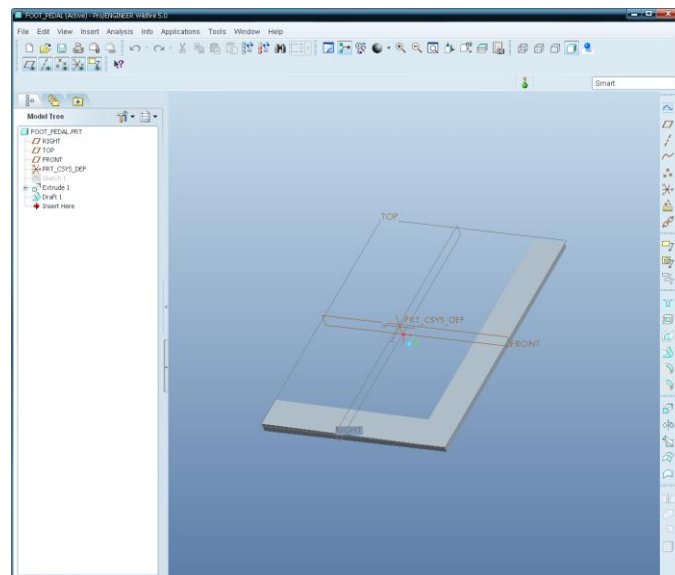


Рисунок 2.4 – Деталь пристосування

Для аналізу міцності потрібно задати силу тиску на певну частину деталі пристосування. Для цього заходимо в меню Insert -> Force / Moment Load. У діалоговому вікні задаємо назву навантаження в полі Name, в списку References вибираємо тип елементів, до яких буде присвоєно вплив сили, за допомогою покажчика мишки вибираємо елементи і в групі компонентів Force вводимо значення в заданих нижче одиницях для відповідної площині. Тепер приступимо до створення блокових елементів. Для цього виберемо в меню Insert -> Beam. З'явиться діалогове вікно наведене на рис. 2.5.

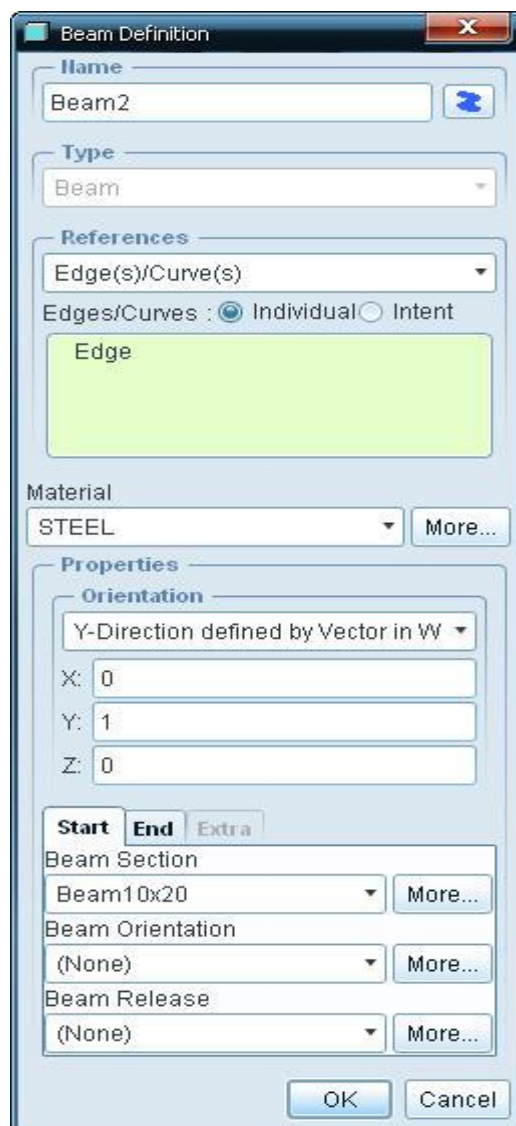


Рисунок 2.5 – Діалогове вікно Beam Definition

В полі Name задаємо назву, потім вибираємо грань/криву, на якій будуть розміщені блокові елементи, матеріал і орієнтацію в просторі. Тепер задаємо розмір наших секцій. Натискаємо на кнопку More напроти Beam section. З'явиться вікно показане на рис. 2.6.

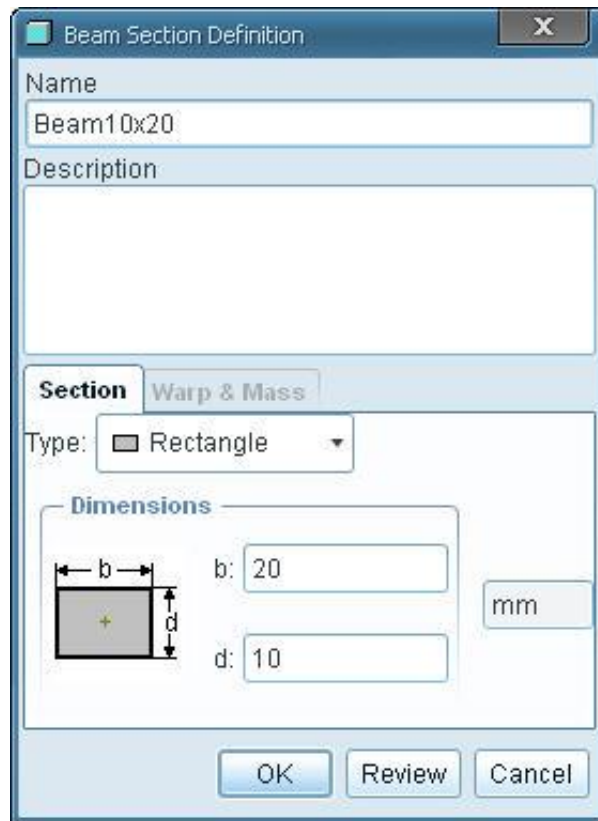


Рисунок 2.6 – Діалогове вікно Beam Section Definition

У цьому вікні задаємо тільки назву і розмір елементів, натискаємо кнопку OK. Повертаємося до вікна Beam Definition і зі списку Beam Section вибираємо потрібний нам розмір блокових елементів.

Після цього необхідно задати обмеження. Оскільки наша деталь фіксується, то ми задаємо для нерухомої частини обмеження. Вибираємо в меню Insert -> Displacement Constraint (рис. 2.7). У цьому вікні задаємо ім'я обмеження, площину або грань, яку фіксуємо, а так само ступінь свободи обраного елемента в кожній з трьох площин. У даному випадку вибираємо всюди Fixed. Після чого модель повинна виглядати, як показано на рис. 2.8.

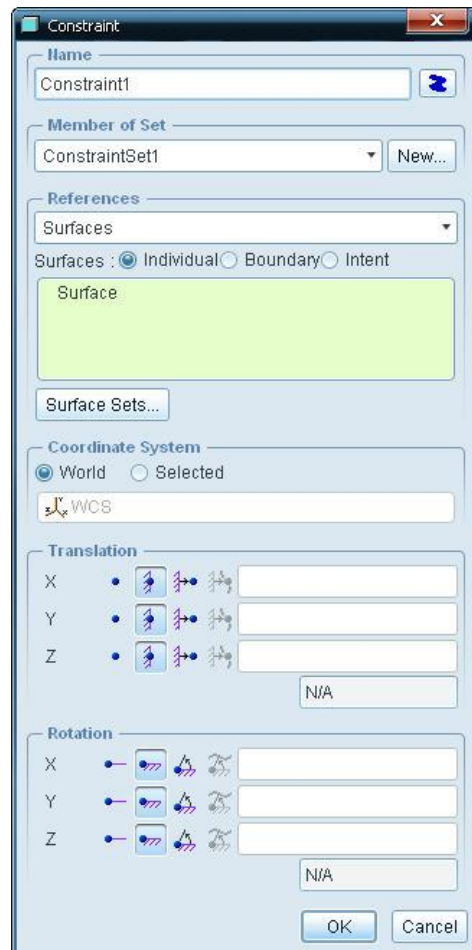


Рисунок 2.7 – Діалогове вікно Displacement Constraint

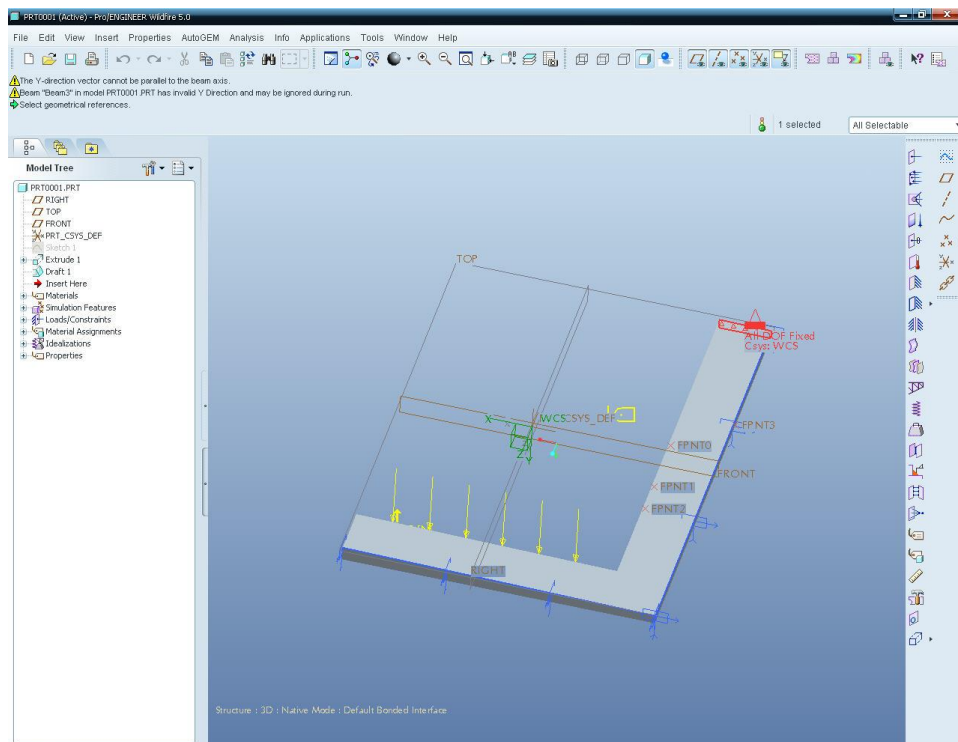


Рисунок 2.8 – Модель після задання всіх параметрів

Після того, як для моделі задані всі параметри, можна починати аналіз. Для цього треба обрати в меню Analysis -> Mechanics Analyses / Studies або просто натиснути на відповідну кнопку на панелі інструментів. З'явиться вікно аналізу (рис. 2.9).

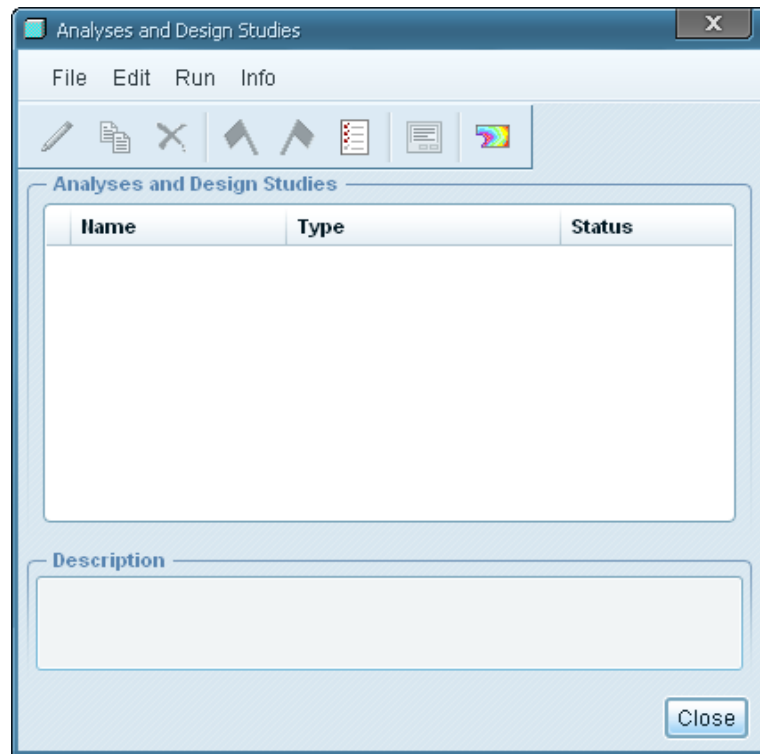


Рисунок 2.9 – Діалогове вікно Analyses and Design Studies

Для створення нового аналізу вибираємо File -> New Static. У діалоговому вікні (рис. 2.10) задаємо ім'я аналізу в полі Name, в блоках Constraints і Loads вибираємо потрібні нам для розрахунків значення обмежень і навантажень відповідно. У списку Method вибираємо, яким методом буде проведено аналіз. У нашому випадку вибираємо Multi-Pass Adaptive і натискаємо ОК. Створений нами аналіз з'явився в списку (рис. 2.11).

