

МАГІСТЕРСЬКА РОБОТА

МР.АКПМ-29.00.00.000 ПЗ

Група АКПМ-24-1

Назарій Шеремета

2025

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу
Факультет автоматизації та енергетики
Кафедра автоматизації та комп'ютерно-інтегрованих технологій

Шеремета Назарій Степанович

(прізвище, ім'я, по батькові)

УДК 681.5
(індекс)

МАГІСТЕРСЬКА РОБОТА

Автоматизоване визначення та ідентифікація реальних приведених

(назва роботи)

характеристик відцентрових нагнітачів

Автоматизація та комп'ютерно-інтегровані технології

(назва освітньої програми)

174 - Автоматизація, комп'ютерно-інтегровані технології та робототехніка

(шифр і назва спеціальності)

Консультант з нормоконтролю

асистент

(посада)

Л.І. Лагойда

(ініціали та прізвище)

Здобувач освітнього ступеня

АКПм-24-1

(шифр групи)

Н.С. Шеремета

(ініціали та прізвище)

Науковий керівник

доцент, к.т.н.

(науковий ступінь, вчене звання) (підпис) (дата) (ініціали та прізвище)

А.І. Лагойда

(ініціали та прізвище)

Допущено до захисту

Завідувач кафедри

доцент, к.т.н.

(посада)

(підпис)

(дата)

А.І. Лагойда

(ініціали та прізвище)

Рецензент

доцент, к.т.н.

(посада)

(підпис)

(дата)

І.І. Чигур

(ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківський національний технічний університет нафти і газу

(повне найменування закладу вищої освіти)

Факультет автоматизації та енергетики

Кафедра автоматизації та комп'ютерно-інтегрованих технологій

Освітній рівень другий (магістерський)

Спеціальність 174 - Автоматизація, комп'ютерно-інтегровані технології та

(шифр і назва)

робототехніка

ЗАТВЕРДЖУЮ

Завідувач кафедри АКІТ

(А.І. Лагойда)

« » _____ 20 року

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТОВІ**

Шеремета Назарій Степанович

(прізвище, ім'я, по батькові)

1. Тема роботи Автоматизоване визначення та ідентифікація реальних
приведених характеристик відцентрових нагнітачів

керівник роботи Лагойда Андрій Іванович, к.т.н., доцент

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

затверджені наказом закладу вищої освіти від «10» листопада 2025 року № 699/7

2. Строк подання студентом роботи 08.12.2025

3. Вихідні дані до роботи Матеріали практик, науково-технічні журнали,
довідники

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Вступ. 1. Обладнання а також призначення компресорних станцій.

2. Визначення безрозмірних характеристик відцентрових нагнітачів.

3. Автоматизація процесу визначення поточних характеристик відцентрового нагнітача. 4. Визначення характеристик відцентрових нагнітачів. Загальні висновки.

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

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 30.10.2024

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Термін виконання етапів роботи	Примітка
1	Обладнання а також призначення компресорних станцій	02.12.2025	
2	Визначення безрозмірних характеристик відцентрових нагнітачів	03.12.2025	
3	Автоматизація процесу визначення поточних характеристик відцентрового нагнітача	04.12.2025	
4	Визначення характеристик відцентрових нагнітачів	05.12.2025	

Студент _____
(підпис)

Шеремета Н.С.
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Лагойда А.І.
(прізвище та ініціали)

АНОТАЦІЯ

Магістерська робота на здобуття кваліфікації магістра за спеціальністю 174 - «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» та освітньою програмою - «Автоматизація та комп'ютерно-інтегровані технології». - Івано-Франківський національний технічний університет нафти і газу. - Івано-Франківськ, 2025.

Об'єктом дослідження в даній магістерській роботі є процес ідентифікації та автоматизованого визначення реальних приведених характеристик відцентрових нагнітачів, який слугує основою для подальшої модернізації програмного забезпечення підсистеми антипомпажного регулювання та захисту. У роботі розглянуто методики та принципи ідентифікації характеристик відцентрових нагнітачів, а також розроблено програмні засоби для їх визначення у середовищі MATLAB, що забезпечує можливість відстеження змін приведених характеристик газоперекачувального агрегату в реальному часі. На підставі отриманих результатів формується оцінка технічного стану нагнітача та газотурбінного двигуна, а також визначається доцільність і напрямки коригування програмного забезпечення підсистеми антипомпажного регулювання і захисту газоперекачувального агрегату.

Ключові терміни: ідентифікація, приведені характеристики, відцентровий нагнітач, газоперекачувальний агрегат.

ANNOTATION

Master's thesis for obtaining a master's degree in specialty 174 - "Automation, computer-integrated technologies and robotics" and educational program - "Automation and computer-integrated technologies". - Ivano-Frankivsk National Technical University of Oil and Gas. - Ivano-Frankivsk, 2025.

The object of research in this master's thesis is the process of identification and automated determination of real reduced characteristics of centrifugal superchargers, which serves as the basis for further modernization of the software of the anti-surge control and protection subsystem. The paper considers the methods and principles of identifying the characteristics of centrifugal superchargers, and also develops software tools for their determination in the MATLAB environment, which provides the ability to track changes in the reduced characteristics of the gas pumping unit in real time. Based on the results obtained, an assessment of the technical condition of the supercharger and gas turbine engine is formed, and the feasibility and directions of software adjustment of the anti-surge control and protection subsystem of the gas pumping unit are determined.

Key terms: identification, reduced characteristics, centrifugal supercharger, gas pumping unit.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ.....	8
ВСТУП.....	9
РОЗДІЛ 1	
ОБЛАДНАННЯ А ТАКОЖ ПРИЗНАЧЕННЯ КОМПРЕСОРНИХ СТАНЦІЙ.....	12
1.1. Характеристика компресорних станцій.....	12
1.2. Характеристика нагнітача природного газу.....	13
1.3. Структура компресора.....	15
1.4. Загальна характеристика нагнітача.....	21
Висновки до розділу 1.....	24
РОЗДІЛ 2	
ВИЗНАЧЕННЯ БЕЗРОЗМІРНИХ ХАРАКТЕРИСТИК ВІДЦЕНТРОВИХ НАГНІТАЧІВ.....	25
2.1. Розрахунок витрати для нагнітачів різних типів (280, 370 і 520) за перепадом тиску на всмоктуючому конфузори.....	25
2.2. Визначення приведених характеристик $Q_{пр}$ і $E_{пр}$	28
2.3 Проведення розрахунків і побудова графіків фактичної приведеної характеристики нагнітача у середовищі Mathcad.....	31
Висновки до розділу 2.....	35
РОЗДІЛ 3	
АВТОМАТИЗАЦІЯ ПРОЦЕСУ ВИЗНАЧЕННЯ ПОТОЧНИХ ХАРАКТЕРИСТИК ВІДЦЕНТРОВОГО НАГНІТАЧА.....	36
3.1. Алгоритм візуальної побудови кривих.....	36
3.2. Методика визначення приведених характеристик.....	38
Висновки до розділу 3.....	49

РОЗДІЛ 4

ВИЗНАЧЕННЯ	ХАРАКТЕРИСТИК	ВІДЦЕНТРОВИХ	
НАГНІТАЧІВ.....			50
4.1. Розпізнавання, оцінка характеристик та визначення оптимальних значень функціоналів якості.....			50
4.2 Оцінка та порівняння методів ідентифікації об'єктів.....			53
4.3. Ідентифікація та параметричне оцінювання режимів функціонування відцентрових нагнітачів у складі газоперекачувальних агрегатів.....			56
Висновки до розділу 4.....			64
ЗАГАЛЬНІ ВИСНОВКИ.....			65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....			67
ДОДАТКИ.....			69

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ ТА ТЕРМІНІВ**

ПСГ - підземне сховище газу.

ВН - відцентровий нагнітач.

ГТУ - газотурбінна установка.

АСР - автоматизована система регулювання.

ГПА - газоперекачувальний агрегат.

КС - компресорна станція.

КЦ - компресорний цех.

МГ - магістральний газопровід.

ПВН - повно напірний.

ТК - турбокомпресор.

ОЗП - оперативно-запам'ятовуючий пристрій.

СЦКУ - система централізованого контролю та управління.

НВН - неповно напірний.

ВСТУП

Актуальність теми. Газотранспортна інфраструктура України являє собою розгалужену систему магістральних газопроводів, що забезпечує транспортування природного газу внутрішнім споживачам і транзит його до країн Західної Європи. Переміщення газу по сталевих трубопроводах здійснюється за допомогою потужних газоперекачувальних агрегатів, розміщених на компресорних станціях уздовж траси газопроводів.

Ефективність функціонування єдиної системи газопостачання значною мірою визначається стабільністю та раціональністю режимів роботи компресорних станцій, які водночас є об'єктами високої енергоємності. Режим їх експлуатації суттєво впливає на загальні енергетичні витрати, необхідні для транспортування природного газу.

Для підвищення ефективності управління робочими режимами компресорних станцій та їх оптимізації необхідно мати достовірну інформацію щодо діапазонів допустимих режимів функціонування, меж енерговитрат, а також фактичного технічного стану лінійної частини та обладнання, який визначається на основі сучасних методів технічної діагностики.

Різкі коливання обсягів споживання природного газу призводять до виникнення нестійких режимів його руху в магістральних трубопроводах. Такі перехідні процеси, зумовлені зміною густини газового потоку, можуть тривати від кількох годин до доби, супроводжуючись суттєвими коливаннями тиску. Це, у свою чергу, порушує стабільність експлуатації газопровідної системи та в окремих випадках може стати причиною її аварійного руйнування.

Під час розрахунку робочих режимів компресорних станцій (КС) магістральних газопроводів (МГ) та визначення навантаження газоперекачувальних агрегатів (ГПА) зазвичай використовують приведені характеристики відцентрових нагнітачів (ВН), отримані в результаті стендових випробувань за стандартних умов - при сталих фізичних параметрах газу та номінальній швидкості обертання ротора. Однак у процесі експлуатації параметри нагнітачів поступово змінюються, що призводить до зростання похибки при

використанні таких характеристик. Це негативно впливає на точність розрахунку запасу до помпажу, знижує достовірність прийняття оперативних рішень та ефективність роботи протипомпажних систем.

Таким чином, розроблення системи автоматизованого визначення актуальних приведених характеристик відцентрових нагнітачів для типових газоперекачувальних агрегатів є науково та практично обґрунтованою і вкрай актуальною задачею сучасної газотранспортної галузі.

Мета і завдання дослідження. Метою даної роботи є створення методики ідентифікації та автоматизованого визначення актуальних витратно-напірних характеристик відцентрових нагнітачів, що слугуватиме базою для подальшої модернізації програмного забезпечення підсистеми антипомпажного регулювання та захисту.

Для досягнення поставленої мети необхідно вирішити такі науково-технічні завдання:

- визначити основні параметри приведених характеристик $Q_{пр}$ та $E_{пр}$;
- побудувати реальні криві приведених характеристик відцентрового нагнітача на основі отриманих експериментальних даних;
- здійснити ідентифікацію параметрів приведеної характеристики з метою формування корекційного сигналу, призначеного для вдосконалення роботи системи антипомпажного регулювання.

Об'єктом дослідження є відцентровий нагнітач, який є типовим елементом конструкції газоперекачувальних агрегатів.

Предметом дослідження є приведені витратно-напірні характеристики відцентрових нагнітачів та підходи, що можуть бути застосовані для їхньої ідентифікації.

Методи дослідження. Дослідження здійснюється із застосуванням методів теорії автоматичного керування, системної ідентифікації та комп'ютерного імітаційного моделювання.

Новизна отриманих результатів є такою:

- Криві приведених характеристик центробіжних нагнітачів сформовані в режимі реального часу.

- Створено систему автоматизованої ідентифікації приведених характеристик центробіжних нагнітачів.

Зв'язок роботи з науковими програмами, планами, темами, грантами.

Тематика магістерської роботи відповідає вимогам Закону України "Про пріоритетні напрямки розвитку науки і техніки", зокрема, статті 7, пункту 6, що стосується "Нових технологій та ресурсозберігаючих технологій в енергетиці, промисловості та агропромисловому комплексі". Автор брав активну участь у виконанні держбюджетної науково-дослідної роботи кафедри Автоматизації та комп'ютерно-інтегрованих технологій Івано-Франківського національного технічного університету нафти і газу (ІФНТУНГ) у межах основного напрямку наукової школи "Автоматизоване управління технологічними процесами на основі методів нечіткої логіки і штучних нейромереж" (сертифікат № 3 від 2007 р.). Наукову підставу для роботи становлять результати, отримані під час виконання науково-дослідних робіт за державними темами, зокрема, "Синтез комп'ютерних систем та розробка програмного забезпечення для об'єктів нафтогазового комплексу" та "Науково-організаційні засади нарощування видобутку вітчизняних нафти і газу та диверсифікації постачання енергетичних ресурсів для підвищення енергетичної безпеки України" (ДР № 011U605890 та ДР № 0115U007099 відповідно).

Практичне значення отриманих результатів. На підставі визначених приведених характеристик здійснюється оцінка технічного стану двигуна та нагнітача, а також приймається рішення щодо потреби внесення коректив у програмне забезпечення підсистеми антипомпажного регулювання та захисту газоперекачувального агрегату.

Структура та обсяг магістерської роботи. Магістерська робота складається із вступу, чотирьох розділів, висновків, списку використаних джерел та 5 додатків. Повний обсяг магістерської роботи становить 99 сторінок: обсяг основного тексту - 66 сторінок, 31 рисунок, 4 таблиці; список використаних джерел, що включає 16 найменувань та займає 2 сторінки; 5 додатків на 31 сторінці.

РОЗДІЛ 1

ОБЛАДНАННЯ А ТАКОЖ ПРИЗНАЧЕННЯ КОМПРЕСОРНИХ СТАНЦІЙ

1.1. Характеристика компресорних станцій

Компресорні станції мають за завдання підвищення енергії газу та його транспортування через лінійні ділянки. Процес підвищення тиску газу відбувається у компресорному цеху (КЦ), який є основою компресорної станції (КС).

Склад компресорної станції включає в себе:

а) технологічні системи, включаючи компресію та охолодження газу, очищення газу, охолодження води або антифризу, охолодження масла, підготовку паливного та пускового газу, подачу імпульсного газу, газу для власних потреб і забезпечення повітропостачання;

б) приміщення для зберігання пально-мастильних матеріалів (ПММ), метанолу, обладнання, матеріалів і реагентів, а також арматури та трубопроводів.в) адміністративно-побутові приміщення;

г) удосконалені системи електропостачання, захисту від блискавки, теплопостачання, утилізації тепла, телефонного зв'язку, опалення а також вентиляції, виробничо-господарського а також пожежного водопостачання і відповідно каналізації, контролю а також керування, радіофікації, годинникової системи, пожежної та охоронної сигналізації, а також автоматичного пожежогасіння;

д) допоміжні об'єкти та приміщення;

є) технологічні комунікації із запірною арматурою, а також об'єкти, такі як ремонтно-механічні майстерні, лабораторія-майстерня КВПіА.

Компресорна станція (КС) виступає в ролі управлінського елемента в цілісній системі дальнього транспортування газу. Шляхом налаштування параметрів роботи КС визначається режим всієї газотранспортної системи. Загальновідомо, що газоперекачувальні агрегати розглядаються як ключове

обладнання КС. Магістральні газопроводи зазвичай оснащені відцентровими нагнітачами, які включають в себе електроприводи а також газотурбінні установки (ГТУ), і відповідно поршневі газомотокомпресорні установки.

Вибір приводу для нагнітача на КС здійснюється враховуючи потужність, яку може споживати нагнітач та можливу швидкість обертання ротора газоперекачувального агрегата (ГПА) [1, 16].

Порівняно з електроприводом, газотурбінний привід володіє численними перевагами, зокрема можливість регулювання режимів роботи нагнітача шляхом зміни швидкості обертання ротора, а це можливо незалежно від джерела енергії. Однак, слід відзначити, що недоліками цього приводу є громіздкість обладнання, можливі великі витрати газу під час пуску агрегату та його можливий вплив на екологію.

Використання електродвигунів для приводу відцентрового нагнітача (ВН) на компресорних станціях зараз є досить рідкісним явищем (див. табл. 1.1). В сучасний період використовуються блоки, що змінюють частоту струму, що, в свою чергу, дозволяє регулювати оберти вала [1, 16].

1.2. Характеристика нагнітача природного газу

У технічних джерелах в Україні термін "нагнітачі" використовується для опису лопаткових компресорних машин, які створюють відношення тиску більше 1,1 та не включають систем охолодження робочого середовища під час стиснення. У зарубіжній технічній літературі, особливо в англійських джерелах, такі машини частіше визначають як компресори або, іноді, як бустери.

Нагнітачі, які приводяться в рух газотурбінними установками або електродвигунами, зазвичай мають відцентровий принцип дії. Осьові машини вимагають вищих обертів, і вони є більш чутливими до забруднень робочого середовища. Це призводить до того, що їх рідко використовують для стиснення природного газу на компресорних станціях. У більшості випадків для лінійних компресорних станцій використовують одно- або двохступінчасті нагнітачі. Три- і

чотириступінчасті нагнітачі застосовуються при закачуванні газу в системи підземного зберігання газу, обробці газу та на збірних пунктах промисловості за низького тиску в пласті.

Таблиця 1.1

Технологічна характеристика відцентрового нагнітача

Тип ВН	Робочий тиск, МПа	Номінальні оберти ротора, об/хв	Ступінь стиску	Продуктивність комерційна, млн м ³ /добу	Параметри режиму при працюючих ВН					
					Продуктивність об'ємного потоку при умовах входу, м ³	Продуктивність об'ємного потоку при умовах входу,	Номінальний тиск на вході, МПа	Продуктивність об'ємного потоку при умовах входу, м ³	Номінальний тиск на вході, МПа	Стандартний тиск на вході, МПа
280-11-2	-	7700	-	13,0	178,0	210,0	4,65	227,0	3,950	3,60
260-13-2	5,60	5550	1,250	14,0	204,0	245,0	4,50	286,0	3,750	3,200
370-15-2	3,80	5550	1,250	13,0	303,0	363,0	3,06	421,0	2,550	2,200
370-17-1	5,60	5300	1,250	19,80	289,0	346,0	4,50	398,0	3,750	3,250
НГ-280-9	5,60	5000	1,200	34,00	486,0	582,0	4,63	660,0	3,900	3,450
520-12-1	5,60	4800	1,270	29,30	425,0	530,0	4,40	620,0	3,540	3,000
260-13-1	7,350	4800	1,170	23,00	350,0	290,0	5,70	0,0	5,300	0,0
Н-16-56	5,60	4600	1,250	51,00	800,0	950,0	4,50	0,0	3,600	0,0
650-21-1	7,60	3700	1,450	53,00	640,0	0,0	5,20	0,0	0,0	0,0
CDP-224	7,520	6200	1,510	17,20	219,0	0,0	4,97	0,0	0,0	0,0
PCL802/24	7,520	6200	1,510	17,20	219,0	0,0	4,97	0,0	0,0	0,0
PCL1002/40	7,520	4600	1,510	45,0	572,0	0,0	4,97	0,0	0,0	0,0
С3044-30	5,60	15700	2,20	2,90	-	0,0	2,50	0,0	0,0	0,0
RF288-30	7,60	5000	1,50	21,80	274,0	0,0	5,10	0,0	0,0	0,0

Важливо розрізняти між проміжним і кінцевим ступенями нагнітача. Проміжна ступінь складається із робочого дифузора а також зворотного направляючого апарату, колеса, тоді як кінцевий ступінь, натомість, обладнаний збірною камерою або равликом замість зворотного направляючого апарату. Дифузор разом із збірною камерою часто отримує назву вихідного пристрою. У дослідницькій літературі структуру, що складається з робочого колеса та дифузора, часто називають двохланковим ступенем, а структуру, яка включає зворотний направляючий апарат або збірну камеру, - триланковим. Зворотний направляючий апарат іноді називають діафрагмою [2].

1.3. Структура компресора

Турбокомпресор (ТК) типу 650-21-2 призначений для стиснення та транспортування природного газу. Він може експлуатуватися як автономний агрегат або в паралельному режимі з іншими турбокомпресорами цього ж типу, проте ефективність спільної роботи обмежується робочими характеристиками кожного ТК.

Зовнішній корпус турбокомпресора виготовлено з чавуну у циліндричній формі. На передній частині встановлена кришка, закріплена за допомогою болтів та гайок. Всмоктуюча та нагнітальна горловини інтегровані з корпусом, а на їх кінцях змонтовані кільця, відповідні діаметрам з'єднувальних трубопроводів. У нижній частині корпусу передбачені монтажні п'ятки для кріплення на фундаменті, а на верхній — розташований резервуар для зберігання напірного масла.

На кришці передбачені отвори для подачі масла до контактної кутового сальника та його відведення, включно з направленням у сепараторну систему, а також для з'єднання з вирівнювальним поршнем всмоктуючої горловини ТК. Додатково на кришці розташований отвір для підключення імпульсного турбоприводу. На задньому торці корпусу виконані отвори для з'єднання внутрішніх компонентів компресора з корпусом за допомогою спеціальних болтів, при цьому герметичність забезпечується ущільнювальними кільцями.

Взаємозв'язані отвори утворюють кільцевий паз на задньому торці корпусу, який заповнюється високонапірним маслом для змащення сальників і запобігання можливому витоку газу з агрегату. Крім того, у задньому торці передбачені отвори для подачі та відведення високонапірного масла до контактної сальника, відведення газу разом з маслом у сепараторну систему, подачі газу до вхідної частини кінцевого сальника та підключення імпульсного трубопроводу.

Герметичність усіх отворів у точках їхнього контакту з отворами пакета забезпечується ущільнювальними кільцями.

Ротор компресора складається з двох робочих зварних коліс. Робочі колеса

однакові за конструкцією, вони відрізняються лише шириною лопаток. Лопатки врізані в опорний диск, а захисний диск зварений до лопаток. Робочі колеса монтується на вал зі зазором.

Врівноважувальний поршень, який призначений для зменшення осьової сили, також має туге кріплення на валу. На внутрішній поверхні врівноважувального поршня виконані пази, які становлять частину лабіринтного ущільнювача.

Опорний підшипник (рис. 1.1) призначений для компенсації неврівноважених осьових сил та забезпечення регулювання відносного положення ротора щодо статора. Він має двоскладовий корпус (1), всередині якого розташовані бронзові сегменти (2), а також вирівнювальний механізм, що забезпечує рівномірний розподіл навантаження на сегменти.

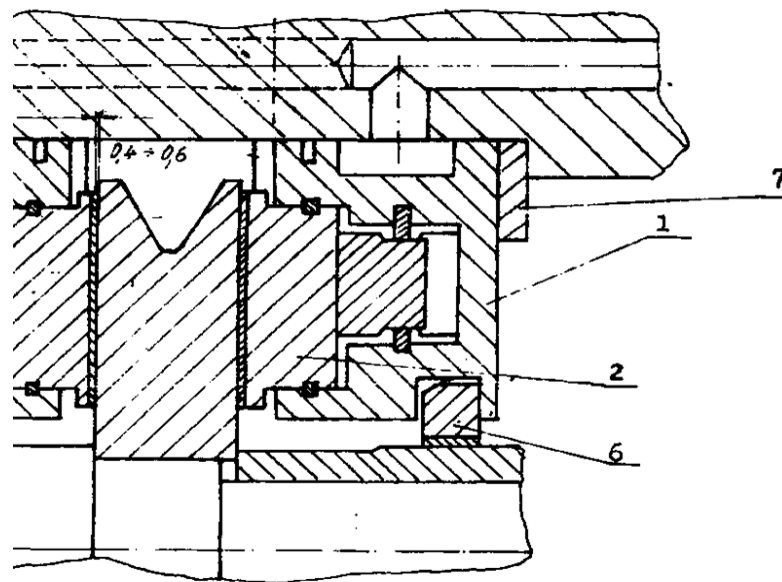


Рис. 1.1. Підшипник опорний

Вирівнюючий механізм (див. рис. 1.2) включає в себе ковзачі (3), стрижні (4) і роздільник (5). Роздільник складається з двох частин і встановлюється в жолоби двоскладного корпусу.

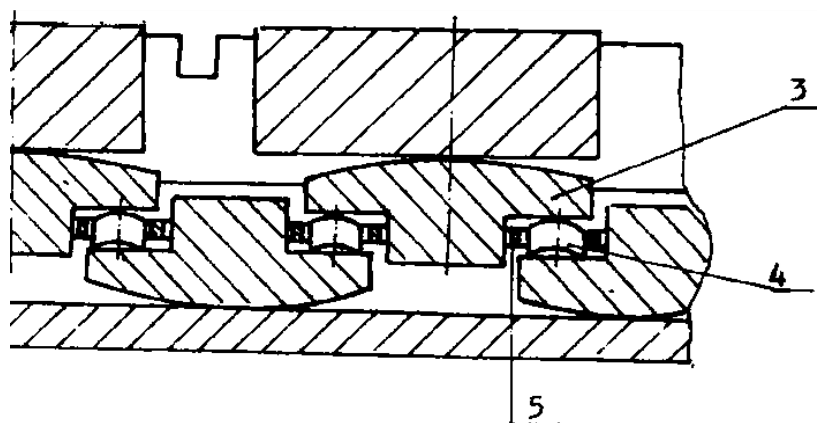


Рис. 1.2. Механізм вирівнювання

Ширина зазору між сегментами та диском опорного підшипника дорівнює 0,4–0,6 мм.

Для нагляду за коректною роботою опорного підшипника використовують термоелементи, які вмонтовані в два сегменти з обох боків опорного підшипника.

Ротор турбокомпресора має підшипникове кріплення на двох однакових радіальних підшипниках. Внутрішній компонент підшипника, складаючись з двох частин (1), розташований в корпусі підшипника (2) зі зазором в межах 0 - 0,08 мм. Радіальний зазор між вкладишем підшипника та цапфою ротора становить 0,3–0,376 мм.

Моніторинг стану підшипників у процесі експлуатації здійснюється за допомогою двох термоелементів, встановлених у нижній частині підшипника.

Ротор разом із підшипниками, контактними та лабіринтними ущільнювачами, внутрішнім корпусом і іншими компонентами утворює цілісний блок, що називається пакетом.

Внутрішній корпус турбокомпресора складається з всмоктуючої частини, дифузоров та нагнітальної секції, яка горизонтально розділена проміжною стінкою. У всмоктуючій частині передбачено канали для подачі масла до контактного ущільнювача та радіального підшипника, канали для відведення газу разом з маслом до сепараторної системи, а також отвори для відведення масла до регулятора. Додатково в цій частині корпусу є отвір для подачі газу у простір перед кінцевим лабіринтним ущільнювачем.

Розділова площа нижньої частини пакета обладнана двома пазами у всмоктуючій та нагнітальній секціях: один паз сполучений із простором подачі масла до контактної ущільнювача, інший — із простором відведення масло+газ, що забезпечує додатковий захист від можливого витоку газу через роздільну площину. У неробочому стані пакет перебуває у положенні рівноваги і закріплюється до корпусу вісьмома болтами через дно корпусу, герметичність з'єднання забезпечується ущільнювальними кільцями.

У нагнітальній частині передбачені отвори для подачі високонапірного масла до контактної ущільнювача та радіального підшипника, для подачі та відведення масла в опорний підшипник, для відведення масло+газу у систему ущільнювачів, а також для подачі високонапірного масла до регулятора [5].

Для зменшення втрат газу через протікання між ступенями, врівноважувачем поршнем та простором за контактним ущільнювачем пакет оснащено лабіринтними ущільнювачами з латунними кромками.

Основні параметри:

- Денний обсяг виробництва Турбокомпресора (ТК): 46 мільйонів кубометрів газу.

- Величина тиску у всмоктуючій горловині становить 5,18 МПа.

- Величина температури у всмоктуючій горловині становить 15 °С.

У рамках цього магістерського дослідження будуть застосовані дані відцентрового нагнітача типу 370-14-1, тому доцільно розглянути його конструктивні та експлуатаційні особливості більш детально.