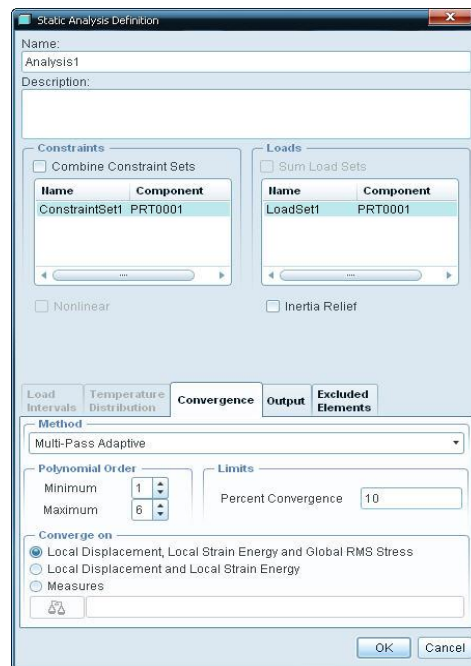


Рисунок 2.10 – Діалогове вікно Static Analysis Definition

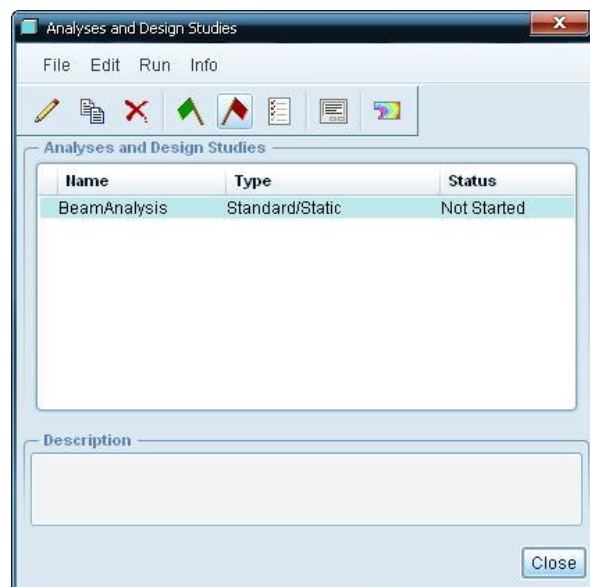
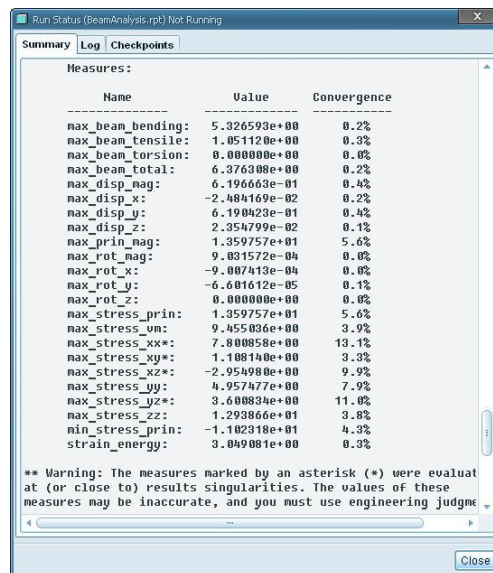


Рисунок 2.11 – Діалогове вікно аналізу

Для запуску аналізу натискаємо кнопку Start . Для перегляду поточного статусу аналізу потрібно натиснути на кнопку Display study status , після чого з'явиться діалогове вікно з поточним станом аналізу (рис. 2.12).

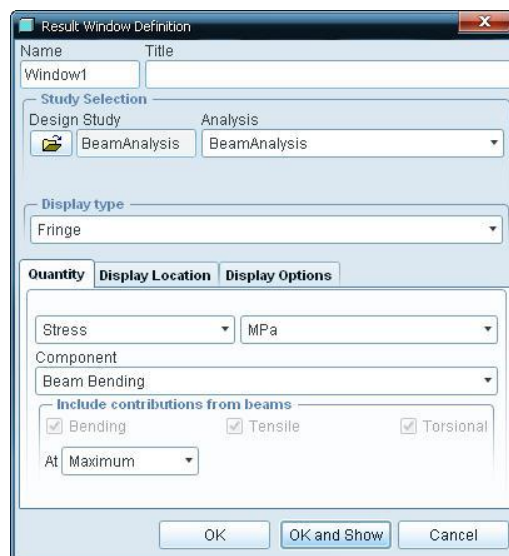


Measure	Value	Convergence
max_beam_bending:	5.326593e+00	0.2%
max_beam_tensile:	1.051120e+00	0.3%
max_beam_torsion:	0.000000e+00	0.0%
max_beam_total:	6.376308e+00	0.2%
max_disp_mag:	6.196663e-01	0.4%
max_disp_x:	-2.484169e-02	0.2%
max_disp_y:	6.190423e-01	0.4%
max_disp_z:	2.354799e-02	0.1%
max_prin_mag:	1.359757e+01	5.6%
max_rot_mag:	9.031572e-04	0.0%
max_rot_x:	-9.007413e-04	0.0%
max_rot_y:	-6.601612e-05	0.1%
max_rot_z:	0.000000e+00	0.0%
max_stress_prin:	1.359757e+01	5.6%
max_stress_vm:	9.455036e+00	3.9%
max_stress_xx*:	7.800858e+00	13.1%
max_stress_xy*:	1.108140e+00	3.3%
max_stress_xz*:	-2.954990e+00	9.9%
max_stress_yy:	4.957477e+00	7.9%
max_stress_yz*:	3.600834e+00	11.0%
max_stress_zz:	1.293866e+01	3.8%
min_stress_prin:	-1.102318e+01	4.3%
strain_energy:	3.049081e+00	0.3%

** Warning: The measures marked by an asterisk (*) were evaluated at (or close to) results singularities. The values of these measures may be inaccurate, and you must use engineering judgment.

Рисунок 2.12 – Чисельні дані аналізу

Після закінчення аналізу, для перегляду результатів потрібно натиснути на кнопку Review results . У вікні, потрібно вказати, в якому вигляді і які дані потрібно отримати. Наприклад, треба переглянути, яке було навантаження, представивши нашу деталь у вигляді променя, на якому розташовувалися створені нами блокові елементи. Для цього вибираємо показники відображення, як показано на рис. 2.13. Після того, як всі параметри відображення встановлені, натискаємо OK and Show. Результат показаний на рис. 2.14.



Result Window Definition

Name: Window1 Title:

Study Selection

Design Study: BeamAnalysis Analysis: BeamAnalysis

Display type

Fringe

Quantity **Display Location** **Display Options**

Stress: MPa

Component: Beam Bending

Include contributions from beams

☒ Bending ☒ Tensile ☒ Torsional

At: Maximum

OK OK and Show Cancel

Рисунок 2.13 – Вікно параметрів відображення результатів

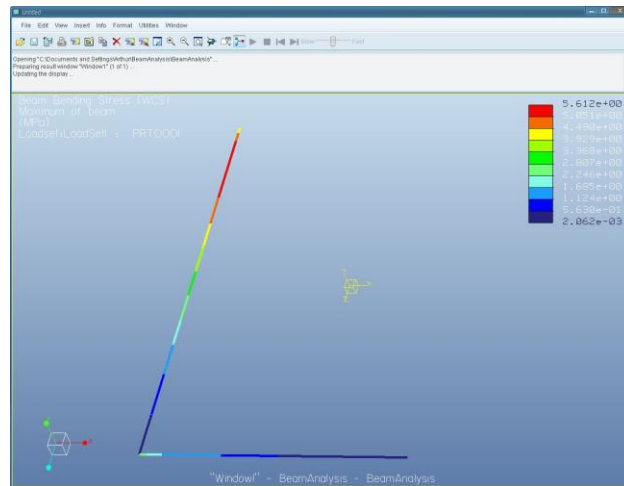


Рисунок 2.14 – Результати аналізу

Якщо ж змінити показники відображення, як показано на рис. 2.15, в результаті буде представлено тривимірну модель деталі, як на рис. 2.16.

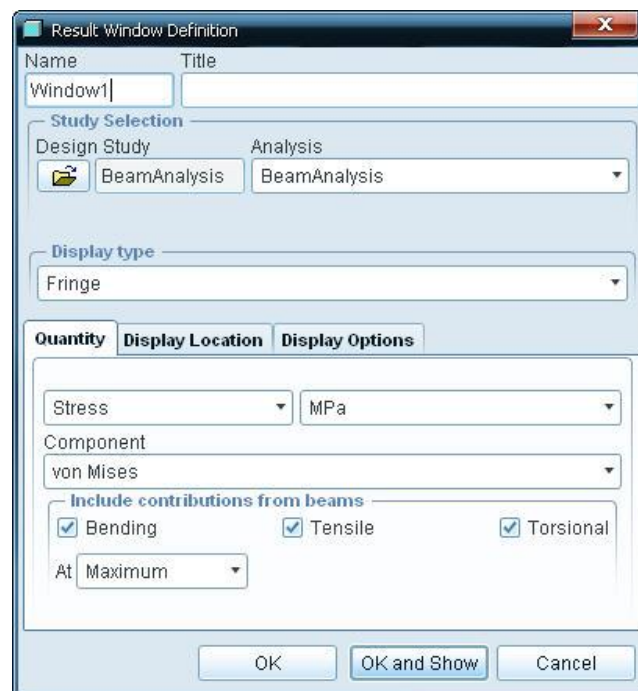


Рисунок 2.15 – Вікно параметрів відображення результатів

Таким чином, в результаті застосування пакету Pro/Mechanica було виконано аналіз показників міцності деталі пристосування для кріплення заготовки.

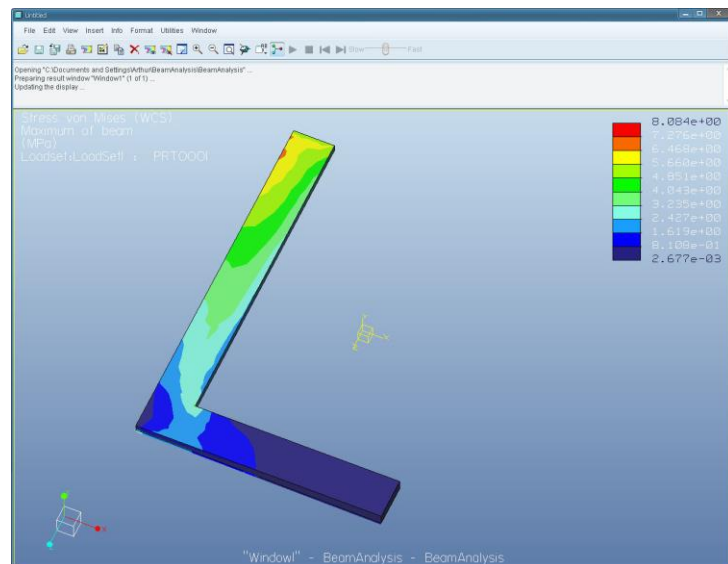


Рисунок 2.16 – Візуалізація результатів аналізу

2.3 Висновки до розділу 2

Як показали проведені дослідження, існує багато пакетів для автоматизації інженерних розрахунків. Здебільшого вони базуються на методі скінчених елементів та дозволяють дослідити параметри та характеристики різних об'єктів та фізичних процесів.

Але, ці пакети є платними, вимогливими до обчислювальних ресурсів та складними у вивченні та практичному застосуванні.

Серед доступних систем автоматизації інженерних розрахунків було обрано пакет Pro/Mechanica, в якому фактично можливо провести лише статичний структурний аналіз та визначити міцність конструкції. Цього недостатньо для розрахунку та забезпечення показників для узгодження положення заготовки та інструмента в системі координат верстата. Зокрема, цей пакет не дозволяє розрахувати маси противаг для динамічного балансування ротору, а також додаткові параметри міцності деталей пристосувань для кріплення заготовки.

Тому, актуальною є задача розробки програмного комплексу автоматизації інженерних розрахунків (ПКАІР), що надасть відповідні функціональні можливості.

3 РОЗРОБКА ПРОГРАМНОГО КОМПЛЕКСУ АВТОМАТИЗАЦІЇ ІНЖЕНЕРНИХ РОЗРАХУНКІВ

3.1 Призначення та функціональні можливості ПКАІР

ПКАІР призначений для автоматизації інженерних розрахунків параметрів технологічного обладнання при механічній обробці корпусних деталей.