Основна версія нагнітача цього типу оснащена приводом від газотурбінної установки ГТ-750-6 з потужністю 6,0 тисяч кВт. Однією із модифікацій цих агрегатів є 370-15-2, що представляє собою варіант нагнітача для дотискувальних станцій, який приводиться в рух газовою турбіною ГТК-5. Нагнітачі 260-13, що функціонують без редуктора у складі газотурбінної установки ГТК-5 на лінійних компресорних станціях, а також нагнітачі з вихідним тиском 76 кгс/см² і приводом від ГТК-10, розроблені на базі конструкції нагнітача 370-14-1.

В таблиці 1.2 представлені дані щодо геометрії робочих коліс та короткі

технічні відомості про стан модифікації нагнітача типу 370.

Таблиця 1.2

Особливості конструктивної форми робочих коліс нагнітача типу 370

Конструктивний тип робочого колеса	Кут виходу β_2 , градус	Значення ширини колеса на виході, мм	Значення діаметру робочого колеса D_2 , мм	Число лопаток z_2	Діаметр робочого колеса D_1 , мм	Кут виходу β_1 , градус
Колесо нагнітача типу 370 підвищеної напірності	45,0	43,0	845,0	20,0	360,0	32,0
Високонапірне колесо	90,0	43	845	24,0	360	35
Колесо типу 370	32	43	845,0	14	360	32°31'
Колесо для дожимної КС	70	43,0	845	24,0	360,0	32,0
Колесо типу 260	22,0	43	845,0	12,0	360	30,0

Регулювання режиму виконується шляхом зміни оборотів в межах від 3800,0 до 5500,0 обертів на хвилину. Серед вихідних умов враховується значення коефіцієнта стисливості газу $z=0,91$, температуру газу на вході $t_{вх}=15^{\circ}\text{C}$ та питому газову постійну $R=50 \frac{\text{кгс} \cdot \text{м}}{\text{кг} \cdot ^{\circ}\text{C}}$. Деякий номінальний режим функціонування нагнітача 370-14-1 наведено в таблиці 1.2.

На рисунку 1.3 представлено продовжений розріз нагнітача. Даний нагнітач є одноступінчастим агрегатом із консольно встановленим робочим колесом. Проточна частина оснащена лопатевим дифузором та кільцевою збірною камерою.

Ротор із підшипниками та іншими елементами ходової частини розташований у спеціальній гільзі, що встановлюється всередині сталевго корпуса з тангенціальними патрубками для подачі та відведення газу,

забезпечуючи горизонтальне співвісне розташування. Робочий диск із лопатками виготовлено шляхом фрезерування з однієї поковки, а покривний диск надійно прикріплений до нього заклепками. До торцевої стінки прикріплено корпус підшипників, в якому розміщені підшипники. Конструкція нагнітача дозволяє його зупинку під значенням повного тиску та короткочасну роботу для можливості опресовки газопроводу при значенні тиску газу на виході із агрегату 65 кгс/см^2 .

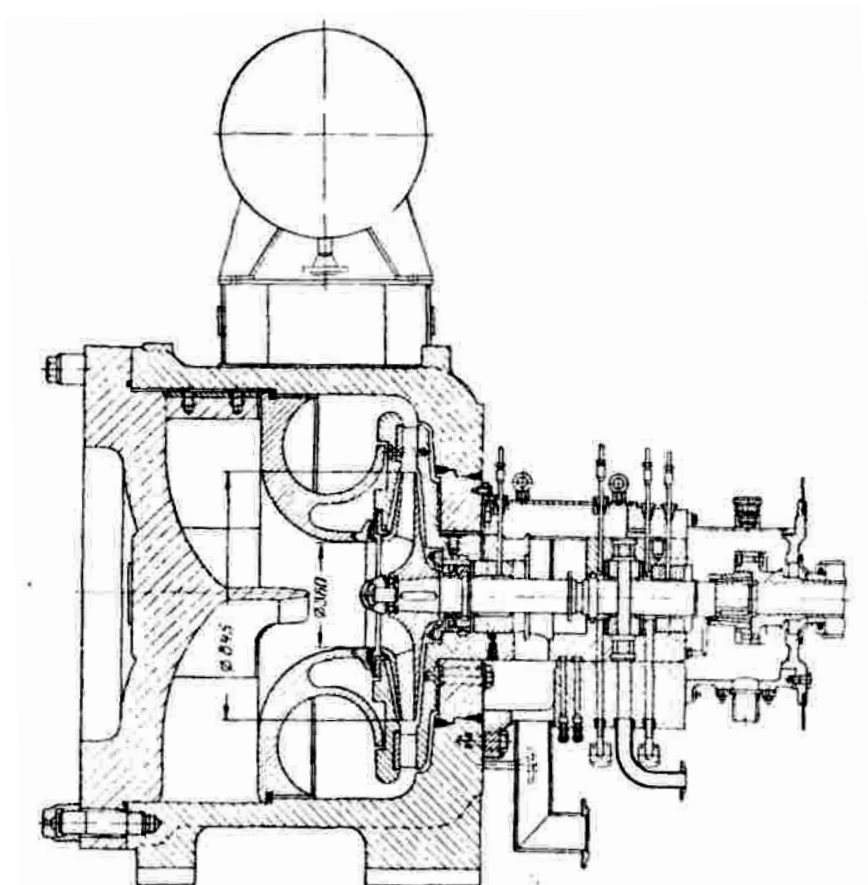


Рис. 1.3. Нагнітач типу 370-14-1 (повздовжній переріз)

Для відведення високотискового масла використовується поплавкова камера, яка подає це масло на ущільнення валу за допомогою гвинтового насоса. У разі зупинки агрегату при відсутності живлення змінним струмом передбачено використання акумулятора для масла. Як і в сучасних нагнітачах НЗЛ останніх років випуску, дана машина оснащена торцевим контактним ущільненням валу, яке запобігає витоку газу з корпусу в картер. Загальна маса нагнітача становить близько 23 тонн.

1.4. Загальна характеристика нагнітача

Характеристикою відцентрового компресора вважають залежність політропного коефіцієнта потужності ($\eta_{\text{пол}}$), ступеня стиснення (ε) та внутрішньої потужності (N_i) від витрати газу при різних значеннях частоти обертання ротора.

Продуктивність компресора визначається як об'єм газу, що перекачується через нього за певний період часу. Ступінь стиснення газу визначається відношенням тисків на виході та вході газоперекачувального агрегату (ГПА). Газоперекачувальні агрегати (ГПА) поділяються на повнонапірні (ПВН) та неповнонапірні (НВН) в залежності від ступеня стиснення. Неповнонапірні нагнітачі мають ступінь стиснення в межах від 1,23 до 1,25 і зазвичай використовують послідовно-паралельну схему на компресорних станціях (КС). Кількість ступенів стиснення в групі визначається тиском на вході та виході КС. КС може мати дво- або триступеневе стиснення в залежності від продуктивності, а паралельні групи обираються відповідно до потреб КС. Повнонапірні агрегати включають два робочих колеса в один корпус, забезпечуючи ступінь стиснення від 1,50 до 1,58, і працюють виключно паралельно на КС. Кількість робочих повнонапірних агрегатів на КС визначається продуктивністю та номінальною продуктивністю одного ГПА конкретного типу.

На основі вище наведеного:

$$N_i, \eta_{\text{пол}}, \varepsilon = f_{1,2,3}(p_B, Q_B, \omega, D_2, \rho_B), \quad (1.1)$$

де p_B - тиск;

ρ_B - густина газу при всмоктуванні;

Q_B - витрата;

ω - кутова швидкість;

D_2 - величина діаметру робочого колеса.

З використанням теорії подібності залежність (1.1) можна виразити у критерійній формі:

$$N_i / \rho_B D_2^2 u^3, \eta_{\text{пол}}, \varepsilon = f_{1,2,3}\left(Q_B / D_2^2 u_2, u_2 / \sqrt{z_B RT_B}\right), \quad (1.2)$$

де $Q_B / D_2^2 u_2$ - коефіцієнт витрати;

$u_2 = D_2 \omega$ - кругова швидкість;

$u_2 / \sqrt{z_B R T_B}$ - число Маха.

Відповідно до (1.2), можна створити графіки залежності певної безрозмірної потужності $N_i / D_2^2 u_2^3 \rho_B$ від безрозмірної витрати $Q_B / D_2^2 u_2$ при різних числах Маха. Проте відбувається це за допомогою іншого підходу. Оскільки D_2 - фіксована величина, замість окружної швидкості u_2 використовують значення частоти обертання ротора n а також переходять до наведеного нище виразу:

$$N_i / \rho_B D_2^2 u^3, \eta_{пол}, \varepsilon = f_{1,2,3} \left(Q_B / n, n / \sqrt{z_B R T_B} \right), \quad (1.3)$$

де $N_i / \rho_B D_2^2 u^3$, Q_B / n та $n / \sqrt{z_B R T_B}$ - розмірні величини, котрі являються пропорційними відповідним безрозмірним параметрам у (1.2).

Щоб узгодити характеристики з різними складами природного газу, їх необхідно привести до певних фіксованих умов, що обрані виходячи з практичної доцільності. Параметри приведення такі: $T_{B.пр} = 288$ К; $z_{B.пр} = 0,91$, $n_{пр} = n_n$ та $R_{пр} = 490$ Дж/(кг·К) (номінальна). У результаті отримуємо:

$$\eta_{пол}, \varepsilon, [N_i / \rho_B]_{пр} = f_{1,2,3} \left((n / n_n)_{пр}, Q_{пр} \right), \quad (1.4)$$

де відповідно

$$\begin{aligned} (N_i / \rho_B)_{пр} &= N_i / \rho_B (n_n / n)^3; \quad Q_{пр} = Q_B \cdot n_n / n; \\ (n_n / n)_{пр} &= n_n / n \sqrt{z_{B.пр} R_{пр} T_{B.пр} / z_B R T_B}. \end{aligned} \quad (1.5)$$

Приведені характеристики були розроблені у ВНИИгазі. На підставі широкомасштабних натурних випробувань відцентрових нагнітачів були опубліковані спеціальні альбоми з їх описом.

Визначення помпажних характеристик газоперекачувальних агрегатів із газотурбінним приводом залишається важливою науково-практичною задачею. Саме на основі аналізу цих характеристик формуються висновки щодо необхідності проведення профілактичних робіт на двигуні та нагнітачі або оновлення помпажних характеристик. Виявлене відхилення ε використовується

для корекції антипомпажної системи або регулювання витрати за допомогою байпасного крану, щоб уникнути аварійних ситуацій.

Характеристики для нагнітачів НГ-280-9, PCL-1002, PCL802/24, 520-12-1, які є приведеними наведені на рисунку 1.4.

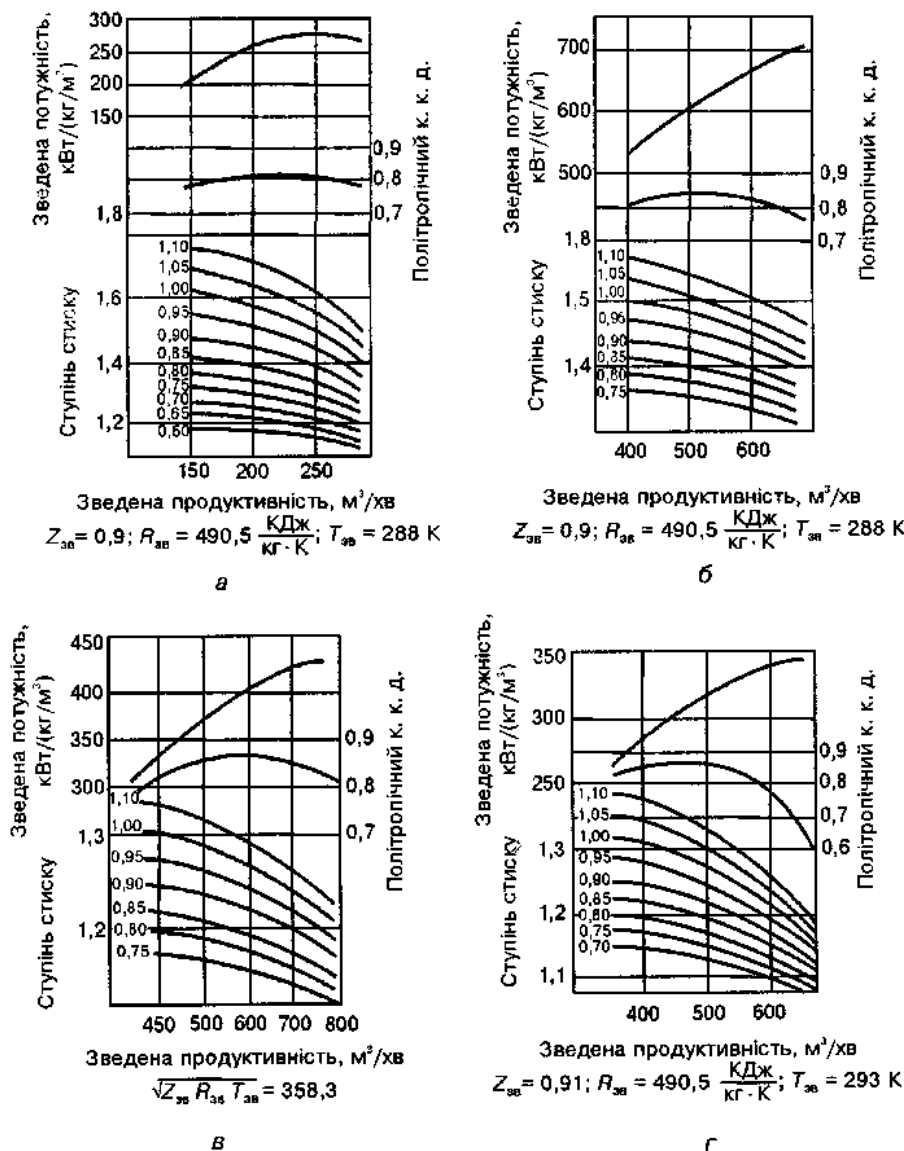


Рис. 1.4. Візуалізація приведених характеристик найпоширеніших нагнітачів:

в - НГ-280-9; а - PCL802/24; г - 520-12-1; б - PCL-1002

Для наочного підтвердження актуальності дослідження реальних кривих приведених характеристик розглянемо їхню зміну в процесі зношування елементів відцентрових нагнітачів і газоперекачувальних агрегатів. Прийнемо значення ступеня стиснення $\varepsilon=1,6$ за умови, що приведена частота обертання $n_{пр}=1,1$ для трьох типів ГПА. Для агрегату №2 (із незначним зносом) приведена

витрата становить $Q_{пр}=370$, для агрегату №3 (із середнім зносом) — $Q_{пр}=240$, а для агрегату №1 (із максимальним зносом) — $Q_{пр}=270$.

Як видно, навіть за однакового ступеня стиснення ε , агрегати №1 та №3 потрапляють у зону помпажу, що свідчить про необхідність коригування параметрів протипомпажної системи для забезпечення стабільної роботи установки.

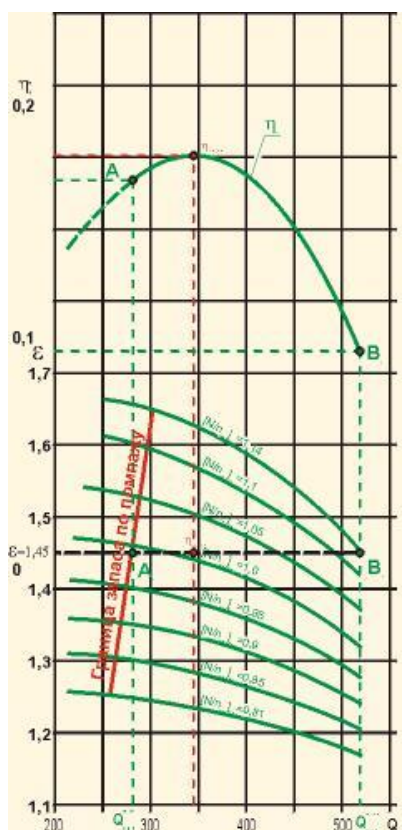


Рис. 1.5. Приведені характеристики для ГПА № 1 (із максимальним зносом)

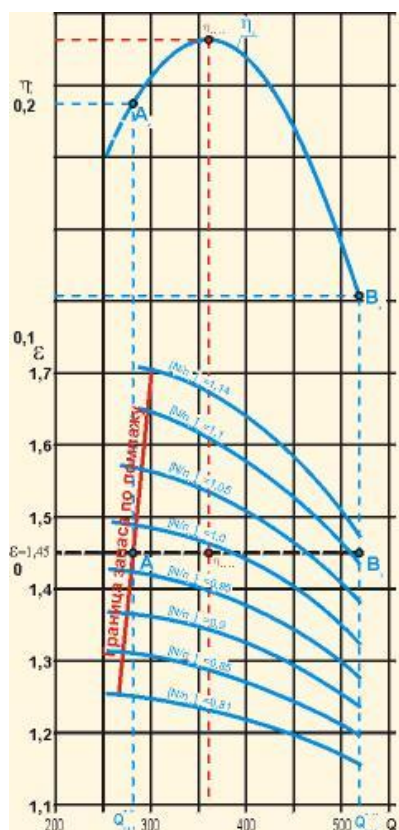


Рис. 1.6. Приведені характеристики для ГПА № 2 (із мінімальним зносом)

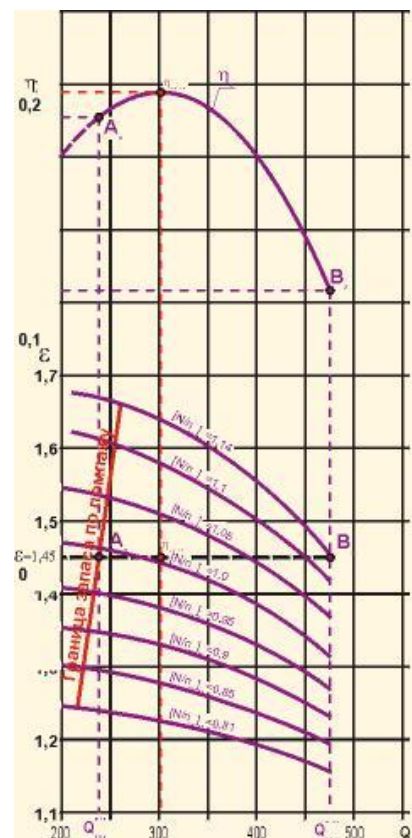


Рис. 1.7. Приведені характеристики для ГПА № 3 (із середнім зносом)

Висновки до розділу 1

Проаналізовано організаційну структуру компресорної станції та детально охарактеризовано конструктивні особливості відцентрових нагнітачів газоперекачувальних агрегатів, що розглядаються як об'єкти автоматизованого керування. Подано детальний аналіз приведених характеристик відцентрових

нагнітачів, зокрема їх зміну в умовах різного виду зносу відцентрових нагнітачів (малий, середній, максимальний). Обґрунтовано важливість даної теми і сформульовано ключові завдання для подальших досліджень.

РОЗДІЛ 2

ВИЗНАЧЕННЯ БЕЗРОЗМІРНИХ ХАРАКТЕРИСТИК ВІДЦЕНТРОВИХ НАГНІТАЧІВ

2.1. Розрахунок витрати для нагнітачів різних типів (280, 370 і 520) за перепадом тиску на всмоктуючому конфузурі

Визначення об'єму газу, що піддається компресії газоперекачувальною станцією, не завжди можливо з великою точністю. З цією метою, зокрема для контролю робочого режиму газоперекачувальних станцій, особливо тих, що мають багатоцехову структуру, доцільно використовувати метод розрахунку, ґрунтований на вимірі перепаду тиску на всмоктуючому конфузурі.

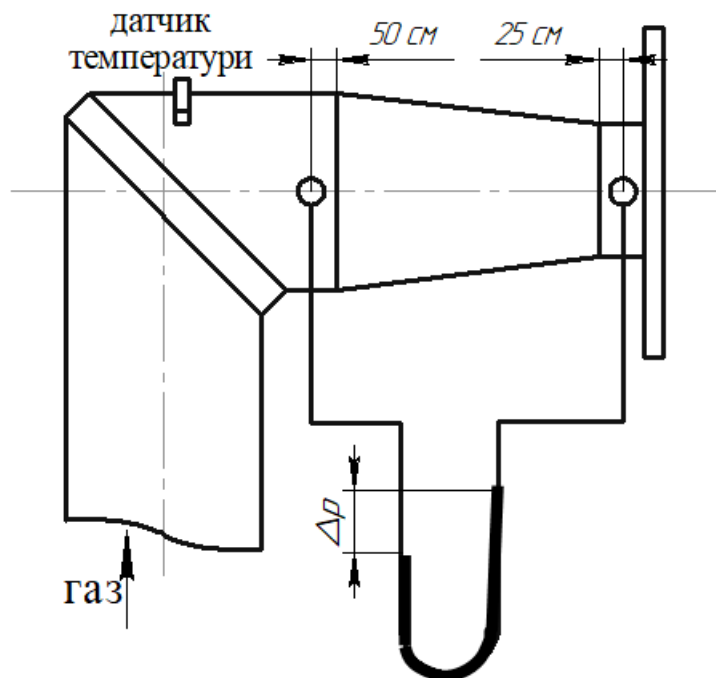


Рис. 2.1. Діаграма вимірювання об'ємної витрати газу через нагнітач, заснована на вимірюванні перепаду величини тиску на конфузурі

Об'ємну витрату газу (в м³/хв) для нагнітача типу 280 визначають шляхом проведення вимірів з використанням схеми (рис. 2.1), яка є рекомендацією заводу виробника:

$$Q = k \sqrt{\Delta p / \gamma_n}, \quad (2.1)$$

де γ_n - питома вага газу перед відцентровим нагнітачем, кг/м³;

Δp - перепад величини тиску газу на конфузори, мм рт. ст.;

k - коефіцієнт пропорційності конфузора (для нагнітачах типу 280, рекомендується приймати 68,4).

На рисунку 2.2 подано номограму для визначення питомої ваги газу γ , об'ємної витрати за перепадом тиску на конфузори нагнітача а також величини $\sqrt{z \cdot R \cdot T}$. З метою визначення продуктивності відцентрового нагнітачів, які мають тип 370 та відповідно 520 проводять вимірювання перепаду тиску Δp на всмоктуючих камерах на деякій ділянці між вхідним патрубком відцентрового нагнітача та входом у певне робоче колесо. Тиск при вході в колесо відбирають за допомогою дренажів, які використовуються для подачі імпульсу на систему регулювання. Наявні дренажні отвори розташовані на виході з конфузора всмоктуючої камери і закінчуються двома фланцями діаметром $D_y=15$ мм на корпусі нагнітача, які служать для двох дублюючих один одного дренажних трубок.

Точна обробка поверхні конфузорів із допуском 0,1 мм забезпечує уніфікованість їх геометричних параметрів і, відповідно, однакові коефіцієнти витрати у всмоктуючих камерах для всіх нагнітачів одного типу. Отвори для вимірювання тиску перед всмоктуючою камерою необхідно розташовувати на всмоктуючому трубопроводі перед вхідним патрубком нагнітача.

Об'ємну витрату газу при умовах всмоктування визначають за такою залежністю::

$$Q = A \cdot l \sqrt{\Delta p / \gamma}, \quad (2.2)$$

де A - коефіцієнт витрати;

Δp - значення перепаду тиску у камері всмоктування, мм рт. ст.;

γ - значення питомої густини газу, кг/м³;

l – значення поправки на стисливість газу у камері згорання.

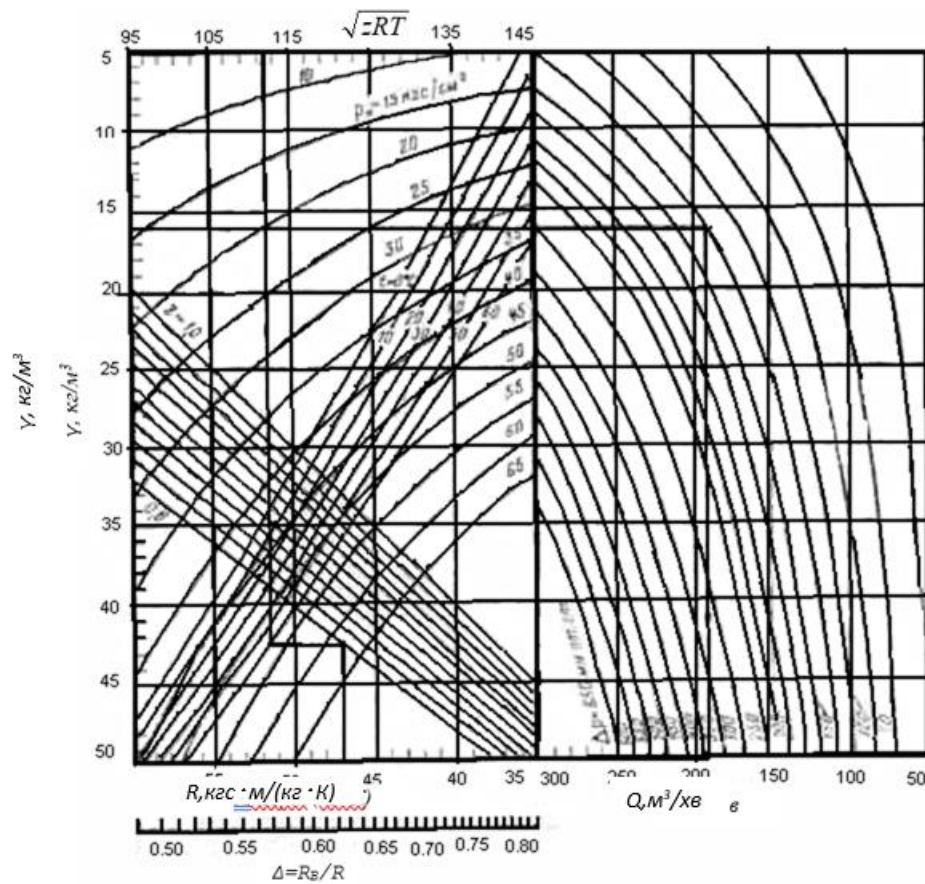


Рис. 2.2. Графічна залежність, що використовується для визначення питомої ваги газу, об'ємної витрати за перепадом тиску на конфузори нагнітача та відповідних параметричних значень \sqrt{zRT}

Послідовність визначення значень за допомогою номограми:

$$\begin{array}{ccccccc}
 R & \rightarrow & z & \rightarrow & T & \rightarrow & p_n & \rightarrow & \Delta p \\
 & & & & \downarrow & & \downarrow & & \downarrow \\
 & & & & \sqrt{zRT} & & \gamma & & Q
 \end{array}$$

Об'ємна витрата нагнітача (у мільйонах кубічних метрів на добу) при температурі 20,0 °С і атмосферному тиску 760,0 мм рт. ст. розраховується за такою формулою:

$$Q = 1.44 \cdot 10^{-3} \cdot A \cdot l / \gamma_n \sqrt{\gamma \cdot \Delta p}, \quad (2.3)$$

де γ_n - питома густина газу за температури 20 °С і тиску 760 мм рт. ст., кг/м³.

$$\gamma_n = 1033 / 293 \cdot R = 35,2 / R. \quad (2.4)$$

Значення коефіцієнтів витрати для камер типу А та відповідні рівняння для визначення поправок l , отримані в ході стендових випробувань, подані в таблиці 2.1. Формула для обчислення l наведена для випадку, коли показник адиабати газу дорівнює $k = 1,310$.

Таблиця 2.1

Визначення коефіцієнта витрати та розрахунок корекційного параметра, що враховує стисливість газу

Вид нагнітача	Обрахунок	А
370	$l = 1 - 1.102 \cdot 10^{-3} \cdot \Delta p / p_n$	91,00
520	$l = 1 - 1.385 \cdot 10^{-3} \cdot \Delta p / p_n$	128,80

Характеристики, які були приведені для нагнітача типу 370-14-1, представлено на рисунку 2.3.