Функціональні можливості ПКАІР включають в себе:

- введення вхідних даних в зручному для користувача вигляді;
- розрахунок показників міцності деталей пристосувань для кріплення заготовки деталі на верстаті з ЧПУ;
- розрахунок мас і просторового положення противаг для динамічного врівноваження ротора верстата;
- візуалізація результатів розрахунків;
- збереження в файл для подальшої передачі в систему автоматизації технологічної підготовки виробництва.

3.2 Структура ПКАІР

Узагальнену структуру ПКАІР наведено на рис. 3. 1.

Модуль розрахунку на міцність деталей пристосувань дозволяє виконати наступні види розрахунків:

- а) визначення діючих на пристосування осьових навантажень (поздовжніх, поперечних, вертикальних);
- б) визначення запасів міцності конструкції пристосування в критичних (найбільш слабких) перетинах вузлів кріплень, де в залежності від матеріалу основи визначається:
 - 1) межа міцності на розтяг;
 - 2) межа міцності на зріз;
 - 3) межа міцності на зминання;

- 4) запас міцності матеріалу конструкції;
- 5) межа плинності матеріалу.

Модуль розрахунку параметрів ротора дозволяє наступне:

- визначення маси та координат центру мас кожної деталі;
- визначення радіус-векторів дисбалансів всіх елементів конструкції;
- складення системи рівнянь, використовуючи проєкції радіус-векторів дисбалансів на вісі X і Y , а також моменти цих складових;
- зведення до мінімуму динамічного дисбалансу ротору, змінюючи кількість пластин N комплекту № 1 і пластин M комплекту № 2, а також положення комплекту № 1 щодо вісі обертання;
- підбір противаг для динамічного балансування ротору з наперед заданого набору двох комплектів пластин;
- розрахунок положення комплекту пластин відносно вісі обертання.

Модуль візуалізації результатів розрахунку забезпечує виведення технологічних параметрів в форму та збереження в файл для передачі в систему автоматизації технологічної підготовки виробництва.

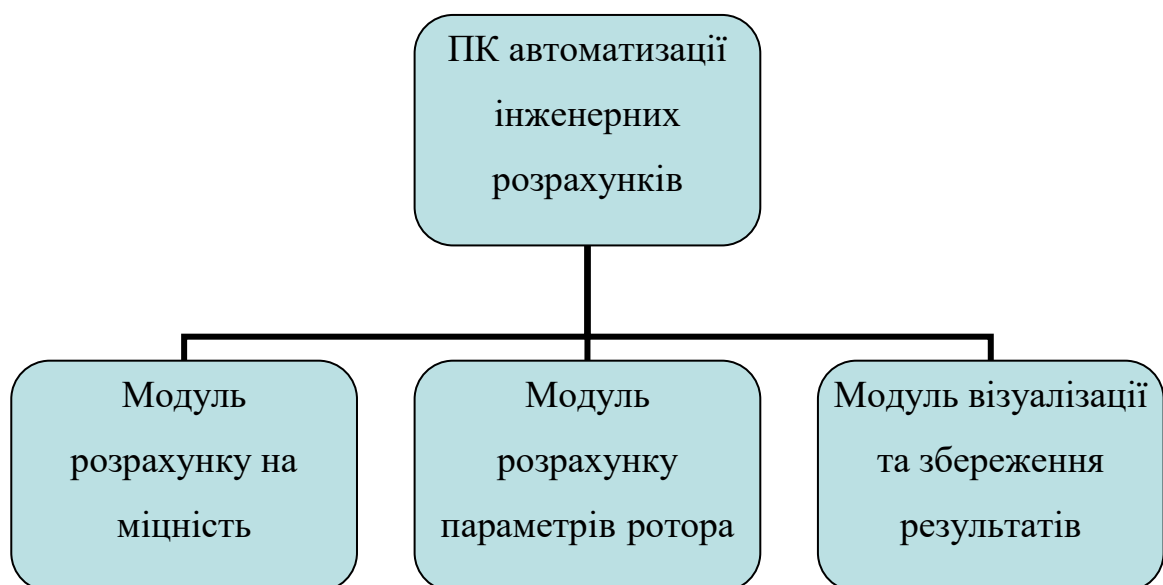


Рисунок 3.1 – Узагальнена структура ПК АІР

3.3 Вибір інструментарію розробки

Для реалізації проєкту було обрано C++ мову програмування, яка вивчалася, та середовище Qt Creator [22] - [24].

C++ – це компільована мова програмування загального призначення. Підтримує різні парадигми програмування: процедурну, узагальнену, функціональну. Але найбільшу увагу приділяє підтримці об'єктно-орієнтованого програмування [22]. На відміну від C, мова C++ дозволяє писати програми в об'єктно-орієнтованому стилі, представляючи програму як сукупність взаємодіючих між собою класів і об'єктів, що спрощує створення великих застосунків.

C++ є мовою зі статичною типізацією та підходить для створення самих різних застосунків. На сьогоднішній день C++ є однією з найпопулярніших і поширених мов. C++ є потужною мовою, яка успадкувала від мови C багаті можливості по роботі з пам'яттю. C++ можна використовувати в програмах будь-якого рівня, де важливі швидкість роботи та продуктивність. Нерідко вона застосовується для створення графічних застосунків, різних прикладних програм програмування [23].

C++ є компільованою мовою, а це значить, що компілятор транслює вихідний код на C++ в виконуваний файл, який містить набір машинних інструкцій. Але різні платформи мають свої особливості, тому скомпільовані програми не можна просто перенести з однієї платформи на іншу і там вже запустити. Однак на рівні вихідного коду програми на C++ володіють переносимістю, якщо не використовуються якісь специфічні для поточної ОС функції. А наявність компіляторів, бібліотек та інструментів розробки майже під всі поширені платформи дозволяє компілювати один і той же вихідний код на C++ в застосунки під ці платформи програмування [23].

Для розробки програм на C++ необхідний компілятор, який транслює вихідний код на мові C++ в виконуваний файл, який потім можна запускати. Зараз є дуже багато різних компіляторів, які можуть відрізнятися за різними

аспектами, зокрема, реалізацією стандартів. Прикладом є вільно розповсюджуваний компілятор g++.

Також для створення програм на C ++ можна використовувати інтегровані середовища розробки (Integrated Development Environment, IDE), такі як Visual Studio, Eclipse, Qt Creator і т.д. [23].

Visual Studio – це широко відоме повнофункціональне середовище розробки від компанії Microsoft, яке дозволяє працювати з такими платформами, як Windows, Інтернет, хмара і Android. Можливості IDE Visual Studio дозволяють правильно та ефективно писати код, реорганізовувати, аналізувати та виправляти проблеми з кодом. Система IntelliSense пропонує API-інтерфейси по мірі їх написання й виконує автоматичне завершення для підвищення швидкості та ефективності розробки [24].

Особливості Visual Studio:

- свій компілятор MSVC;
- існують версії, що працюють на macOS і Windows.;
- підтримує такі мови: ASP.NET, Ajax, DHTML, Visual C ++, JavaScript, JScript, Visual Basic, Visual C #, Visual F #, XAML і ін.

Плюси IDE Visual Studio:

- безкоштовно розповсюджується Visual Studio Community, з достатнім набором можливостей;
- платні версії можуть надаватися навчальним закладам і студентам безкоштовно;
- зручна система розумного автодоповнення;
- велика кількість налаштувань середовища розробки під «себе», завдяки вбудованим механізмам і доступним доповненням.

Мінуси IDE Visual Studio:

- версії Visual Studio Professional і Visual Studio Enterprise є платними;
- IDE досить вимоглива до апаратних ресурсів;
- ряд розробників вважають її надлишковою для створення невеликих проєктів;

- досить назька швидкість запуску проєктів і програм;
- не працює на Linux;
- платні версії з розширеним функціоналом поширюються за підпискою [24].

IDE Eclipse, заснована на платформі Eclipse, є однією з провідних IDE для розробників на C ++ та C. Eclipse пропонує підтримку розробки проєктів і кероване середовище розробки з різними наборами інструментів. Він також підтримує різні інструменти, сортування, навігацію за гіперпосиланнями, редактор коду з функцією підсвічування синтаксису, інструменти візуального налагодження і багато іншого.

Це безкоштовне ПЗ з відкритим вихідним кодом, що працює з усіма основними ОС, включаючи Windows, Mac OS X і Linux. Eclipse має зручний інтерфейс.

Особливості IDE Eclipse:

- а) працює на таких ОС як Windows, macOS, Linux;
- б) працює з такими мовами як C, C ++, Java, Perl, PHP, Python, Ruby та інші;
- в) працює з такими компіляторами:
 - 1) GCC / G ++;
 - 2) MinGW;
 - 3) Cygwin.

Плюси IDE Eclipse:

- простота установки і застосування;
- зручний механізм розробки графічних інтерфейсів;
- інструмент з відкритим вихідним кодом;
- розповсюджується безкоштовно;
- кросплатформеність;
- велика кількість доступних модулів розширення; багато розробники відзначають, що можливості модулів цього IDE значно випереджають такі у інших середовищах розробки.

Мінуси IDE Eclipse: ряд розробників відзначають незручність використання механізмів автодоповнення при написанні коду [24].

QT Creator – зручне і швидке інтегроване середовище розробки C ++, що пропонує цікаві можливості для розробника. Це кроссплатформне IDE має сучасний редактор коду C ++, вбудований зручний графічний інтерфейс, дизайнер форм, інструменти навігації та багато іншого. Серед розробників багато хто вибирає його за його API і бібліотеки, оскільки вони послідовні, докладні, зручні і добре документовані.

QT Creator поставляється з повним набором інструментів розробника, які призначені для одночасного створення застосунків і користувальницьких інтерфейсів, а потім для подальшого їх розгортання в різних мобільних ОС і настільних ПК. Часто відзначають простоту та інтуїтивність інтерфейсу. Він включає зручний редактор коду з можливістю підсвічування синтаксису та автозаповненням, можливість створення інтерфейсу користувача використовуючи перетягування, інструмент візуального налагодження та ін.

Особливості IDE QT Creator:

а) доступні версії під Linux, macOS, Windows;

б) підтримувані компілятори:

- 1) GCC / G ++;
- 2) MinGW;
- 3) Linux ICC;
- 4) Clang;
- 5) Nim;
- 6) QCC;
- 7) MSVC.

в) доступні розширення, що дозволяють працювати з такими мовами як Python, Ruby, Java, PHP і іншими.

Плюси IDE QT Creator:

- наявність безкоштовної версії;
- інтелектуальна система підказок;

- можливість компіляції проєктів для різних систем.

Мінуси IDE QT Creator:

- платна версія від 459 \$ в місяць;
- безкоштовна версія має ряд обмежень [24] - [26].

Результати порівняльного аналізу середовищ розробки зведено до табл. 3.1.

Таблиця 3.1 – Порівняння середовищ розробки

	Visual Studio	Eclipse	Qt Creator
Кількість підтримуваних компіляторів	1	3	7
Кросплатформність	-	+	+
Безкоштовна версія	-	+	+
Невимогливість до апаратних ресурсів	-	+	+
Зручний редактор коду	+	-	+

3.4 Розробка алгоритмів функціонування ПКАІР

Схему алгоритму функціонування модулю розрахунку на міцність наведено на рис. 3.2:

Блок 1. Початок роботи модулю.