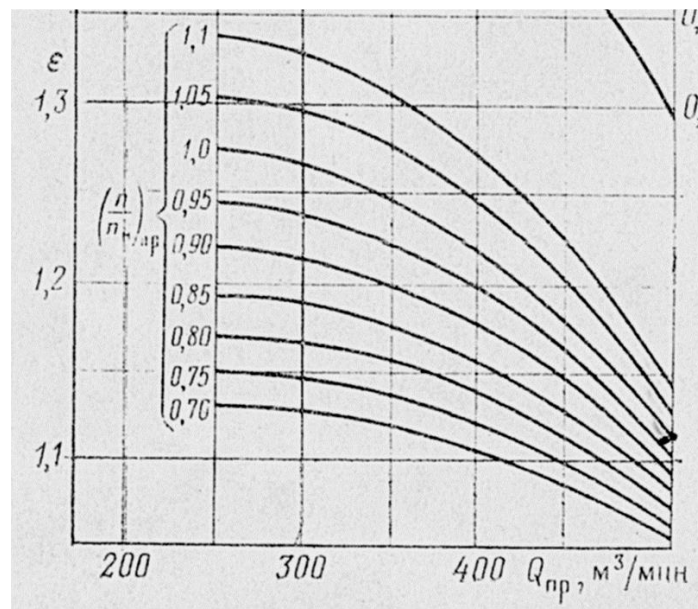


Рис. 2.3. Приведені ($n_n=5300$ об/хв; $z_{B.pr} = 0,91$; $T_{B.pr} = 288$ К, $R_{пр} = 500$ Дж/(кг·К)) характеристики нагнітача типу 370-14-1

2.2. Визначення приведених характеристик $Q_{пр}$ і $E_{пр}$

Об'ємна витрата нагнітача (у $m^3/хв$) визначається згідно з наступною формулою:

$$Q_{пр} = Q \cdot n_{ном} / n,$$

де n - реальна швидкість обертання ротора компресора, виражена в обертах за хвилину;

$n_{ном}$ - проектна швидкість обертання ротора, вимірювана в обертах за хвилину.

Під час проведення розрахунків скоригованих (приведених) характеристик необхідно визначити такі параметри:

1. Сталу газову, R , Дж/кг·К:

$$R = 288 / \Delta,$$

де Δ - відносна щільність газу щодо повітря;

288 - повітряна газова стала, Дж/кг·К.

2. При температурі $T = 293$ °К та тиску $P = 0,10130$ МПа, кг/м³ густина газу:

$$\rho_n = \Delta \cdot 1,2060,$$

де 1,206 - щільність повітря за стандартних умов, кг/м³

3. Величину коефіцієнта стисливості природного газу (Z) можна визначити за допомогою номограми, зображеної на рис. 2.4.

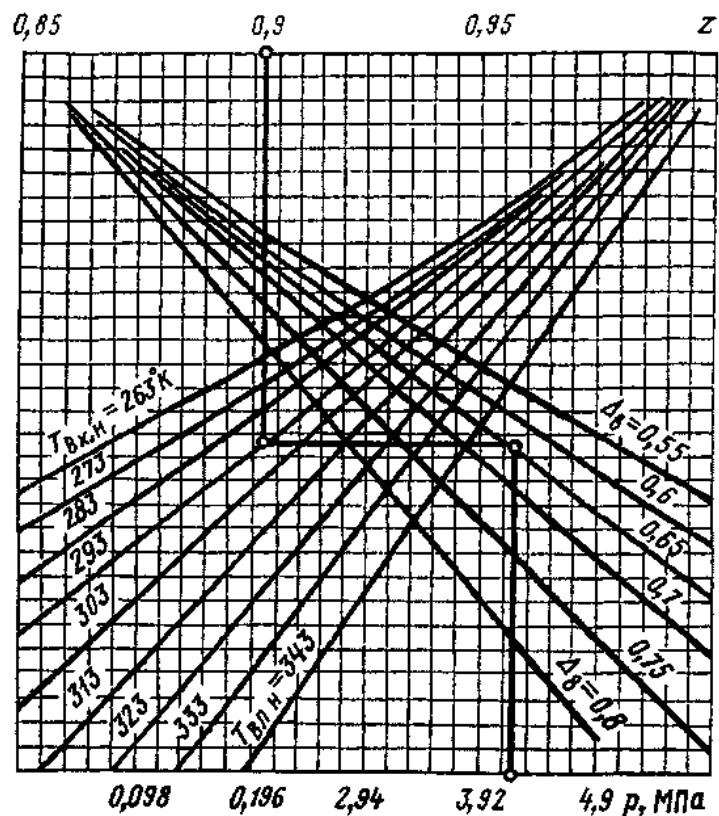


Рис. 2.4. Графічний засіб для визначення коефіцієнта стисливості газу Z в залежності від тиску P , температури $T_{вх.н}$ та відносної густини Δ_v

Етапи визначення:

$$P \rightarrow \Delta_b, T_{ex. n} \rightarrow Z$$

4. Величина густини газу при процесі всмоктування, кг/м³:

$$\rho = P_{ex} / R \cdot Z \cdot T,$$

де P_{ex} та T - абсолютний тиск (у паскалях) та температуру (у градусах кельвіна) під час всмоктування.

5. Нормалізовану частоту обертання: визначимо за формулою:

$$\left[n / n_n \right]_{np} = n / n_n \sqrt{Z_{np} R_{np} T_{np} / Z \cdot R \cdot T},$$

де T_{np} , R_{np} , Z_{np} - характеристики газу, які використовуються при описі роботи нагнітача.

6. Величина тиску на виході із нагнітача у МПа:

$$P_{вих} = \varepsilon \cdot P_{ex}.$$

Після розрахунку приведеної витрати на основі вимірних параметрів газоперекачувального агрегату (ГПА), визначається приведена ступінь стиснення нагнітача, відповідна приведеним параметрам газу, за допомогою формули:

$$E_{np} = E \left[\left(n / n_{ном} \right)_{np} \right]^{-2},$$

де $(n / n_{ном})_{np}$ - нормована частота обертання нагнітача;

E_{np} - ступінь стиснення нагнітача яка є приведеною;

E - величина стиснення нагнітача.

Наступною кроком є формування системи рівнянь, що містить вирази для приведеної ступені стиснення та приведеної витрати в кожному з трьох режимів роботи газоперекачувального агрегату. Система представлена поліномами третього порядку, кожен з яких містить чотири невідомі коефіцієнти для відповідного режиму:

$$E_{np1} = A_3 Q_{np1}^3 + A_2 Q_{np1}^2 + A_1 Q_{np1} + A,$$

$$E_{np2} = A_3 Q_{np2}^3 + A_2 Q_{np2}^2 + A_1 Q_{np2} + A,$$

$$E_{np3} = A_3 Q_{np3}^3 + A_2 Q_{np3}^2 + A_1 Q_{np3} + A.$$

Завдяки вирішенню цієї системи рівнянь знаходяться значення коефіцієнтів

A - A₃. Це здійснюється за допомогою рівняння (2.6) та наступного рівняння:

$$E_{np} = A_3 Q_{np}^3 + A_2 Q_{np}^2 + A_1 Q_{np} + A.$$

На цій основі формується графік фактичної характеристики нагнітача яка є приведеною $E=f(a)$, і після введення значень $(n/n_{ном})_{пр.}=1,1; 0,85; 1,05; 1; 0,8; 0,95; 0,75; 0,9; 0,7$ відбувається побудова графіків фактичних приведених характеристик відцентрових нагнітачів.

2.3 Проведення розрахунків і побудова графіків фактичної приведеної характеристики нагнітача у середовищі Mathcad

Нагнітач типу Н-370-14-1 і параметри.

Значення густини газу: $\rho_n := 0,6720 \text{ кг/м}^3$.

Значення коефіцієнту стисливості: $z := 0,90$.

Значення температури газу на вході нагнітача: $t := 15,0^\circ\text{C}$; $T := 273,0 + t$;
 $T = 288,0^\circ\text{K}$.

Фактичне значення частоти обертання ротора нагнітача:

$$n := (3800,0 \quad 4450,0 \quad 5500,0), \text{ об/хв.}$$

Умови середовища для проведення: $R_{pr} := 500, \text{ Дж/кг} \cdot ^\circ\text{K}$; $z_{pr} := 0,91$.;
 $n_n := 5300, \text{ об/хв.}$; $T_{pr} := 288, ^\circ\text{K}$.

Величина перепаду тиску на всмоктуючій камері:

$$dP := (0.0214 \quad 0.056066 \quad 0.121336), \text{ МПа.}$$

$$dP := dP / 0.1333 \cdot 10^{-3}.$$

$$dP = (160.54 \quad 420.6 \quad 910.248), \text{ мм рт.ст.}$$

Значення тиску перед всмоктуючою камерою:

$$P := (4.9023 \quad 4.8062 \quad 4.7033), \text{ МПа.}$$

$$P := P / 0.0980665.$$

$$P = (49.99 \quad 49.01 \quad 47.96), \text{ кгс/см}^2.$$

Значення барометричний тиск:

$$P_a := 0,10 \text{ МПа} .$$

$$P_a := P_a / 735,5 \cdot 0.133322 \cdot 10^{-3} .$$

$$P_a = 1,020, \text{ кгс/см}^2 .$$

Значення тиску на вході нагнітача:

$$P_{vx} := P + P_a .$$

$$P_{vx} = (51.009 \quad 50.029 \quad 48.98), \text{ кгс/см}^2 .$$

Робочий тиск на виході нагнітача:

$$P_{vux} := 5,60 \text{ МПа} .$$

$$P_{vux} := P_{vux} / 0.0980665 .$$

$$P_{vux} = 57,1040, \text{ кгс/см}^2 .$$

Визначення відносної густини газу за повітрям:

$$\Delta := \rho_n / 1.206 .$$

$$\Delta := 0,5570 \text{ кг/м}^3 .$$

Визначення газової сталої:

$$R := 288 / \Delta .$$

$$R := 516,8570, \text{ Дж/кг} \cdot \text{°К} .$$

Визначення густини газу при всмоктуванні:

$$\rho := P_{vx} \cdot 10^5 / R \cdot z \cdot T .$$

$$\rho = (38,0750 \quad 37,3440 \quad 36,5610), \text{ кг/м}^3 .$$

Величина коефіцієнтів витрати камер:

$$A := 91,0 .$$

Значення поправок для нагнітача типу 370:

$$l_0 := 1.102 \cdot 10^{-3} \cdot dP_0 / P_{vx_0} + 1 .$$

$$l_2 := -1.102 \cdot 10^{-3} \cdot dP_2 / P_{vx_2} + 1 .$$

$$l_1 := 1.102 \cdot 10^{-3} \cdot dP_1 / P_{vx_1} + 1 .$$

$$l = (0.997 \quad 0.991 \quad 0.98) .$$

Визначення витрат на вході у нагнітач:

$$Q_{v_0} := l_0 \cdot A \cdot \sqrt{dP_0 / \rho_0} .$$

$$Q_{v_1} := l_1 \cdot A \cdot \sqrt{dP_1 / \rho_1} . \text{ю бю}$$

$$Q_{v_2} := l_2 \cdot A \cdot \sqrt{dP_2 / \rho_2} .$$

$$Q_v = (186.21 \quad 302.569 \quad 444.762), \text{ м}^3 / \text{хв}.$$

Частота обертання, яка є приведеною:

$$n_{pr} := n / n_n \cdot \sqrt{R_{pr} \cdot z_{pr} \cdot T_{pr} / R \cdot z \cdot T} .$$

$$n_{pr} = (0.709 \quad 0.83 \quad 1.026) .$$

Визначимо степінь стиснення:

$$E_0 := P_{vux} / P_{vx_0} . \quad E_1 := P_{vux} / P_{vx_1} . \quad E_2 := P_{vux} / P_{vx_2} .$$

$$E = (1.119 \quad 1.141 \quad 1.166)$$

Визначимо відносні частоти наявного обертання відцентрового нагнітача (ВН):

$$n' := n_n / n .$$

$$n' = (1.395 \quad 1.191 \quad 0.964)$$

Визначимо режимні параметри ВН:

$$E_{pr1} := E_0^{(n_{pr0})^{-2}} .$$

$$E_{pr1} = 1,2520 .$$

$$Q_{pr1} := n'_0 \cdot Q_{v_0} .$$

$$Q_{pr1} = 259,7130 \text{ м}^3 / \text{хв} .$$

$$Q_{pr2} := n'_1 \cdot Q_{v_1}$$

$$Q_{pr2} = 360,3630 \text{ м}^3 / \text{хв} .$$

$$Q_{pr3} := n'_2 \cdot Q_{v_2} .$$

$$Q_{pr3} = 428,58900 \text{ м}^3 / \text{хв} .$$

$$E_{pr2} := E_1^{(n_{pr1})^{-2}}$$

$$E_{pr2} = 1,2110 .$$

$$E_{pr3} := E_2^{(n_{pr2})^{-2}}.$$

$$E_{pr3} = 1,1570.$$

Визначимо параметри А - А₃:

$$A_3 := 0; \quad A_2 := 0; \quad A_1 := 0; \quad A := 0.$$

Given

$$E_{pr1} := A_3 \cdot Q_{pr1}^3 + A_2 \cdot Q_{pr1}^2 + A_1 \cdot Q_{pr1} + A.$$

$$E_{pr2} := +A_3 \cdot Q_{pr2}^3 + A_2 \cdot Q_{pr2}^2 + A_1 \cdot Q_{pr2} + A.$$

$$E_{pr3} := A_3 \cdot Q_{pr3}^3 + A_2 \cdot Q_{pr3}^2 + A_1 \cdot Q_{pr3} + A.$$

$$\begin{pmatrix} A_3 \\ A_2 \\ A_1 \\ A \end{pmatrix} := \text{Find}(A_3, A_2, A_1, A).$$

$$A_3 := -2.984 \cdot 10^{-9}; \quad A_2 := 7.557 \cdot 10^{-7}; \quad A := 1.253; \quad A_1 := 0.$$

Побудова діаграми з кривими фактичних приведених характеристик відцентрового нагнітача:

$$Q_{pr} := 250,0 \dots 500,0.$$

$$E(Q_{pr}) := (A_3 \cdot Q_{pr}^3 + A_2 \cdot Q_{pr}^2 + A_1 \cdot Q_{pr} + A)^{1.21}$$

$$E1(Q_{pr}) := (A_3 \cdot Q_{pr}^3 + A_2 \cdot Q_{pr}^2 + A_1 \cdot Q_{pr} + A)^{1.1025}$$

$$E2(Q_{pr}) := (A_3 \cdot Q_{pr}^3 + A_2 \cdot Q_{pr}^2 + A_1 \cdot Q_{pr} + A)$$

$$E3(Q_{pr}) := (A_3 \cdot Q_{pr}^3 + A_2 \cdot Q_{pr}^2 + A_1 \cdot Q_{pr} + A)^{0.9025}$$

$$E4(Q_{pr}) := (A_3 \cdot Q_{pr}^3 + A_2 \cdot Q_{pr}^2 + A_1 \cdot Q_{pr} + A)^{0.81}$$

$$E5(Q_{pr}) := (A_3 \cdot Q_{pr}^3 + A_2 \cdot Q_{pr}^2 + A_1 \cdot Q_{pr} + A)^{0.7225}$$

$$E6(Q_{pr}) := (A_3 \cdot Q_{pr}^3 + A_2 \cdot Q_{pr}^2 + A_1 \cdot Q_{pr} + A)^{0.64}$$

$$E7(Q_{pr}) := (A_3 \cdot Q_{pr}^3 + A_2 \cdot Q_{pr}^2 + A_1 \cdot Q_{pr} + A)^{0.512}$$

$$E8(Q_{pr}) := (A_3 \cdot Q_{pr}^3 + A_2 \cdot Q_{pr}^2 + A_1 \cdot Q_{pr} + A)^{0.49}$$

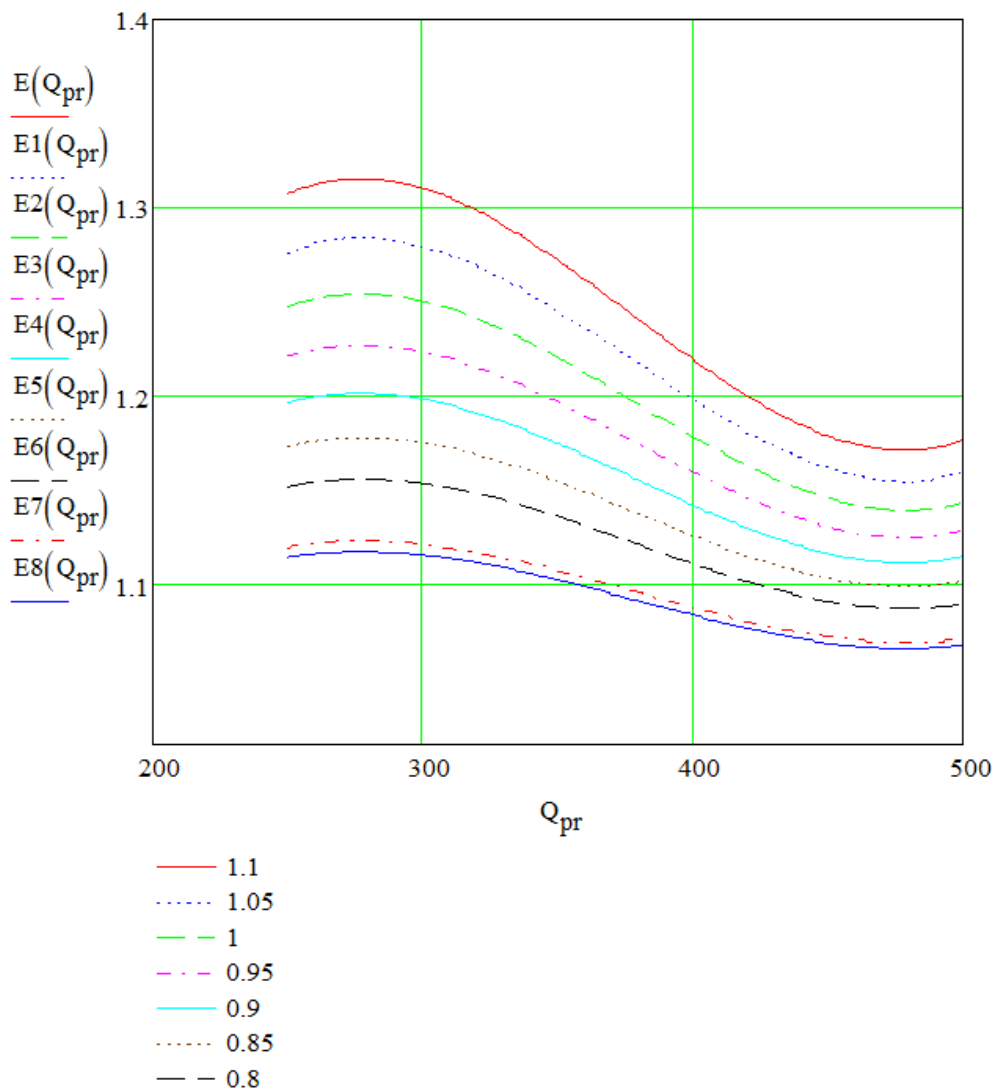


Рис. 2.5. Графіки, де зображено залежність ступеня стиснення (E) від значення приведеної витрати (Q_{pr})

Висновки до розділу 2

Запропонований метод визначення об'ємної витрати газу через відцентровий нагнітач за перепадом тиску на всмоктуючому конфузори дозволяє формувати його приведені характеристики в режимі реального часу. Отримані приведені величини Q_{pr} та E_{pr} , а також побудовані графіки фактичних приведених характеристик відцентрових нагнітачів газоперекачувальних агрегатів (ГПА), забезпечують можливість оперативного аналізу технічного стану ГПА та прогнозування потенційних аварійних ситуацій.

РОЗДІЛ 3

АВТОМАТИЗАЦІЯ ПРОЦЕСУ ВИЗНАЧЕННЯ ПОТОЧНИХ ХАРАКТЕРИСТИК ВІДЦЕНТРОВОГО НАГНІТАЧА

3.1. Алгоритм візуальної побудови кривих

На основі проведених обчислень можна створити алгоритм програми, який представлений на рисунку 3.1.

Опис алгоритму:

• Блок №2. Здійснюється введення таких даних:

- $P_{вх}$ - величина тиску на вході у ГПА;
- $T_{вх}$ - значення температури газу на вході у відцентровий нагнітач;
- $P_{вих}$ - величина тиску на виході із ГПА;
- dP_k - значення перепаду тиску на конфузіві всмоктуючому;
- n – фактичне значення частоти обертання ротора нагнітача.

Ці дані отримуються з сенсорів, встановлених на газоперекачувальному апараті (ГПА). Такі параметри як $P_{вх}$, n , dP_k зчитуються для усіх трьох режимів роботи ГПА.

• Блок №3. Введення даних про нагнітач:

1. Значення з лабораторії:

- P_a – тиск атмосферний;
- ρ – значення густини газу;
- z - коефіцієнт стисливості.

2. Значення приведених характеристик:

- $z_{пр}$ - коефіцієнт стисливості приведений;
- $R_{пр}$ - приведена газова стала;
- $T_{пр}$ - температура приведена.

3. Режимні значення нагнітача:

- $n_{ном}$ – величина номінальних обертів нагнітача;
- A – розрахунковий коефіцієнт витрати;

- l – значення поправки на стиснення газу в камері.

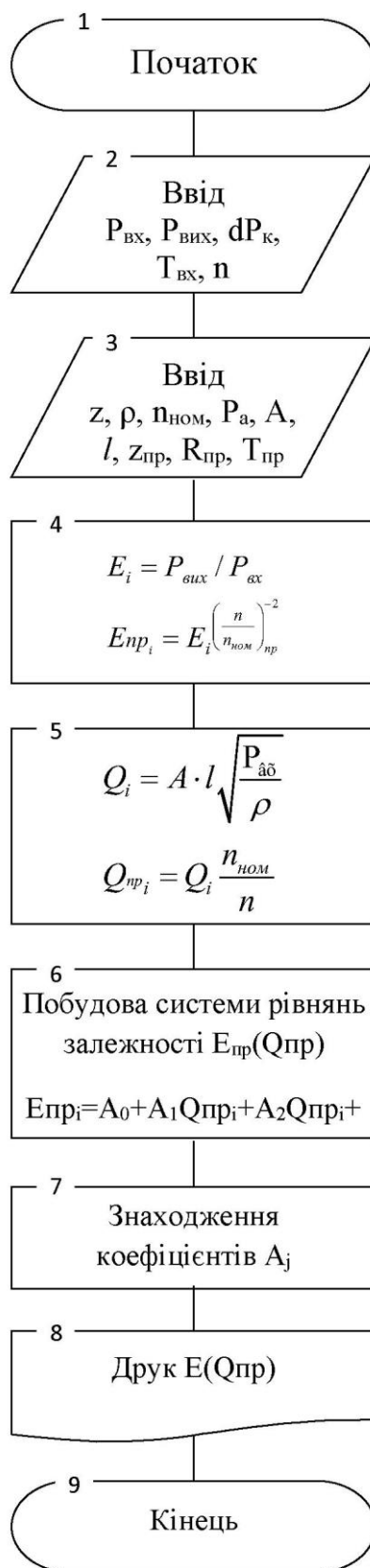


Рис. 3.1. Алгоритм роботи

- Блок 4. Розрахунок приведеної потужності стиснення $E_{пр}$ відцентрового нагнітача для кожного з трьох режимів роботи газоперекачувального агрегату (ГПА).
- Блок 5. Обчислення приведеної витрати $Q_{пр}$ для кожного із наявних трьох режимів роботи для газоперекачувального апарата (ГПА).
- Блок 6. Формується система рівнянь, що містить вирази для приведенного ступеня стиснення та приведеної витрати для кожного з трьох режимів роботи газоперекачувального агрегату (ГПА). Ці вирази представлені поліномами третього порядку з чотирма невідомими коефіцієнтами.
- Блок 7. Обчислення із системи рівнянь коефіцієнтів $A_0 - A_3$.
- Блок 8. Створення кривих фактичних приведених характеристик відцентрового нагнітача (ВН) при номінальних значеннях 1,05; 1; 1,1; 0,95; 0,9; 0,8; 0,85; 0,7; 0,75.

3.2. Методика визначення приведених характеристик

При розробці системи визначення приведених характеристик відповідно до описаної методології, необхідно провести вимірювання перепаду тиску газу (dP_k), тиску на вході ($P_{вх}$) та виході ($P_{вих}$) газоперекачувального апарата (ГПА), температури газу на вході ($T_{вх}$), густини газу (ρ) та фактичної частоти обертання (n).

Створена система складається з програмованого логічного контролера, первинних вимірювальних перетворювачів, блоку перетворення інтерфейсів а також робочої станції на якій встановлено програмне забезпечення (див. рис. 3.2). У таблиці 3.1 подано технологічні параметри ГПА, які необхідні для операції системи, а також наявні технічні характеристики вимірюваних каналів та первинних перетворювачів.

Всі зазначені параметри, за винятком перепаду тиску dP_k , автоматично зчитуються та використовуються в системі централізованого контролю а також управління (СЦКУ) газоперекачувального апарата (ГПА). Отже, додаткове

встановлення сенсорів не є необхідним.



Рис. 3.2. Блокова схема розробленої системи

Таблиця 3.1

Параметри процесу та властивості
каналів вимірювання газоперекачувального апарату

№ з/п	Найменування параметрів	Скорочення	Одиниці вимірювання	Давач	Тип сигналу
1	Величина перепаду тиску газу на конфузори	dP_k	0,0 - 0,15 МПа	Honeywell 3000 series	4-20мА
2	Тиск газу на виході нагнітача	$P_{вих}$	0,0 – 10,0 МПа	TEX UAG97L	
3	Тиск газу на вході нагнітача	$P_{вх}$	0,0 – 10,0 МПа	TEX UAG97L	
4	Температура газу на вході	$T_{вх}$	-50 ÷ 60 °С	ТСП 23-є град.	інтерфейс*
5	Частота обертів ВН	n	0,0 – 5000,0 об/хв	ВЦТ-1/Л	

Отримані дані були внесені до програмованого мікропроцесорного контролера МК-51. Давайте наведемо короткий опис його характеристик:

- Функціональні особливості:
 - Контролер МК-51 є конструктивно-монтажним виробом, що надає користувачеві можливість вибору необхідного набору модулів та блоків в залежності від кількості та типів вхідних та вихідних сигналів.
 - Вбудовані засоби самодіагностики включають в себе сигналізацію та ідентифікацію несправностей, пов'язаних з виходженням сигналів за встановлені межі деяких допустимих значень, або виникненням проблем у операційному засобі, порушенням обміну даними по мережі та інші подібні неполадки.
 - Програмування контролера можливо як безпосередньо через клавіші на передній панелі, так і через відповідний інтерфейс із використанням

спеціалізованого програмного забезпечення, зокрема візуального редактора FBD-програм АЛЬФА. Для конфігурації контролера застосовується мова функціональних блоків (FBD). Редактор FBD оснащений вбудованим відладчиком програм, системою логічного контролю поточного стану виконання, а також інструментами для документування, друку програм, їх відображення у вигляді таблиць і низкою інших функцій, що забезпечують ефективну розробку та тестування програм.

- Обчислення та обробка математичної інформації за допомогою різноманітних алгоритмів:
 - Алгоритми введення-виведення включають аналоговий та дискретний методи.
 - Алгоритми регулювання охоплюють ПД-регулювання, ПД-каскадне регулювання, швидкість зміни значення параметра, програмний задатчик, алгоритм користувачької панелі із трьома вказаними параметрами та сигнальний задатчик.
 - Процедури керування та обслуговування передньої панелі контролера.
 - Алгоритми динамічних перетворень включають інтегрування, диференціювання, фільтрацію, балансування, обмеження та затримку сигналу.
 - Алгоритми статичних перетворень охоплюють сумування, множення, ділення, обчислення квадратного кореня, визначення абсолютного значення, обчислення середнього, пошук максимуму та мінімуму, визначення екстремумів і масштабування.
 - Аналого-дискретні перетворення передбачають перемикання аналогових сигналів (мультиплексор), вибір за номером, роботу компаратора, трьохпозиційного перемикача та імпульсне модуляційне управління (ШІМ-модулятор).
 - Логічні операції включають АБО, тригер, І, виключаюче АБО, регістр із записом за значенням рівня, регістр із записом за значенням фронту, та виділення фронту.
 - Дискретне керування включає таймер, лічильник, мультивібратор,

шифратор та дешифратор.

- Алгоритми групового контролю та управління включають кроково-етапну програму.

- Кількість входів та виходів контролера:

- Базова модель включає:

- Є 4 аналогових входи (2 універсальні та 2 уніфіковані) і 1 аналоговий вихід.

- Кількість дискретних виходів - 5. Кількість дискретних входів - 3.