Блок 2. Введення вхідних даних для розрахунку:

- вага деталі пристосування, кг;
- складові перевантаження;
- коефіцієнт безпеки;
- кількість отворів кріплення, шт;
- діаметр отвору, мм;
- товщина матеріалу, мм;

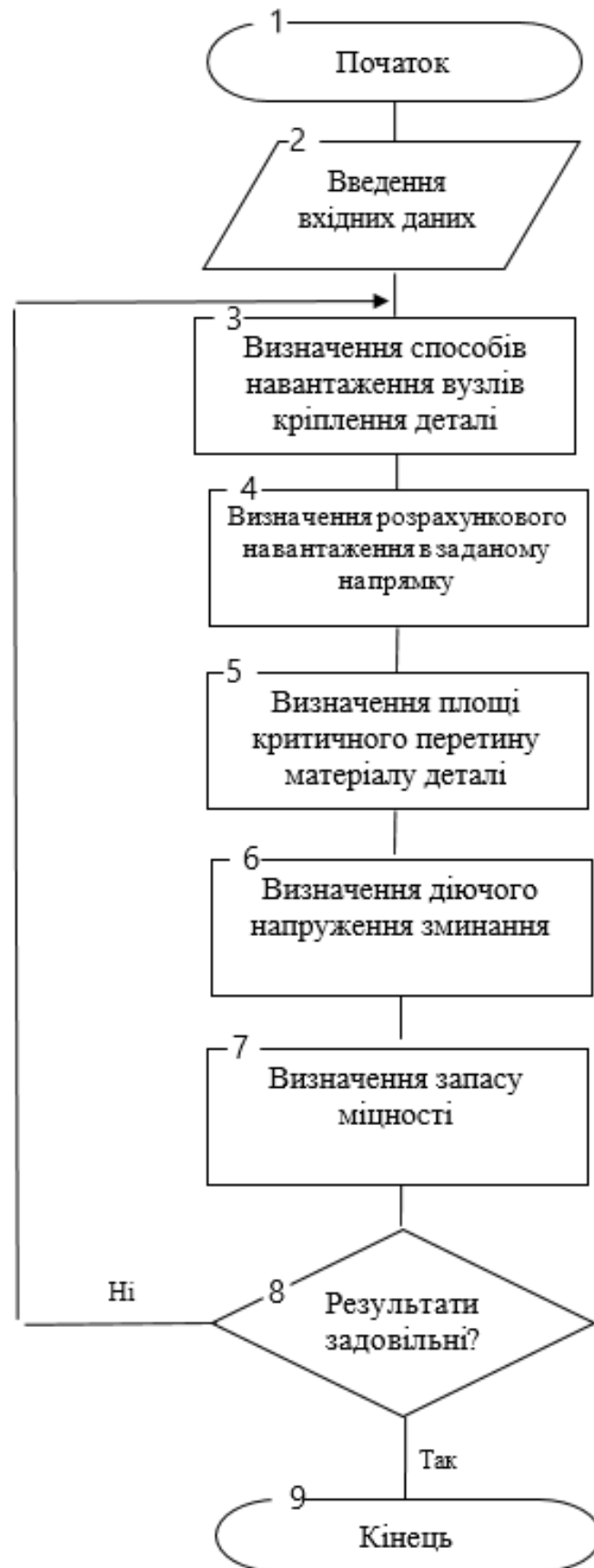


Рисунок 3.2 – Схема алгоритму роботи модуля розрахунку на міцність

Блок 3. Визначення способів навантаження вузлів кріплення деталі пристосування:

- досягнення максимального (мінімального) експлуатаційного перевантаження у вертикальній площині;
- аварійна зупинка з виникненням інерційних перевантажень по осях пов'язаної з верстатом системи координат.

Блок 4. Визначення розрахункового навантаження в заданому напрямку. Передбачає розрахунок діючих на пристосування осьових навантажень (поздовжніх, поперечних, вертикальних).

Блок 5. Визначення площі критичного перетину матеріалу деталі пристосування для кріплення заготовки.

Блок 6. Визначення діючого напруження зминання.

Блок 7. Визначення запасу міцності конструкції в критичних (найбільш слабких) перетинах вузлів кріплень. Під запасом міцності в даному випадку розуміється відношення межі міцності матеріалу деталі (на розтягнення, зріз чи зминання) до розрахункового напруження.

При цьому розраховуються:

- межа міцності на розтяг;
- межа міцності на зріз;
- межа міцності на зминання;
- запас міцності матеріалу конструкції;
- межа плинності матеріалу.

Блок 8. Перевірка результатів на відповідність нормам. Якщо результати відповідають встановленим експлуатаційним параметрам, то вони можуть бути збережені в файл. У випадку незадовільних результатів, потрібно повернутися на Блок 3 та перевизначити спосіб навантаження вузлів кріплення деталі.

Блок 9. Кінець. Завершення роботи модуля розрахунку на міцність.

Схему алгоритму роботи модулю розрахунку параметрів ротора представлено на рис. 3.3.

Блок 1. Початок роботи модулю.

Блок 2. Введення вхідних даних. Вихідними даними є:

- кількість елементів конструкції;
- відстань між площинами корекції, мм;
- координати по вісям x та y центра мас деталі, мм;
- відстань від точки відліку до центра мас деталі, мм;
- маса неурівноваженого тіла, г.

Блок 3. Ініціалізація масиву деталей. Потрібно вважати, що:

- частота обертання ротору: 3000 об / хв;
- всі деталі виготовлені із сталі з щільністю 7700 кг / м³;
- в якості противаг для балансування використовуються пакети пластин пакет № 1 з N шайб розмірами $\varnothing 100 \times 1$ мм і центральним отвором $\varnothing 16$ мм, і пакет № 2 з M шайб розмірами $\varnothing 50 \times 2$ мм (кількість N і M визначається залежно від розрахованих дисбалансів).

– пакет шайб № 1 кріпиться гвинтами в паз планшайби та може переміщатися уздовж цього паза в межах $f = 80$ мм, пакет №2 кріпиться в отвір для гвинта кронштейна.

Блок 4. Розрахунок сумарного дисбалансу та суми моментів інерції деталей.

Блок 5. Розрахунок радіус-вектору для площини корекції з даного набору мас.

Блок 6. Перевірка відповідності отриманого значення довжини радіус-вектора заданому діапазону пересування (80 мм). У випадку відповідності – перехід до Блоку 7. У випадку невідповідності – на Блок 3 для повторного розрахунку.

Блок 7. Обчислення дисбалансу для площини корекції.

Блок 8. Розрахунок набору мас для урівноваження дисбалансу.



Рисунок 3.3 – Схема алгоритму роботи модуля розрахунку параметрів врівноваження ротора

3.5 Особливості програмної реалізації ПКАІР

ПКАІР – це Qt-проект, в якому окрім файлів вихідного коду на мові C++ присутній файл проекту з розширенням pro. З цього файлу викликом qmake (програма, яка інтерпретує файли проектів, що містять різні параметри) створюється make-файл. Цей make-файл містить в собі всі необхідні інструкції для створення готового виконуваного модуля.

У make-файлі міститься виклик метаоб'єктного компілятора (Meta Object Compiler, MOC) або препроцесор, для створення додаткового коду C++ і необхідних заголовочних файлів. У проекті також присутній файл форми з розширенням ui, для якого теж буде створений файл C++ за допомогою компілятора ресурсів RCC. Після цього всі вихідні файли компілюються C++ компілятором в файли об'єктного коду, які об'єднуються компоновщиком link в готовий виконуваний модуль.

Схема взаємодії компонентів ПКАІР представлена на рис 3.4.

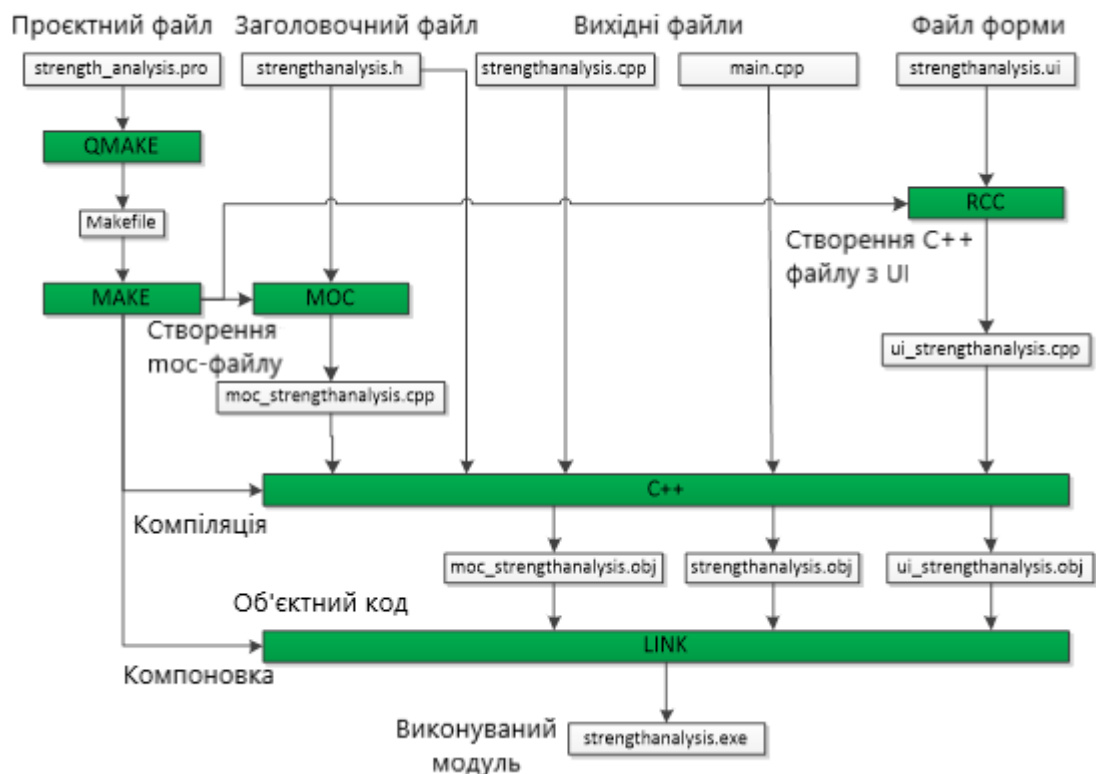


Рисунок 3.4 – Схема взаємодії компонентів ПКАІР

Розроблений модуль розрахунку на міцність містить в собі такі функціональні елементи: проєкт strengthanalysis, виконаний в середовищі розробки Qt, і клас MainWindow.

Опис змінних та методів класу міститься в таблиці 3.2 и 3.3.

Таблиця 3.2 – Опис глобальних змінних

Ідентифікатор	Тип	Коментар
mass	double	Вага пристосування
overload	double	Складові перевантаження
secutity	double	Коефіцієнт безпеки
inertPower	double	Інерційна сила
numbHoles	double	Кількість отворів
diamHoles	double	Діаметр отворів
thicknessMat	double	Товщина матеріалу
critArea	double	Площа критичного перетину
crush	double	Діюча напруга
ultimStrenght	double	Межа міцності
factorSafety	double	Запас міцності

Таблиця 3.3 – Методи класу MainWindow

Назва	Опис
bool writeFile(const QString & fileName)	Виконує запис даних в файл
void on_calculate_clicked()	Виконує рзрахунок
void on_close_clicked()	Вихід з програми
void on_setNull_clicked()	Обнулити рядки для вводу даних
bool saveAs()	Виведення діалогового вікна «Зберегти як»
void createActions()	Створення та налаштування дій
void createMenu()	Створення меню та додавання до нього дій

Модуль розрахунку на міцність складається з наступних файлів:

- `strengthanalysis.pro` – кореневий файл, в якому знаходяться всі записи про інші файли, що входять в проєкт і бібліотеки;
- `strengthanalysis.h` – заголовочний файл з кодом, який визначає та описує клас `MainWindow`;
- `strengthanalysis.cpp` – вихідний файл, з'єднаний з заголовним файлом `strengthanalysis.h`. Цей вихідний файл містить код, який описує виконання програми, тобто містить основну частину коду, що виконується безпосередньо під час запуску програми;
- `main.cpp` – вихідний файл, який містить точку входу в програму – функцію `main ()`. У ньому створюється об'єкт класу `QApplication`, об'єкт класу `MainWindow` (створюється головне вікно) і відображається створене вікно. Після цього програма переходить до нескінченного циклу обробки повідомлень операційної системи викликом `return a.exec ()`;
- `strengthanalysis.ui` – скриптовий файл, що описує форму цього застосунка.

Найбільш важливі частини коду знаходяться в наступних файлах: `strengthanalysis.h` і `strengthanalysis.cpp`. Так в файлі `strengthanalysis.h` вказується підключення необхідних бібліотек, виконується опис: використовуваного класу, конструктора, слотів, відповідних покажчиків і функцій. В таблиці 3.4 наведено призначення програмних компонентів файлу `strengthanalysis.h`.

Сигнали та слоти – це фундаментальний механізм Qt, що дозволяє пов'язувати об'єкти один з одним. Зв'язаним об'єктам немає необхідності «знати» що-небудь один про одного. Сигнали і слоти набагато зручніше механізму функцій зворотного виклику (callbacks) і чітко вписуються в концепцію об'єктно-орієнтованого програмування. Для використання механізму сигналів і слотів опис класу повинен містити спеціальний макрос `Q_OBJECT` (метаінформація для МОС компілятора Qt) в наступному після ключового слова `class` рядку:

Таблиця 3.4 – Опис програмних компонентів файлу strengthanalysis.h

Компонент	Опис
#include <QMainWindow>	Бібліотека для створення вікна QMainWindow
#include <QLineEdit>	Бібліотека класу управління віджетом для редагування однострочного тексту
#include <QLabel>	Бібліотека класу віждета відображення тексту або зображення
#include <QComboBox>	Бібліотека класу віждета надає комбінування кнопок у спливаючому списку
class MainWindow : public QMainWindow	Опис класу MainWindow та наслідування його від класу QMainWindow
explicit MainWindow(QWidget *parent = 0)	Опис конструктора MainWindow()
~MainWindow()	Опис деструктора ~MainWindow()
Ui::MainWindow *ui	Показчик на клас MainWindow

```

class MainWindow : public QMainWindow {
    Q_OBJECT
public:
    ...
};

```

Після макросу Q_OBJECT крапка з комою не ставиться. Перед виконанням компіляції Meta Object Compiler (MOC) аналізує всі класи та автоматично вбудовує в них всю необхідну інформацію. За виконання дій при натисканні відповідають слоти, описані в табл. 3.5. Для файлу strengthanalysis.cpp опис програмних компонентів, зокрема деяких бібліотек, наведено в табл. 3.6.