- Конфігурації модулів систем автоматизованого управління:

- Один вхід з дискретним сигналом і чотири виходи з дискретними сигналами;

- Чотири входи з дискретними сигналами і один вихід з дискретним сигналом;

- Два входи з дискретними сигналами і три виходи з дискретними сигналами.

Контролер МІК-51 оснащений засобами оперативного управління, розташованими на передній панелі, які забезпечують можливість ручного регулювання режимів роботи, встановлення необхідних завдань, контролю за поточним виконанням програми, ручного керування виконавчими пристроями, моніторингу сигналів та відображення інформації про виниклі помилки. Контролери МІК можуть бути об'єднані в локальну керуючу мережу з шинною конфігурацією, використовуючи інтерфейс RS-485 та протокол ModBus, і можуть бути підключені до 32 контролерів.

Процес програмування мікропроцесорного контролера виконується через клавіші передньої панелі чи інтерфейс за допомогою спеціалізованого програмного забезпечення, а також редактора АЛЬФА для створення FBD-програм. У системі використовується мова програмування функціональних блокових діаграм (Function Block Diagram, FBD). Бібліотека містить понад 50 типів вбудованих у ПЗП функціональних блоків для обробки неперервних і дискретних сигналів, включно з ПІД-регуляторами, динамічними, нелінійними,

математичними, аналого-дискретними та логічними перетвореннями.

Час обробки функціональних блоків та інтерфейсного каналу не перевищує тривалості циклу і становить 0,1 секунди. Програмне середовище відповідає стандарту Міжнародної електротехнічної комісії ІЕС 1131-3 і призначене для розробки прикладного програмного забезпечення для збору технологічних даних та управління технологічними процесами на програмованих логічних контролерах.

Програма конфігурації контролера МІК-51 у вигляді функціональної блокової діаграми (FBD) наведена в Додатку А.

Контролер МІК-51 здійснює обробку даних та передає їх через протокол MODBUS-RTU на блок інтерфейсного перетворення БПІ-52. Далі інформація передається через інтерфейс USB 2.0 на робочу станцію (ПК) з OPC-сервером, що забезпечує аналіз даних різними методами та порівняння результатів без додаткових витрат часу на синхронізацію, обробку та передачу інформації.

Використання технології OPC (OLE for Process Control) дозволяє одночасний доступ до даних у реальному часі для необмеженої кількості клієнтів, відкриваючи широкі можливості для наукового аналізу за допомогою різних програмних пакетів, а також для використання спеціалістами з експлуатації та науковцями. Збір і аналіз даних із різних типів газоперекачувальних агрегатів (ГПА) є обов'язковим етапом при створенні адаптивних та інтелектуальних систем протипожежного захисту.

Програмне забезпечення ModBus OPC Server, розроблене компанією "МІКРОЛ" (далі – «сервер»), створене для автоматизації обміну даними між приладами виробника та клієнтськими додатками на ЕОМ. «Сервер» забезпечує стандартний інтерфейс доступу до даних OPC Data Access 2.0 відповідно до специфікації OPC Data Access Custom Interface Specification 2.0. Приклад налаштування конфігурації «сервера» для контролера МІК-51 наведено на рис. 3.3.

Для розробки програми визначення приведених характеристик використано програмний пакет Matlab, який надає широкий спектр функцій для виконання

технічних розрахунків, а також інтерактивне середовище для аналізу даних, створення алгоритмів та розробки додатків. Остання версія Matlab містить значні вдосконалення у програмуванні, генерації виконуваного коду, побудові графіків, візуалізації, математичних обчисленнях та підвищенні продуктивності.

Новий розширювальний пакет OPC Toolbox дозволяє ефективно створювати моніторингові, аналітичні, оптимізаційні та керуючі застосунки, використовуючи виробничі дані в режимі реального часу. Він забезпечує підключення до OPC-серверів та можливість читання і запису OPC-даних у середовищі Matlab як у синхронному, так і в асинхронному режимі. Крім того, пакет надає графічний інтерфейс для перегляду OPC-серверів і налаштування клієнтських параметрів даних.

Таким чином, використання програми ModBus OPC Server дозволяє забезпечити обмін інформацією між пристроями, виготовленими компанією "МІКРОЛ", та пакетом Matlab.

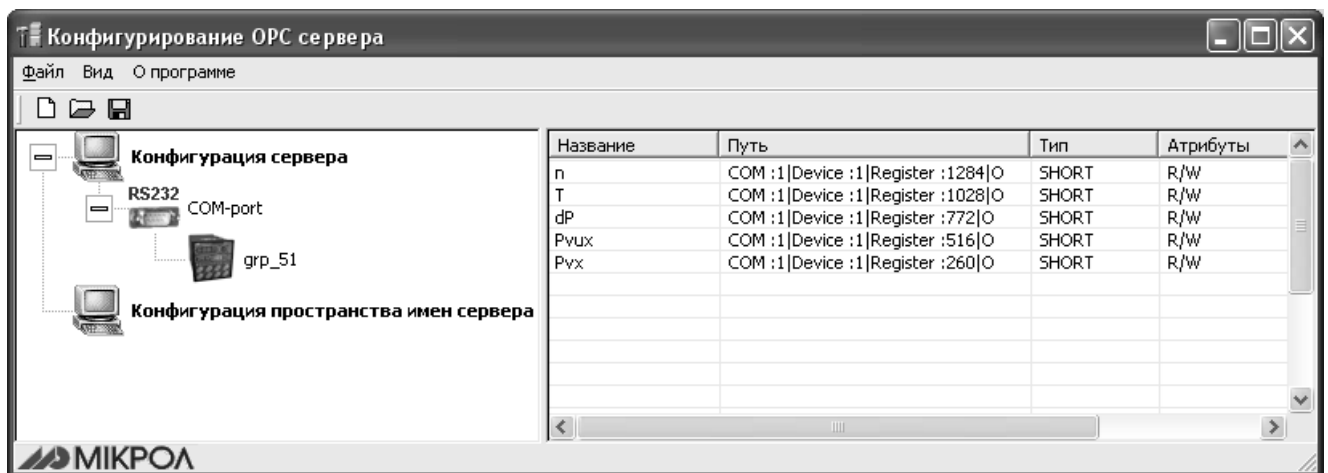


Рис. 3.3. Налаштування конфігурації OPC-сервера для використання з контролером МІК-51

Пакет програм Matlab включає в себе п'ять основних компонентів:

- Matlab є високорівневою мовою програмування, орієнтованою на роботу з матрицями та масивами, що включає вбудовані можливості для керування потоками виконання, функціями, структурами даних, операціями введення-виведення та підтримку об'єктно-орієнтованого програмування. Це дозволяє

розробляти як невеликі програми для швидкого прототипування, так і великі, складні програмні застосунки.

- Середовище Matlab представляє собою комплекс інструментів і налаштувань, з якими користувач взаємодіє під час роботи. Воно забезпечує управління змінними у робочому просторі Matlab, обробку введення і виведення даних, а також створення, контроль і відладку М-файлів та програмних додатків.

- Графічний модуль Matlab, відомий як керована графіка, забезпечує високорівневі інструменти для візуалізації дво- та тривимірних даних, обробки зображень, створення анімацій та ілюстративної графіки. Крім того, він містить низькорівневі команди, що дозволяють гнучко налаштовувати вигляд графічних об'єктів і застосовувати їх при розробці графічного інтерфейсу користувача (GUI) для підпрограм Matlab.

- Бібліотека математичних функцій Matlab охоплює широкий спектр обчислювальних алгоритмів, починаючи з базових функцій, таких як синус, косинус, сумування та комплексна арифметика, і включає більш складні операції, зокрема роботу з матрицями, функції Бесселя, обчислення власних значень та швидке перетворення Фур'є, що забезпечує вирішення як стандартних, так і спеціалізованих математичних задач.

- Програмний інтерфейс Matlab представляє собою бібліотеку, що забезпечує можливість розробки програм на мовах C і Fortran з інтеграцією у Matlab. Він включає засоби для виклику зовнішніх програм з Matlab (динамічний зв'язок), використання Matlab як обчислювальної платформи, а також читання та запис файлів у форматі MAT.

Simulink, додатковий компонент Matlab, є інтерактивним середовищем для моделювання нелінійних динамічних систем. Керування моделями здійснюється за допомогою миші, дозволяючи створювати процеси шляхом перетягування блоків діаграми та взаємодії з ними. Середовище Simulink підтримує моделювання різноманітних систем, включно з нелійними, дискретними, лінійними, безперервними та багатовимірними.

Blocksets представляють собою розширення Simulink, що надають

бібліотеки блоків для специфічних застосувань, таких як енергетичні системи, телекомунікації та обробка сигналів. Real-Time Workshop забезпечує генерацію коду на мові C з блокових діаграм та його виконання в режимі реального часу на різних апаратних платформах.

У середовищі GUIDE розроблено графічний інтерфейс програми, який наведено на рис. 3.4.

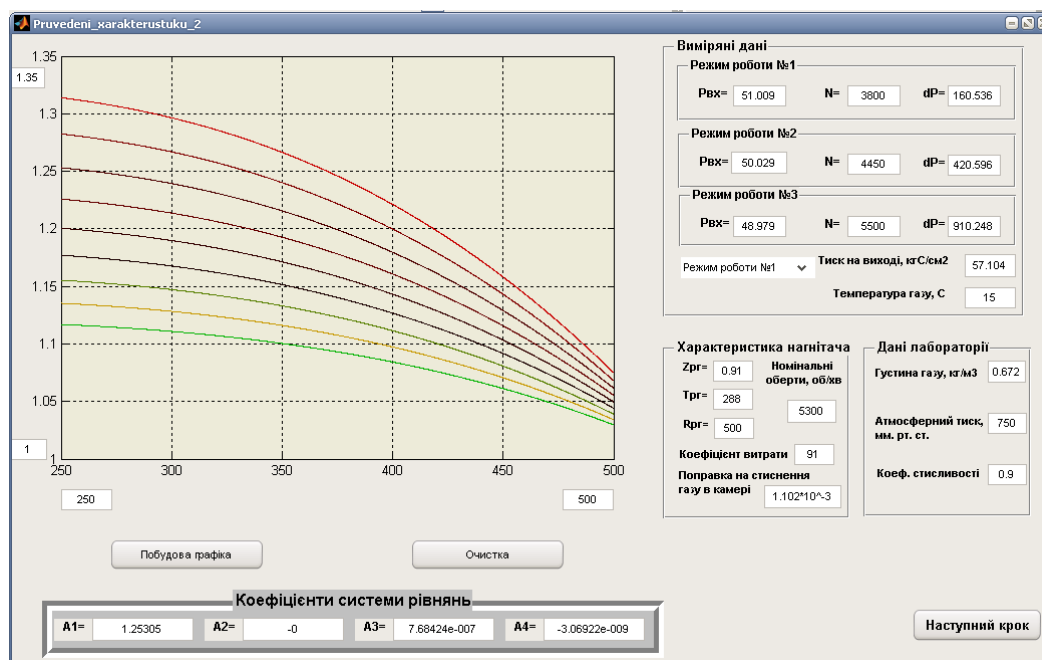


Рис. 3.4. Графічний інтерфейс користувача в середовищі GUIDE

На панелі "Measured Data" розташовані три режимні панелі для нагнітача. У кожній з них будуть відображатися дані, які отримуються з датчиків для кожного режиму роботи, зокрема: значення тиску перед камерою всмоктування - $P_{вх}$ (кгС/см³); значення фактичної частоти обертів - N (об/хв); значення перепаду тиску на камері всмоктування - dP_k (мм рт. ст.). Зчитування здійснюється шляхом вибору відповідного режиму з випадаючого списку.

Вимірні дані		
Режим роботи №1		
R _{вх} =	51.009	N= 3800 dP= 160.536
Режим роботи №2		
R _{вх} =	50.029	N= 4450 dP= 420.596
Режим роботи №3		
R _{вх} =	48.979	N= 5500 dP= 910.248
Режим роботи №1	Тиск на виході, кгС/см2	57.104
Режим роботи №2	Температура газу, С	15
Режим роботи №3		

Рис. 3.5. Вигляд панелі «Measured Data»

Для побудови графіка спершу необхідно внести вхідні дані лабораторії, включаючи густину газу (кг/м^3), коефіцієнт стисливості, атмосферний тиск (мм рт. ст.), а також параметри нагнітача, такі як коефіцієнт витрати, номінальні оберти (об/хв), поправка на стиснення газу у камері. Також потрібно врахувати умови проведення - $R_{\text{пр}}$, $Z_{\text{пр}}$, $N_{\text{пр}}$.

Для побудови графіка спершу необхідно ввести вихідні дані лабораторії, що включають густину газу (кг/м^3), коефіцієнт стисливості, атмосферний тиск (мм рт. ст.), а також параметри нагнітача: коефіцієнт витрати, номінальну частоту обертання ротора (об/хв) і поправку на стискування газу в камері. Крім того, слід врахувати умови проведення експерименту - $Z_{\text{пр}}$, $R_{\text{пр}}$, $N_{\text{пр}}$.

Характеристика нагнітача	Дані лабораторії
Z _{пр} = 0.91	Густина газу, кг/м ³ 0.672
T _{пр} = 288	Атмосферний тиск, мм. рт. ст. 750
R _{пр} = 500	Коеф. стисливості 0.9
Номінальні оберти, об/хв 5300	
Коефіцієнт витрати 91	
Поправка на стиснення газу в камері 1.102*10 ⁻³	

Рис. 3.6. Вигляд панелі "Характеристика нагнітача" а також панелі "Дані лабораторії"

Після введення всіх необхідних даних графік будується за допомогою кнопки «Побудова графіка». У результаті відображаються коефіцієнти $A_1 - A_3$ та криві фактичної приведенної характеристики високонапірного нагнітача. Для побудови нових кривих після зміни вхідних параметрів слід натиснути кнопку «Очистити» та повторно здійснити введення і зчитування даних.

Програма реалізована у вигляді М-файлу, у якому запрограмовано всі елементи графічного інтерфейсу GUIDE. Приклад структури програми наведено в Додатку Б. Файл `Privedeni_harakterustuku_2.m` може зберігатися в будь-якому каталозі, проте перед запуском необхідно вказати Matlab шлях до нього.