Таблиця 3.5 – Опис слотів

Слот	Опис
void on_calculate_clicked();	Слот, який відповідає за клік по кнопці «Розрахувати»
void on_close_clicked();	Слот, який відповідає за клік по кнопці «Вихід»

Продовження табл. 3.5

Слот	Опис
void on_setNull_clicked();	Слот, який відповідає за клік по кнопці «Скидання»
bool saveAs();	Слот, який відповідає за вибір пункту меню «Зберегти як»

Таблиця 3.6 – Опис програмних компонентів файлу strengthanalysis. cpp

Компонент	Опис
#include «mainwindow.h»	Підключення файлу mainwindow.h
#include «ui_mainwindow.h»	Підключення файлу форми
#include <QtWidgets>	Загальна бібліотека класу управління віджетами
#include <QTextStream>	Бібліотека класу, представляє зручний інтерфейс для запису тексту в потік і читання з нього
#include <QFile>	Бібліотека класу, представляє інтерфейс для читання з і записи в файл
#include <iostream>	Бібліотека введення виведення

Для виконання розрахунків використовується метод on_calculate_clicked (). Опис функцій наведено в табл. 3.7.

Таблиця 3.7 – Опис функцій

Функція	Опис
on_close_clicked ();	Функція виходу з програми
close ();	Стандартна функція Qt, що закриває вікно застосунка
on_setNull_clicked ()	Функція, що обнуляє всі введені значення
createActions ()	Функція, що створює дії, які потім зв'язуються з конкретною функцією
createMenu ()	Функція створення меню «File»
saveAs ()	Функція, що відповідає за дію «Зберегти як»
writeFile ()	Функція запису результатів розрахунку

До модуля розрахунку параметрів динамічного балансування ротора входять такі основні файли:

- `Coord.h` – заголовний файл з кодом, який визначає та описує клас **Coord** для виконання дій з координатами;
- `Coord.cpp` – вихідний файл, з'єднаний з заголовним файлом `Coord.h`. Цей вихідний файл містить код, який описує виконання програми, тобто містить основну частину коду, що виконується безпосередньо під час запуску програми розрахунку координат балансуючих мас;
- `Balans.h` – заголовний файл з кодом, який визначає та описує клас `Balans` для виконання дій з розрахунку мас для балансування ротора;
- `Balans.cpp` – вихідний файл, з'єднаний з заголовним файлом `Balans`. Цей вихідний файл містить код, який описує виконання програми, тобто містить основну частину коду, що виконується безпосередньо під час запуску програми розрахунку мас для балансування ротора;
- `main.cpp` – вихідний файл, який містить точку входу в програму – функцію `main ()`.

Особливості програмної реалізації модулю розрахунку ротора у вигляді коментарів показані в тексті програми в Додатку А. 1. Текст програми модулю розрахунку на міцність наведено в Додатку А. 2.

3.6 Тестування ПКАІР

Тестування програми має на увазі перевірку коректності роботи програми, а також виявлення різних помилок.

В першу чергу необхідно було протестувати програму на достовірність інформації, яка відображається. Мало ймовірно, що використання цієї програми може спровокувати глобальну помилку, тому що більшість операцій реалізовані стандартними методами середовища розробки.

Результати функціонального тестування програми наведені в таблиці 3.8.

Таблиця 3.8 – Функціональне тестування програми

Тестована функція	Очікуваний результат	Отриманий результат	Висновок
Запуск ПЗ	Відображення головного вікна програми	Головне вікно програми візуалізується	ПЗ запускається без помилок
Введення вихідних даних	Вихідні дані введені і відображаються на екрані	Вихідні дані відображені на екрані	Введення даних відбувається без помилок
Натискання кнопки «Розрахувати» і виконання розрахунку	Буде виконаний розрахунок з відображенням отриманих даних	Отримані дані відображені на екрані	Розрахунок виконується без помилок
Натискання кнопки «Скидання» і обнулення значень	Всі дані будуть скинуті, комірки введення даних будуть порожніми	Всі дані скинуті, комірки введення даних порожні	Кнопка «Скидання» працює без помилок
Натискання кнопки «Вихід» і закриття програми	Програма закриється	Програма закрита	Кнопка «Вихід» працює без помилок
Вибір в меню «File» дії «Save As ...»	З'явиться діалогове вікно "Save report"	Діалогове вікно "Save report"	Функція «Save As ...» працює коректно
Натискання в діалоговому вікні "Save report" кнопки «Зберегти»	Виконається збереження даних в файл	Дані збережено	Натискання в діалоговому вікні "Save report" кнопки «Зберегти» виконується без помилок
Вибір в меню «File» дії «Exit»	Програма закриється	Програма закрита	Функція «Exit» працює коректно

Порівняння очікуваних і отриманих результатів дозволяє зробити висновок про правильну роботу розробленої програми. Оперативність і достовірність отриманої інформації дозволяють зробити висновок про правильність обробки та візуалізації даних.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Призначення та умови виконання програми

ПКАІР призначений для автоматизації інженерних розрахунків параметрів технологічного обладнання при механічній обробці корпусних деталей.

Для роботи програми необхідний IBM PC-сумісний комп'ютер не нижче наступної конфігурації:

- процесор с тактовою частотою 1,6 ГГц або вище;
- 1 Гб та більше оперативної пам'яті;
- вільний дисковий простір не менше 10 Гб;
- монітор;
- клавіатура;
- маніпулятор «миша».

На ПК має бути встановлено наступне ПЗ:

- операційна система WINDOWS 7 та вище;
- Qt Creator 2.8.1.

4.2 Характеристики програми

Програма розроблена в середовищі програмування Qt Creator 2.8.1.

Для візуалізації вхідних та вихідних даних застосовується графічний інтерфейс, а також консоль програми (рис. 4.1).

Проект складається з наступних файлів:

- strengthanalysis.pro – кореневий файл, в якому знаходяться всі записи про інші файли, що входять в проєкт і бібліотеки;
- strengthanalysis.h – заголовний файл з кодом, який визначає та описує клас MainWindow;

```

C:\Qt\Qt5.1.1\Tools\QtCreator\bin\qtcreator_process_stub.exe
Плотность стали деталей конструкции 7700 кг/м^3
Вес шайбы из пакета пластин №1 : 58.7 гр.
Вес шайбы из пакета пластин №2 : 27.13 гр.
-----
Введите количество элементов конструкции : 2
Введите расстояние между плоскостями корекции (в миллиметрах): 60
Введите координаты центра масс детали
По оси X (в миллиметрах): 12
По оси Y (в миллиметрах): 15
Введите расстояние от точки отсчёта до центра масс детали
по оси Z (в миллиметрах): 10
Введите массу неуравновешенного тела (в граммах): 20
Введите координаты центра масс детали
По оси X (в миллиметрах): 12
По оси Y (в миллиметрах): 60
Введите расстояние от точки отсчёта до центра масс детали
по оси Z (в миллиметрах): 40
Введите массу неуравновешенного тела (в граммах): 55
Радиус-вектор дисбаланса ( 240 , 300 )
-----
Радиус-вектор дисбаланса ( 660 , 3300 )
-----
Количество пластин из пакета № 1 для плоскости 1 : 0
Количество пластин из пакета № 2 для плоскости 1 : 0
Радиус-вектор для плоскости 2 от оси вращения : 85 мм
Количество пластин из пакета № 1 для плоскости 2 : 0
Количество пластин из пакета № 2 для плоскости 2 : 1
Для продолжения нажмите любую клавишу . . .

```

Рисунок4.1 – Консоль програми

– strengthanalysis.cpp – вихідний файл, з'єднаний з заголовним файлом strengthanalysis.h. Цей вихідний файл містить код, який описує виконання програми, тобто містить основну частину коду, що виконується безпосередньо під час запуску програми;

– main.cpp – вихідний файл, який містить точку входу в програму – функцію main (). У ньому створюється об'єкт класу QApplication, об'єкт класу MainWindow (створюється головне вікно) і відображається створене вікно. Після цього програма переходить до нескінченного циклу обробки повідомлень операційної системи викликом return a.exec ();

– strengthanalysis.ui – скриптовий файл, що описує форму цього застосунка;

– Coord.h – заголовний файл з кодом, який визначає та описує клас **Coord** клас для виконання дій з координатами;

- `Coord.cpp` – вихідний файл, з'єднаний з заголовним файлом `Coord.h`. Цей вихідний файл містить код, який описує виконання програми, тобто містить основну частину коду, що виконується безпосередньо під час запуску програми розрахунку координат балансуючих мас;
- `Balans.h` – заголовний файл з кодом, який визначає та описує клас `Balans` для виконання дій з розрахунку мас для балансування ротора;
- `Balans.cpp` – вихідний файл, з'єднаний з заголовним файлом `Balans.h`. Цей вихідний файл містить код, який описує виконання програми, тобто містить основну частину коду, що виконується безпосередньо під час запуску програми розрахунку мас для балансування ротора.

4.3 Звернення до програми

Запуск ПКАІР здійснюється запуском файлу «Rotor.exe». Після запуску має з'явитися головне вікно програми.

4.4 Вхідні та вихідні дані

Вхідними даними для модулю розрахунку параметрів ротора є:

- кількість елементів конструкції;
- відстань між площинами корекції, мм;
- координати по вісям x та y центру мас деталі, мм;
- відстань від точки відліку до центра мас деталі, мм;
- маса неурівноваженого тіла, г.

Вихідними даними для модулю розрахунку параметрів ротора є:

- радіус-вектор сумарного дисбалансу, мм;
- радіус-вектор для площини корекції, мм;
- радіус-вектор для площини корекції від вісі обертання, мм;
- кількість пластин з пакету 1 та пакету 2 для кожної площини корекції.

Вхідними даними для модулю розрахунку на міцність деталі пристосування є:

- вага деталі пристосування, кг;
- складові перевантаження;
- коефіцієнт безпеки;
- кількість отворів кріплення, шт;
- діаметр отвору, мм;
- товщина матеріалу, мм.

Вихідними даними для модулю розрахунку на міцність деталі пристосування є:

- інерційна сила, кгс;
- площа критичного перетину матеріалу деталі, мм²;
- діюче напруження зминання, кгс/ мм²;
- межа міцності матеріалу деталі, кгс/ мм²;
- запас міцності конструкції, кгс/ мм².

4.6 Повідомлення

ПКАІР має декілька видів повідомлень, пов'язаних з перевіркою відповідності результатів розрахунку заданим експлуатаційним параметрам.

Перше повідомлення – на етапі розрахунку показників міцності деталі пристосування для кріплення заготовки деталі. Якщо розрахований показник менше заданого значення, програма повідомить про це та запропонує перевизначити спосіб навантаження.

Друге повідомлення – на етапі розрахунку показників балансування ротору. Якщо розрахована довжина радіус-вектора перевищує заданий діапазон пересування (80 мм), тоді треба задати іншу частоту обертання ротора або корегувати параметри набору пластин для формування корегувальних мас.

5 КЕРІВНИЦТВО КОРИСТУВАЧА

5.1 Призначення програми

ПКАІР призначений для виконання розрахунків на міцність вузлів кріплення обладнання та динамічного врівноваження ротора верстата при обробці корпусних деталей.

Використання ПКАІР дозволяє прискорити процес розрахунків та скоротити загальний термін підготовки технологічної документації.

Функції ПКАІР:

- введення вхідних даних в зручному для користувача вигляді;
- розрахунок показників міцності деталей пристосувань для кріплення заготовки деталі на верстаті з ЧПУ;
- розрахунок мас і просторового положення противаг для динамічного врівноваження ротора верстата;
- візуалізацію результатів розрахунків;
- збереження в файл для подальшої передачі в систему автоматизації технологічної підготовки виробництва.

5.2 Умови виконання програми

Для роботи програми необхідний IBM PC-сумісний комп'ютер не нижче наступної конфігурації:

- процесор с тактовою частотою 1,6 ГГц або вище;
- 1Гб та більше оперативної пам'яті;
- вільний дисковий простір не менше 10Гб;
- монітор;
- клавіатура;
- маніпулятор «миша».

На ПК має бути встановлено наступне ПЗ:

- операційна система WINDOWS 7 та вище;

- середовище програмування Qt Creator 2.8.1.

5.3 Виконання програми

Для того щоб виконати розрахунок показників міцності, потрібно спочатку ввести вихідні дані (рис. 5.1):

- вага деталі пристосування, кг;
- складові перевантаження;
- коефіцієнт безпеки;
- кількість отворів кріплення, шт;
- діаметр отвору, мм;
- товщина матеріалу, мм;

При натисканні клавіші «Tab» на клавіатурі відбувається послідовне переміщення курсору по комірках введення даних, що зручно для навігації.

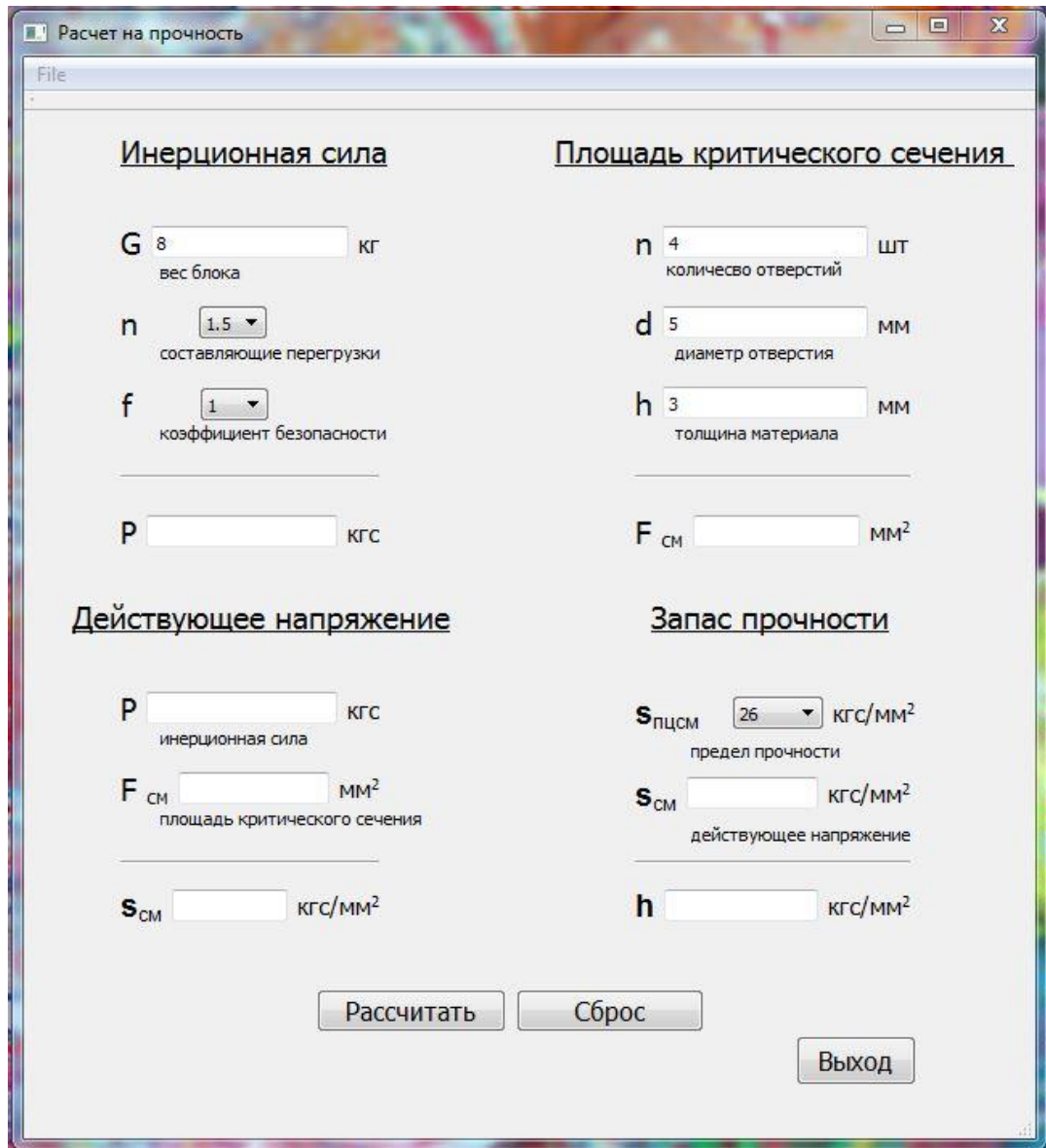
Далі, для проведення розрахунків, натиснути кнопку «Розрахувати». Дані розрахунків будуть показані у відповідних комірках:

- інерційна сила, кгс;
- площа критичного перетину матеріалу деталі, мм²;
- діюче напруження зминання, кгс/ мм²;
- межа міцності матеріалу деталі, кгс/ мм²;
- запас міцності конструкції, кгс/ мм².

Для збереження даних розрахунків, вибираємо у верхній частині програми меню «File / Save As ...» (рис. 5.2).

З'являється діалогове вікно «Save Report», в якому за замовчуванням звіт зберігається в форматі .txt. Також можна зберегти в будь-якому іншому текстовому форматі, ввівши розширення вручну.

Для проведення розрахунку параметрів балансування ротору потрібно за послідовними запитами програми ввести таку інформацію:



Инерционная сила

G 8 кг
вес блока

n 1.5
составляющие перегрузки

f 1
коэффициент безопасности

P кгс

Площадь критического сечения

n 4 шт
количество отверстий

d 5 мм
диаметр отверстия

h 3 мм
толщина материала

F_{cm} мм²

Действующее напряжение

P кгс
инерционная сила

F_{cm} мм²
площадь критического сечения

s_{cm} кгс/мм²

Запас прочности

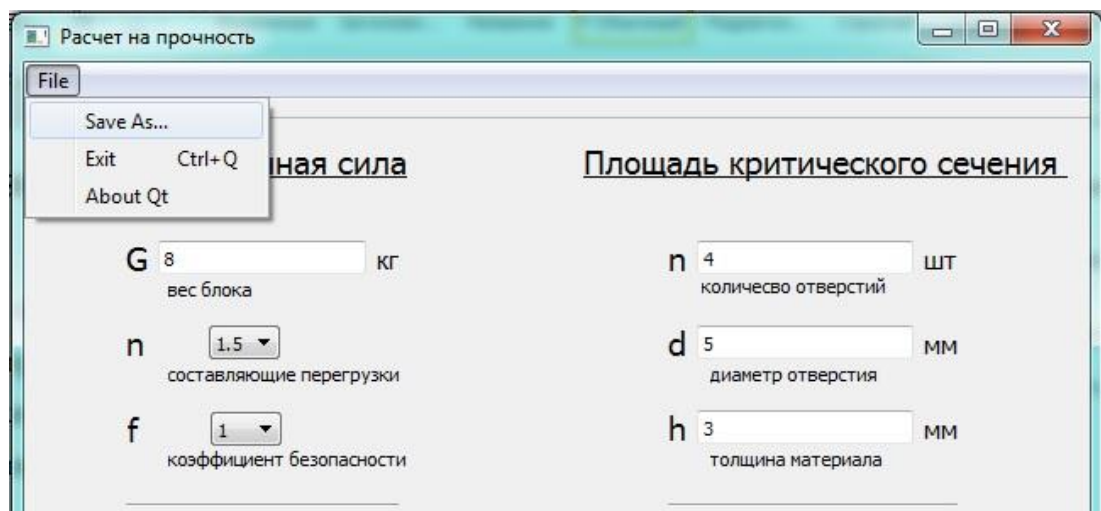
$s_{лцсм}$ 26 кгс/мм²
предел прочности

$s_{см}$ кгс/мм²
действующее напряжение

h кгс/мм²

Рассчитать Сброс Выход

Рисунок 5.1 – Диалогове вікно програми



File

- Save As...
- Exit Ctrl+Q
- About Qt

Инерционная сила

G 8 кг
вес блока

n 1.5
составляющие перегрузки

f 1
коэффициент безопасности

Площадь критического сечения

n 4 шт
количество отверстий

d 5 мм
диаметр отверстия

h 3 мм
толщина материала

Рисунок 5.2 – Меню «File/Save As...»

- кількість елементів конструкції;
- відстань між площинами корекції, мм;
- координати по вісям x та y центру мас деталі, мм;
- відстань від точки відліку до центра мас деталі, мм;
- маса неурівноваженого тіла, г.

Особливості вхідних параметрів для розрахунку балансу ротора:

- координати центра мас кожної деталі повинні бути вказані в міліметрах відносно осі обертання;
- вага кожної неурівноваженої маси повинна бути вказана в грамах;
- відстань між площинами корекції повинна бути вказана у міліметрах;
- всі деталі виготовлені із сталі з щільністю 7700 кг / м^3 ;
- в якості противаг для балансування використовуються пакети пластин пакет № 1 з N шайб розмірами $\varnothing 100 \times 1 \text{ мм}$ і центральним отвором $\varnothing 16 \text{ мм}$, і пакет № 2 з M шайб розмірами $\varnothing 50 \times 2 \text{ мм}$ (кількість N і M визначається залежно від розрахованих дисбалансів);
- пакет шайб № 1 кріпиться гвинтами в паз планшайби та може переміщатися уздовж цього паза в межах $f = 80 \text{ мм}$, пакет №2 кріпиться в отвір для гвинта кронштейна.

Після введення вихідних даних програма відображає у консолі необхідні для балансування ротора дані.

Ці дані можуть бути передані у систему ЧПУ верстата для балансування ротора.

ВИСНОВКИ

В ході виконання дипломної роботи виконано дослідження процесу автоматизованої технологічної підготовки виробництва корпусних деталей на верстатах з ЧПУ.

Дослідження показали, що на точність оброблюваних поверхонь суттєво впливають налаштування обладнання. Тому, потрібно не лише забезпечити точні переміщення робочих органів верстата в процесі обробки, але винятково точно узгодити положення заготовки та інструмента в системі координат верстата. Також виявлено, що при дисбалансі ротору вал не тільки передає зайве зусилля, але й дестабілізує роботу системи. Тому, застосування системи противаг для динамічного врівноваження ротора дозволяє усунути вібрацію, завдяки чому підвищується точність обробки.

Використання сучасних САЕ-систем для інженерних розрахунків параметрів налаштування технологічного обладнання, обмежене через високу вартість та складність інтерфейсу програм, що базуються здебільшого на МСЕ.

Розроблений ПКАІР може застосовуватися для автоматизації розрахунків показників міцності деталей пристосувань для кріплення заготовки корпусної деталі, а також параметрів динамічного балансування ротору верстата.

Розробку ПЗ виконано на мові C++ в середовищі Qt Creator 2.8.1. Функціональне тестування підтвердило працездатність розробленого ПЗ.

Використання ПКАІР дозволяє автоматизувати та відповідно прискорити процес інженерних розрахунків технологічних параметрів, за рахунок чого скоротити загальний термін технологічної підготовки виробництва.

В подальшій роботі розроблений ПКАІР може бути інтегрований з існуючою САМ-системою підприємства.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Энциклопедия по машиностроению XXL [Электрон. ресурс]. – Режим доступа: <https://mash-xxl.info/info/601800/>.
2. Изготовление и обработка корпусных деталей [Электрон. ресурс]. – Режим доступа: <https://reduktor.org/mekhanicheskaya-obrabotka-na-zakaz/322-izgotovlenie-i-obrabotka-korpusnyh-detalej>.
3. Типовая технология изготовления корпусных деталей [Электрон. ресурс]. – Режим доступа: <https://poznayka.org/s45577t2.html>.
4. Ковалевский, С. В. Технология обработки типовых деталей и сборки машин: конспект лекций / С. В. Ковалевский, С. Г. Онищук, Ю. Б. Борисенко. – Краматорск: ДГМА, 2015. – 119 с.
5. Способы повышения эффективности растачивания основных отверстий корпусных деталей [Электрон. ресурс]. – Режим доступа: <http://surl.li/sftb>.
6. Балансировка вращающихся тел. Термины: ГОСТ 19534-74. – [Чинний від 1974-02-25]. – М.: Издательство стандартов, 1974. – 49 с.
7. Машины и технологическое оборудование. Система классов точности балансировки. Основные положения: ГОСТ 22061—76. – [Чинний від 1977-07-01]. – М.: Издательство стандартов, 1974. – 135 с.
8. Писаренко Г.С. Справочник по сопротивлению материалов / Г.С. Писаренко, А.П. Яковлев, В.В. Матвеев. – К.: Наукова думка, 1988 – 550 с.
9. Нержавеющие стали и сплавы коррозионно-стойкие, жаростойкие и жаропрочные: ГОСТ 5632-2014. – [Чинний від 2015-01-01]. – М.: Стандартиформ 2015. – 112 с.
10. Загорулько, А. В. Сучасні обчислювальні комплекси : конспект лекцій / А. В. Загорулько, Д. О. Кайота. – Суми : Сумський державний університет, 2019. – 48 с.

11. Ansys. Система конечно-элементного (МКЭ) анализа [Электрон. ресурс]. – Режим доступа: <http://surl.li/sftc>.
12. Встречайте новую версию программного обеспечения: Ansys 2020 R2 [Электрон. ресурс]. – Режим доступа: <https://www.ansys.soften.com.ua/about-ansys/novosti/570-ansys-2020-r2-overview.html>.
13. ANSYS CFD – универсальная программная система конечно-элементного анализа задач гидрогазодинамики [Электрон. ресурс]. – Режим доступа: https://www.techgidravlika.ru/view_post.php?id=60.
14. SIMULIA: Моделирование продуктов, природы и жизни [Электрон. ресурс]. – Режим доступа: https://tesis.com.ru/cae_brands/simulia/.
15. ABAQUS UNIFIED FEA. Комплексные решения для реалистичного моделирования [Электрон. ресурс]. – Режим доступа: <https://www.3ds.com/ru/produkty-i-uslugi/simulia/produkty/abaqus/>.
16. HyperWorks – лидер в области CAE-систем [Электрон. ресурс]. – Режим доступа: <http://www.hyperworks.compmechlab.ru/>.
17. Meshing Gets a Makeover with Next-Generation HyperWorks [Electronic resource]. – Access mode: <https://www.engineering.com/story/meshing-gets-a-makeover-with-next-generation-hyperworks>.
18. Новости компьютерного инжиниринга. NX Nastran компании Siemens PLM Software теперь доступен в облачной среде Rescale [Электрон. ресурс]. – Режим доступа: <https://fea.ru/news/5917>.
19. Creo Simulate [Электрон. ресурс]. – Режим доступа: <https://pts-russia.com/products/creo/creo-modules/creo-simulate.html>.
20. Бирбраер, Р. Как обеспечить надежную работу изделия с помощью Pro/ENGINEER Mechanica [Электрон. ресурс] / Р. Бирбраер, И. Радченко, В. Жарков. – Режим доступа: <https://sapr.ru/article/19530>.

21. Pro/MECHANICA Online Books [Электрон. ресурс]. – Режим доступа:

http://sual.narod.ru/ProE/ProM/Us_w_ProE/Introduction/Introduction.htm.

22. C++ [Электрон. ресурс]. – Режим доступа:
<http://progopedia.ru/language/c-plus-plus/>.

23. Введение в C++. Язык программирования C++ [Электронный ресурс] – Режим доступа: <https://metanit.com/cpp/tutorial/1.1.php>.

24. ТОП-7 популярных IDE для программирования на C++ [Электронный ресурс] – Режим доступа:
<https://itvdn.com/ru/blog/article/cplspls-top7>.

25. The Qt project [Electronic resource]. – Access mode: <https://qt-project-org.herokuapp.com/>.

26. Программирование на языке C++ в среде Qt Creator /
Е. Р. Алексеев, Г. Г. Злобин, Д. А. Костюк и др. – М. : ALT Linux, 2015. –
448 с.

ДОДАТОК А
Текст програми

A.1 Файли модуля розрахунку параметрів ротора

A 1.1 Код файлу Balans.h

```
#ifndef BALANS_H
#define BALANS_H

#include <iostream>
#include <locale.h>
#include "Coord.h"

class Balans
{
public:
    Coord crd;          // Координати центру маси неурівноваженого тіла
    float mass;         // Маса неурівноваженого тіла
    float l;            // Відстань від точки відліку до центра маси деталі
    Coord d;            // Дисбаланс
    Coord m;            // Момент

    Balans(){mass,l = 0; addCoord(); addLenMass(); calcDisbMom(); }
    void addCoord();
    void addLenMass();
    void calcDisbMom();
    void showDisbVect();
    Balans& operator+(const Balans& righth);
};

#endif // BALANS_H
```

A.1.2 Код файлу Coord.h

```
#ifndef COORD_H
#define COORD_H

#include <math.h>

// Створюємо клас для виконання дій з координатами
class Coord
{
protected:
    float x;
    float y;
public:
    Coord();
    Coord(float X, float Y);
    float Mod();
    float ShowX();          // Повертає значення x
    float ShowY();          // Повертає значення y
};
```

```

void SetX(float X);           // Задаємо значення x
void SetY(float Y);           // Задаємо значення y
void SetXY(float X, float Y); // Задаємо значення x и y
Coord& operator=(const Coord& right); // Привласнює об'єкту значення x та y
правого об'єкта
bool operator==(const Coord& right); // Повертає true, якщо x та y обох
об'єктів рівні
Coord& operator-(const float right); // Віднімає y кожної координати задане
число
Coord& operator-(const Coord& right); // Віднімає об'єктів класу
Coord& operator+(const float right); // Додає до кожної координати задане
число
Coord& operator+(const Coord& right); // Сума об'єктів класу
Coord operator*(const float right); // Множить координати на число
Coord operator/(const float right); // Ділить координати на число
};
#endif // COORD_H

```

A.1.3 Код файлу Balans.cpp

```

#include "Balans.h"

using namespace std;

void Balans::addCoord()
{
    float tmp = 0;
    wcout<<L"Введіть координати центра мас деталі "<<endl;
    wcout<<L"По вісі X (в міліметрах): ";
    cin>>tmp;
    crd.SetX(tmp);
    tmp = 0;
    wcout<<L"По вісі Y (в міліметрах): ";
    cin>>tmp;
    crd.SetY(tmp);
    cout<<endl;
}

void Balans::addLenMass()
{
    wcout<<L" Введіть відстань від точки відліку до центру маси деталі \nпо вісі Z
(в міліметрах): ";
    cin>>l;
    cout<<endl;
    wcout<<L"Введіть масу неурівноваженого тіла (в грамах): ";
    cin>>mass;
}

void Balans::calcDisbMom()
{
    d = crd * mass;
}

```

```

    m = d * l;
}

Balans& Balans::operator +(const Balans& right)
{
    d = d + right.d;
    m = m + right.m;
    return *this;
}

void Balans::showDisbVect()
{
    wcout<<L"Радіус-вектор дисбалансу ( "<<d.ShowX()<<" , "<<d.ShowY()<<"
)<<endl;
    wcout<<"-----"<<endl;
}

```

A.1.4 Код файлу Coord.cpp

```

#include "Coord.h"

Coord::Coord()
{
    x = 0;
    y = 0;
}

Coord::Coord(float X, float Y)
{
    x=X;
    y=Y;
}

float Coord::Mod()
{
    float tmp;
    tmp = sqrt((x*x) + (y*y));
    return tmp;
}

bool Coord::operator==(const Coord& right)
{
    if(x==right.x && y==right.y)
        return true;
    return false;
}

float Coord::ShowX()
{
    return x;
}

float Coord::ShowY()
{

```

```

    return y;
}
void Coord::SetX(float X)
{
    x=X;
}
void Coord::SetY(float Y)
{
    y=Y;
}
void Coord::SetXY(float X, float Y)
{
    x=X;
    y=Y;
}
Coord& Coord::operator-(const float right)
{
    x=x-right;
    y=y-right;
    return *this;
}
Coord& Coord::operator-(const Coord& right)
{
    x=x-right.x;
    y=y-right.y;
    return *this;
}
Coord& Coord::operator+(const float right)
{
    x=x+right;
    y=y+right;
    return *this;
}
Coord& Coord::operator+(const Coord& right)
{
    x=x+right.x;
    y=y+right.y;
    return *this;
}
Coord Coord::operator*(const float right)
{
    Coord tmp;
    tmp.x=x*right;
    tmp.y=y*right;
    return tmp;
}
Coord Coord::operator/(const float right)
{
    Coord tmp;
    tmp.x=x/right;
    tmp.y=y/right;
    return tmp;
}

```

```

}
Coord& Coord::operator=(const Coord& right)
{
    if (this == &right)
        return *this;
    x = right.x;
    y = right.y;
    return *this;
}

```

A.1. 5 Код файлу Main.cpp

```

#include <QCoreApplication>
#include "Balans.h"

const float n1mass = 58.7;
const float n2mass = 27.13;

int main(int argc, char *argv[])
{
    QCoreApplication a(argc, argv);

    setlocale(LC_ALL, "Rus");
    using namespace std;
    wcout<<L"Щільність сталі деталей конструкції 7700 кг/м^3\n";
    wcout<<L"Вага шайби з пакета пластин №1 : "<<n1mass<<L" гр.\n";
    wcout<<L"Вага шайби з пакета пластин №2 : "<<n2mass<<L" гр.\n";
    wcout<<"-----"<<endl;

    int n = 0;
    float len = 0;
    wcout<<L"Введіть кількість елементів конструкції : ";
    cin>>n;
    cout<<endl;
    wcout<<L"Введіть відстань між площинами корекції (в міліметрах): ";
    cin>>len;
    cout<<endl;
    Balans* b = new Balans[n];
    Coord sdisb;
    Coord smom;
    for (int i = 0; i < n; i++)
    {
        b[i].showDisbVect();
        sdisb = sdisb + b[i].d;
        smom = smom + b[i].m;
    }
    Coord momk2 = smom * -1;
    Coord disbk2 = momk2 / len;

    int p1mass, p2mass = 0;

```

```

int ost1m,ost2m = 999;
int rad = 160;
int cres1,cres2,radres1,radres2 = 0;
while(rad != 80)
{
    Coord tmpmax = disbk2 / rad;
    float m = tmpmax.Mod();
    int count1 = m / (int)n1mass;
    int ost1 = (int)m % (int)n1mass;
    int count2 = m / (int)n2mass;
    int ost2 = (int)m % (int)n2mass;
    if(ost1 < ost1m)
    {
        cres1 = count1;
        radres1 = rad;
        ost1m = ost1;
    }
    else if(ost2 < ost2m)
    {
        cres2 = count2;
        radres2 = rad;
        ost2m = ost2;
    }
    rad--;
}
if(ost1m < ost2m)
{
    rad = radres1;
    p1mass = cres1;
    p2mass = 0;
}
else
{
    rad = radres2;
    p2mass = cres2;
    p1mass = 0;
}
Coord disbk1 = (sdisb + disbk2) * -1;
Coord tmpmassk1 = disbk1 / 80;
float massk1 = tmpmassk1.Mod();
int ok1,ok2,massp1,massp2;
ok1 = ok2 = massp1 = massp2 = 0;
ok1 = (int)massk1 % (int)n1mass;
ok2 = (int)massk1 % (int)n1mass;
if(ok1 < ok2)
    massp1 = (int)massk1 / (int)n1mass;
else
    massp2 = (int)massk1 / (int)n2mass;

wcout<<"-----"<<endl;
wcout<<L"Кількість пластин з пакету № 1 для площини 1 : "<<massp1<<endl;
wcout<<L"Кількість пластин з пакету № 2 для площини 1 : "<<massp2<<endl;

```

```

wcout<<L"Радіус-вектор для площини 2 від вісі обертання : "<<rad<<L" мм\n";
wcout<<L" Кількість пластин з пакету № 1 для площини 2 : "<<p1mass<<endl;
wcout<<L" Кількість пластин з пакету № 2 для площини 2 : "<<p2mass<<endl;
system("pause");
return 0;
}

```

A.2 Файли модуля розрахунку параметрів міцності

A.2. 1 Код файлу Strength_analisis.pro

```

#-----
#
# Project created by QtCreator 2021-04-06T15:25:30
#
#-----

QT      += core gui

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

TARGET = strength_analysis
TEMPLATE = app

SOURCES += main.cpp\
           strengthanalysis.cpp

HEADERS  += strengthanalysis.h

FORMS    += strengthanalysis.ui

```

A.2. 2 Код файлу Strengthanalysis.h

```

#ifndef STRENGTHANALYSIS_H
#define STRENGTHANALYSIS_H
#include <QMainWindow>
#include <QLineEdit>
#include <QLabel>
#include <QComboBox>

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{

```

Q_OBJECT

public:

```
explicit MainWindow(QWidget *parent = 0);
~MainWindow();
bool writeFile(const QString & fileName);
```

private slots:

```
void on_calculate_clicked();
void on_close_clicked();
void on_setNull_clicked();
bool saveAs();
```

private:

```
void createActions();
void createMenu();
Ui::MainWindow *ui;
QLineEdit *massEdit;
QLineEdit *powerEdit;
QLineEdit *inertPowerEdit;
QLineEdit *numbHolesEdit;
QLineEdit *diamHolesEdit;
QLineEdit *thicknessMatEdit;
QLineEdit *critAreaEdit;
QLineEdit *crushEdit;
QComboBox *overloadComboBox;
QComboBox *securityComboBox;
QComboBox *ultimStrenghtComboBox;
QAction *saveAsAction;
QAction *exitAction;
QAction *aboutQtAction;
QMenu *fileMenu;
double mass, overload, security, inertPower,
      numbHoles, diamHoles, thicknessMat, critArea,
      crush,
      ultimStrenght, factorSafety;
```

};

#endif // STRENGTHANALYSIS_H

A.2. 3 Код файла Main.cpp

```
#include "strengthanalysis.h"
#include <QApplication>
```

```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
```

```

MainWindow w;
w.show();

return a.exec();
}

```

A.2. 4 Код файлу strengthanalysis.cpp

```

#include "strengthanalysis.h"
#include "ui_strengthanalysis.h"
#include <QLineEdit>
#include <QLabel>
#include <QtWidgets>
#include <QComboBox>
#include <QMainWindow>
#include <QTextStream>
#include <QFile>
#include <iostream>

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow) // Опис процесу реалізації класу і конструктора з
    параметром і виділення пам'яті
    {
        ui->setupUi(this); // Вказівник на setupUi

        setWindowTitle(tr("Расчет на прочность"));
        //setWindowIcon();

        overloadComboBox = new QComboBox; // Тут і нижче, створення віджетів
        ComboBox
        ui->overloadComboBox->addItem(QString::number(-2));
        ui->overloadComboBox->addItem(QString::number(1.5));
        ui->overloadComboBox->addItem(QString::number(2));
        ui->overloadComboBox->addItem(QString::number(4));
        ui->overloadComboBox->addItem(QString::number(9));

        securityComboBox = new QComboBox;
        ui->securityComboBox->addItem(QString::number(1));
        ui->securityComboBox->addItem(QString::number(1.5));

        ultimStrenghtComboBox = new QComboBox;
        ui->ultimStrenghtComboBox->addItem(QString::number(13.2));
        ui->ultimStrenghtComboBox->addItem(QString::number(26));
        ui->ultimStrenghtComboBox->addItem(QString::number(27));

        createActions();
        createMenu();
    }

```

```

}

MainWindow::~MainWindow() // Реалізація деструктора MainWindow
{
    delete ui; // Видалення виділеної пам'яті
}

void MainWindow::on_calculate_clicked() // Метод для виконання розрахунків
{
    mass = ui->massEdit->text().toDouble(); // Привласнюємо об'єкту mass значення,
яке вводиться користувачем
    overload = ui->overloadComboBox->currentText().toDouble();
    security = ui->securityComboBox->currentText().toDouble();
    inertPower = (mass*overload*security); // Виконується множення введених
значень
    ui->inertPowerEdit->setText(QString::number(inertPower)); // Привласнюємо
результат розрахунку об'єкта inertPowerEdit, який виводить цей результат на екран

    numbHoles = ui->numbHolesEdit->text().toDouble();
    diamHoles = ui->diamHolesEdit->text().toDouble();
    thicknessMat = ui->thicknessMatEdit->text().toDouble();
    critArea = (numbHoles*diamHoles*thicknessMat);
    ui->critAreaEdit->setText(QString::number(critArea));
    ui->inertPowerEdit_2->setText(QString::number(inertPower));
    ui->critAreaEdit_2->setText(QString::number(critArea));
    crush = (inertPower/critArea);
    ui->crushEdit->setText(QString::number(crush));
    ultimStrenght = ui->ultimStrenghtComboBox->currentText().toDouble();
    ui->crushEdit_2->setText(QString::number(crush));
    factorSafety = (ultimStrenght/crush);
    ui->factorSafetyEdit->setText(QString::number(factorSafety));
}

void MainWindow::on_close_clicked() // Функція виходу з програми
{
    close(); // Стадартна функція, що закриває вікно застосунку
}

void MainWindow::on_setNull_clicked()// Функція, що обнуляє всі введені значення
{
    ui->massEdit->setText(""); // Тут і далі, привласнює рядку з даними пусте
значення;
    ui->inertPowerEdit->setText("");
    ui->numbHolesEdit->setText("");
    ui->diamHolesEdit->setText("");
    ui->thicknessMatEdit->setText("");
    ui->critAreaEdit->setText("");
    ui->inertPowerEdit_2->setText("");
    ui->critAreaEdit_2->setText("");
    ui->crushEdit->setText("");
    ui->crushEdit_2->setText("");
    ui->factorSafetyEdit->setText("");
}

```

```

void MainWindow::createActions()// Функція, що створює дії, які потім зв'язуються
з конкретною функцією
{
    saveAsAction = new QAction(tr("&Save As..."), this); // Створюємо новий об'єкт
типу QAction і присвоюємо назву, яке буде показуватися в меню
    saveAsAction->setIcon(QIcon("")); // Привласнюємо дії іконку
    saveAsAction->setShortcut(QKeySequence::SaveAs); // Привласнюємо дії «гарячу
клавішу»
    connect (saveAsAction, SIGNAL(triggered()), this, SLOT(saveAs())); // З'єднуємо
сигнал вибору дії в меню, зі слотом, що відповідає за цю дію
    exitAction = new QAction(tr("&Exit"), this);
    exitAction->setIcon(QIcon(""));
    exitAction->setShortcut(tr("Ctrl+Q"));
    connect (exitAction, SIGNAL(triggered()), this, SLOT(close()));
    aboutQtAction = new QAction(tr("About &Qt"), this);
    connect (aboutQtAction, SIGNAL(triggered()), qApp, SLOT(aboutQt()));
}
void MainWindow::createMenu() // Створюємо меню «File» за допомогою функції
createMenu ()
{
    fileMenu = menuBar()->addMenu(tr("&File")); // Створюємо меню, привласнюємо
назву
    fileMenu->addAction(saveAsAction); // Додаємо дії
    fileMenu->addAction(exitAction);
    fileMenu->addAction(aboutQtAction);
}
bool MainWindow::writeFile(const QString & fileName) // Функція, що виконує
запис результатів розрахунку
{
    QFile data(fileName);
    if (data.open(QFile::WriteOnly | QFile::Truncate)) // Якщо діалогове вікно
повертає логічне «так»
    {
        QTextStream out(&data); // Виконується запис даних
        out << "Inertial force P = " << inertPower << endl;
        out << "Throat area F = " << critArea << endl;
        out << "Actuating force = " << crush << endl;
        out << "Margin of safety = " << factorSafety << endl;
        return true;
    }
}
bool MainWindow::saveAs() // Функція, що відповідає за дію «зберегти як»
{
    QString fileName = QFileDialog::getSaveFileName(this,
tr("Save report"),
tr("Reports files.txt")); // Створюємо діалогове вікно
"Save report"
    return writeFile(fileName); // Функція, що повертає запис даних в потік за
допомогою функції writeFile ()
}

```

ДОДАТОК Б
Слайди презентації

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Дипломна робота
на тему:
**Розробка програмного забезпечення
для інженерних розрахунків ротора**

Виконав:
ст. гр. КНТ – 217

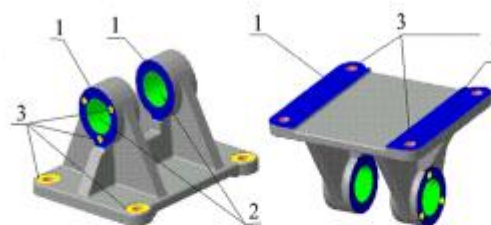
Е.Е. Ситвелієв

Керівник:
к.т.н., доцент

А.В. Пархоменко

Рисунок В.1 – Слайд 1

АКТУАЛЬНІСТЬ РОБОТИ



Поверхні спряження (1),
основні (2) та
додаткові (3) отвори
корпусної деталі



Обробка корпусної деталі на верстаті

Рисунок В.2 – Слайд 2

МЕТА ТА ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

Мета дипломної роботи - розробка програмного забезпечення для автоматизації інженерних розрахунків параметрів технологічного обладнання для підвищення точності обробки корпусних деталей.

Задачі дипломної роботи:

- дослідити функціональні можливості існуючих систем автоматизації інженерних розрахунків;
- виконати проектування програмного комплексу автоматизації інженерних розрахунків (ПКАІР);
- виконати конструювання ПКАІР;
- практично застосувати ПКАІР для визначення параметрів міцності деталі пристосування та балансування ротору при обробці корпусних деталей;
- розробити програмну документацію.

3

Рисунок В.3 – Слайд 3

ПОРІВНЯЛЬНИЙ АНАЛІЗ САЕ-СИСТЕМ

Функції	Pro/ Mechanica	Creo Simulation	Creo Advanced Simulation
Статичний структурний аналіз	+	+	+
Симуляція роботи деталей та складань	+	+	+
Аналіз критичного навантаження	-	+	+
Аналіз твердотільних елементів	-	+	+
Оптимізація конструкцій	-	+	+
Нелінійність великих деформацій	-	-	+
Динамічний аналіз	-	-	+

4

Рисунок В.4 – Слайд 4

ВИКОРИСТАННЯ ПАКЕТУ PRO/MECHANICA



5

Рисунок В.5 – Слайд 5

ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ ПКАІР

- введення вхідних даних в зручному для користувача вигляді;
- розрахунок показників міцності деталей пристосувань для кріплення заготовки деталі на верстаті з ЧПУ;
- розрахунок мас і просторового положення протитяг для динамічного врівноваження ротора верстата;
- візуалізація результатів розрахунків;
- збереження в файл для подальшої передачі в систему автоматизації технологічної підготовки виробництва.

6

Рисунок В.6 – Слайд 6

УЗАГАЛЬНЕНА СТРУКТУРА ПКАІР



7

Рисунок В.7 – Слайд 7

ВИБІР ІНСТРУМЕНТАРІЮ РОЗРОБКИ

	Visual Studio	Eclipse	Qt Creator
Кількість підтримуваних компіляторів	1	3	7
Кросплатформність	-	+	+
Безкоштовна версія	-	+	+
Невимогливість до апаратних ресурсів	-	+	+
Зручний редактор коду	+	-	+

8

Рисунок В.8 – Слайд 8

РОЗРОБКА АЛГОРИТМІВ ФУНКЦІОНУВАННЯ



Модуль розрахунку на міцність



Модуль розрахунку параметрів врівноваження ротора

Рисунок В.9 – Слайд 9

СХЕМА ВЗАЄМОДІЇ КОМПОНЕНТІВ ПКІАР

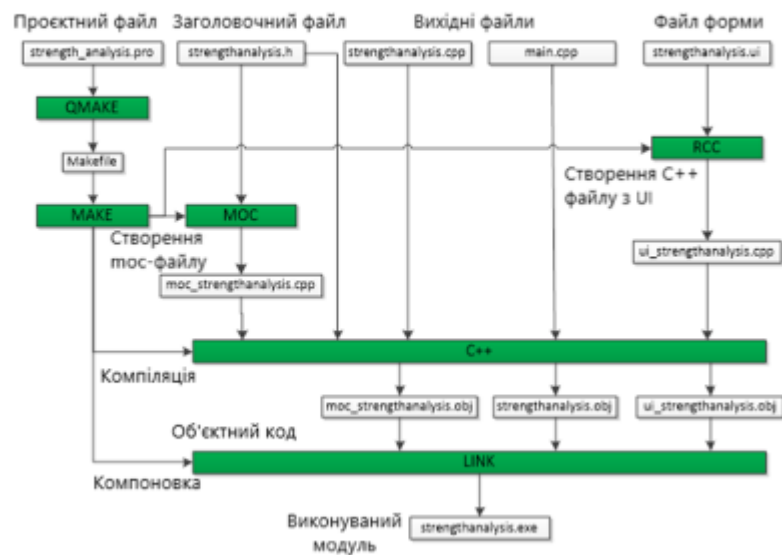
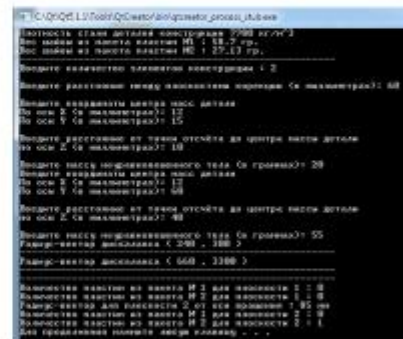


Рисунок В.10 – Слайд 10

ІНТЕРФЕЙС ПРОГРАМИ



Діалогове вікно програми



Консоль програми

9

Рисунок В.11 – Слайд 11

ВИСНОВКИ

- Розроблений ПКАІР може застосовуватися для автоматизації розрахунків показників міцності деталей пристосувань для кріплення заготовки корпусної деталі, а також параметрів динамічного балансування ротору верстата.
- Розробку виконано на мові C++ в середовищі Qt Creator 2.8.1. Функціональне тестування підтвердило працездатність розробленого ПЗ.
- Використання ПКАІР дозволяє автоматизувати та відповідно прискорити процес інженерних розрахунків технологічних параметрів, за рахунок чого скоротити загальний термін технологічної підготовки виробництва.
- В подальшій роботі розроблений ПКАІР може бути інтегрований з існуючою САМ-системою підприємства.

12

Рисунок В.12 – Слайд 12



**ДЯКУЮ ЗА
УВАГУ!**

Рисунок В.13 – Слайд 13