У середовищі Matlab виконуються математичні обчислення, результати яких відображаються графічно у вигляді залежності ступеня стиснення високонапірного нагнітача ((E)) від об'ємної приведенної витрати газу ($(Q_{пр})$).

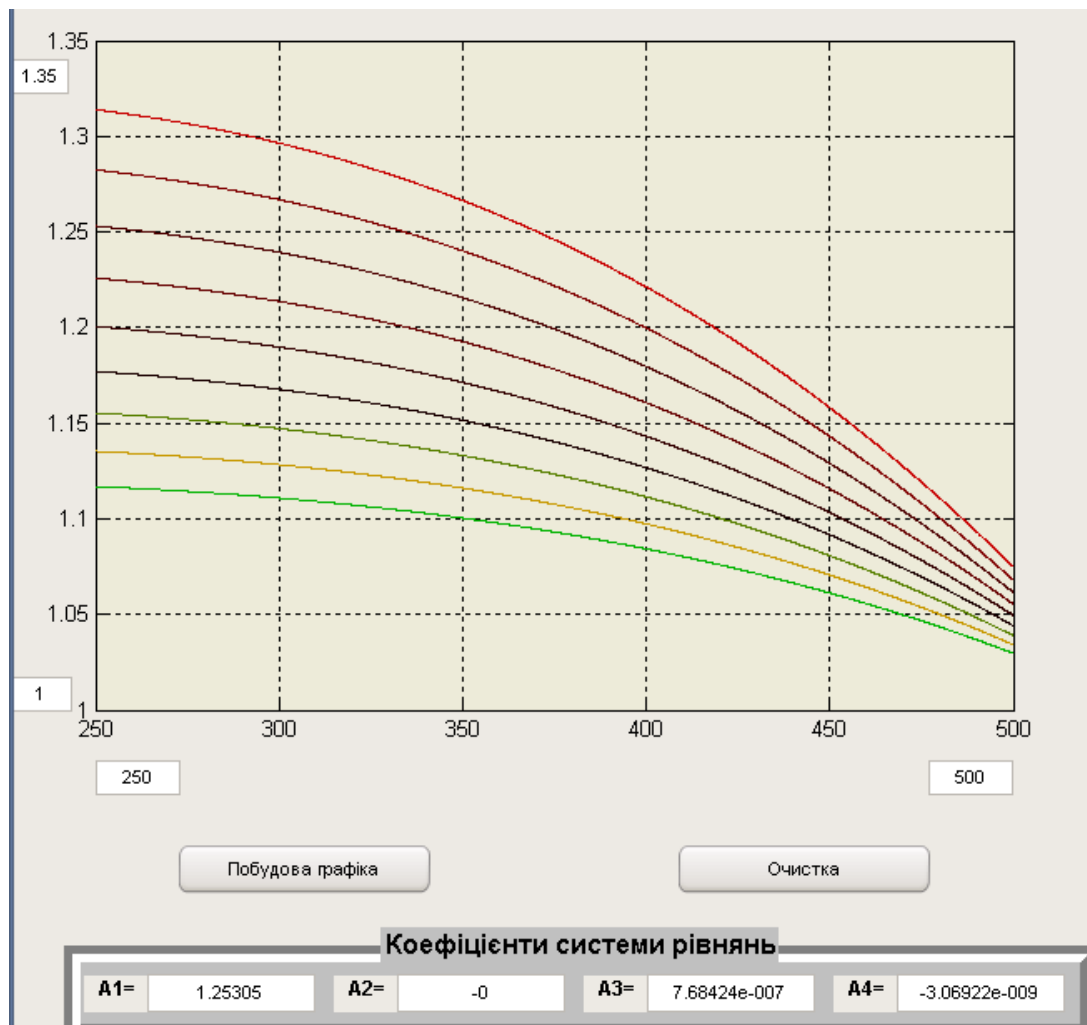


Рис 3.7. В програмі виводяться графіки реальних приведених характеристик відцентрового нагнітача

Зчитування та запис даних регулятора здійснюється за допомогою функцій `read` а також `write`. Для отримання вихідних даних регулятора застосовується команда `read(AO_C)`, де величина `AO_C` має тип «структура». Щоб безпосередньо зчитати значення, використовується покажчик: `read(AO_C.Value)`. При записі значення покажчик не потрібен, і застосовується команда `write(AO_C, 100)`.

Функція `CreateFcn` виконуватиметься тільки у випадку ініціалізації програми, в той час як деяка функція `Callback` активується у відповідь на певні події, наприклад зміну значення або натискання кнопки. Детальна інформація про використані функції представлена у довідковій системі `Matlab Help`.

Висновки до розділу 3

Розроблено алгоритмічну структуру для створення програмного забезпечення, призначеного для визначення фактичних приведених характеристик відцентрових нагнітачів.

Реалізовано систему збору та передачі даних з об'єкта на персональний комп'ютер оператора.

Створено програму для обчислення приведених характеристик відцентрових нагнітачів у середовищі `Matlab`, що забезпечує можливість відстеження зміни приведеної характеристики газоперекачувального агрегату (ГПА) у режимі реального часу.

РОЗДІЛ 4

ВИЗНАЧЕННЯ ХАРАКТЕРИСТИК ВІДЦЕНТРОВИХ НАГНІТАЧІВ

4.1. Розпізнавання, оцінка характеристик та визначення оптимальних значень функціоналів якості

Процеси, що відбуваються в об'єктах та системах взагалі, можна математично описати за допомогою конкретних операторів:

$$W_0(x, a, t),$$

де t - час;

a_i - i -й параметр;

x - вхідний сигнал;

$a(a_1, \dots, a_i, \dots, a_n)$ - значення вектора параметрів оператора.

Загальний вигляд процесу y на виході об'єкта керування (рис. 4.1) може бути охарактеризований за допомогою диференціальних, інтегральних або навіть різницевих рівнянь, ваговими, передавальними або перехідними функціями, а також частотними характеристиками.

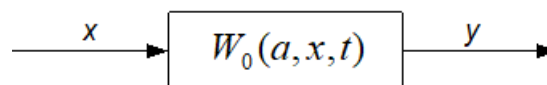


Рис. 4.1. Діаграма структури об'єкта, який піддається ідентифікації

У конкретних ситуаціях вибір конкретного формального вигляду оператора W_0 може базуватися на виборі математичної мови, враховуючи адекватність, конструктивність та зручність цієї мови.

У випадку лінійного об'єкта оператор W_0 повністю незалежний від вхідного сигналу x і має вигляд $W_0(a, t)$. Коли структура та параметри об'єкта є стаціонарними, оператор абсолютно незалежний від часу і виражається як $W_0(a, x)$. Якщо об'єкт є лінійним та стаціонарним, оператор W_0 визначається лише векторами параметрів $a(a_1, \dots, a_i, \dots, a_n)$.

Математична модель, яка розглядає процеси на виході об'єкту при відомих значеннях x та t і отримана в процесі розв'язання поставленої задачі ідентифікації, являється певною оцінкою оператора W_0 . Величина a_i , служить оцінкою i -го параметра об'єкту.

В процесі технічного розв'язання поставленої задачі ідентифікації важливо враховувати існуючі обмеження, які впливають з умов роботи об'єкта, а також враховувати загальну мету ідентифікації та досягнення якісного розв'язку. У випадку коли відомо, що на об'єкт впливають певні дії $x(t)$, які відповідають певній множині дій, в такому випадку модель $\hat{W}_0(\hat{a}, \hat{x}, t)$ повинна точно відображати ці процеси на виході об'єкту, особливо в тих випадках, коли на вхід об'єкта подаються дії зі множини Ω_x .

Якщо об'єкт ідентифікується з метою розробки системи оптимального керування, тоді отримання більш точних оцінок важливо особливо для тих параметрів, від яких залежить міра оптимальності у процесі керування. З очевидністю можна стверджувати, що необхідно здобути точні оцінки саме тих параметрів, вплив на які сильно впливає міра оптимальності керування, в той час як для інших параметрів, від яких ця міра залежить слабо або навіть не залежить зовсім, не потрібно досягати високої точності. Важливо враховувати, що процес ідентифікації в завданнях вимірювання та оптимального керування частково розрізняється.

Деталізація точних математичних моделей об'єктів управління зазвичай представляє собою завдання високої складності. У практичних застосуваннях немає необхідності визначати абсолютно точні моделі, і тому оцінки \hat{W}_0 шукаються в певному класі операторів. Отже, основне завдання ідентифікації може бути сформульоване наступним чином.

Є певний об'єкт, що характеризується оператором $W_0(a, x, t)$ та входить у певний клас операторів Ω_{W_0} . На вхід цього оператора надходять сигнали x , які належать конкретній множині сигналів Ω_x . Завданням є вибір такого оператора

$\hat{W}_0(\hat{a}, \hat{x}, t)$ із деякої множини операторів $\Omega_{\hat{W}_0}$, котрий для аналогічного класу вхідних сигналів Ω_x буде достатньо близьким до наявного оператора $W_0(a, x, t)$, точніше відображатиме процеси на виході об'єкта керування з потрібною точністю.

Зазвичай клас операторів \hat{W}_0 є більш обмеженим та вузьким у порівнянні з Ω_{W_0} . Для вивчення певних характеристик нелінійних систем здебільшого вдаються до застосування певних лінійних моделей, або, в разі вивчення складних систем, які мають високий порядок вдаються до застосування моделей, які мають нижчий порядок.

Отже, завдання ідентифікації полягає в апроксимації певного невідомого оператора $W_0 \in \Omega_{W_0}$ за допомогою оператора \hat{W}_0 . Останній зазвичай вибирається з відомого класу операторів $\Omega_{\hat{W}_0}$, враховуючи конкретний клас вхідних сигналів $x \in \Omega_x$. Міра близькості між операторами W_0 та \hat{W}_0 конструюється з урахуванням наявних вимог стосовно оцінки \hat{W}_0 . Отже, оператор $\hat{W}_0(\hat{a}, \hat{x}, t)$ буде розв'язком задачі ідентифікації.

Системи ідентифікації з використанням моделі об'єкта (рис. 4.2) знаходять широке застосування у різних галузях.

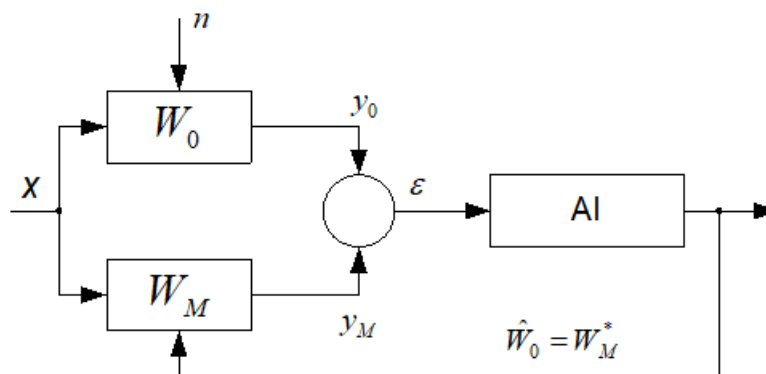


Рис. 4.2. Основна структура системи ідентифікації з моделями об'єкта керування включає алгоритм ідентифікації (AI) та оператор моделі (W_M)

Для конкретного значення W_M^* оператора моделі W_M , визначеного згідно з критерієм оптимальності, ми розглядатимемо його як оцінку оператора об'єкту W_0 та позначатимемо його як $W_M^* = \hat{W}_0$.

При відсутності перешкод процес ідентифікації може бути сформований на основі різниці між вихідними сигналами об'єкту y_0 та моделі.

4.2 Оцінка та порівняння методів ідентифікації об'єктів

В загальному відношенні, методи ідентифікації можна класифікувати відповідно до різних принципів. Зокрема, існують чотири основні групи, в які можна віднести ці методи: статистичні а також нестатистичні (детерміновані), аналітичні а також компенсаційні, градієнтні і неградієнтні, а також пошукові та безпошукові.

Аналітичні методи ідентифікації базуються на аналізі перехідних процесів, частотних а також статистичних характеристик вхідних а також вихідних сигналів. Оцінювання параметрів об'єкта здійснюється на основі співвідношень, що встановлюють зв'язок між характеристиками вихідного сигналу, параметрами об'єкта та характеристиками вхідного сигналу.

Аналітичні методи вирізняються значною швидкістю, оскільки для аналізу характеристик і розв'язання відповідних рівнянь чи систем рівнянь використовуються обчислювальні машини. Використання аналітичних методів не вимагає спостереження чи вимірювання функціоналу якості. З цієї точки зору аналітичні методи вважаються гнучкими.

Однак недоліками цих методів є необхідність математичного опису вхідних та вихідних сигналів, а також їхніх статистичних характеристик. Похибки процесу ідентифікації можуть бути зумовлені обчислювальними неточностями, спрощенням або невідповідністю математичного опису реальних сигналів, особливо у випадках нехтування нестационарними процесами та впливом зовнішніх збурень.

У компенсаційних методах ідентифікації застосовуються моделі об'єкта, які взаємодіють із ним певним чином. Вибір структури моделі здійснюється переважно евристичним шляхом - на основі наявних відомостей про об'єкт, з урахуванням вимог до точності, складності та адекватності опису.

Послідовна модель, як правило, має структуру, обернену до структури гіпотетичної моделі об'єкта, тоді як паралельна модель повторює її побудову. Параметри моделі, що забезпечують досягнення оптимального значення певного критерію якості (J), розглядаються як оцінки параметрів реального об'єкта. У деяких випадках застосовується комбінована - послідовно-паралельна - структура моделі, що поєднує переваги обох підходів.

Компенсаційні методи належать до класу методів ідентифікації із замкненим циклом, оскільки процес оцінювання параметрів здійснюється шляхом безперервного відстеження зміни міри якості (J). Ці методи спрямовані на мінімізацію похибок, зумовлених нестационарністю процесів, та на компенсацію неточностей, що виникають через спрощений або неточний математичний опис вхідних і вихідних сигналів. Важливою перевагою компенсаційних методів є відсутність потреби у строгій математичній моделі сигналів.

Швидкодія таких методів визначається початковими умовами моделі, швидкістю збіжності оптимізаційного алгоритму та параметрами самої міри якості (J), проте, як правило, вона є нижчою порівняно з аналітичними методами ідентифікації. Разом з тим, реалізація компенсаційних методів потребує наявності адекватної моделі об'єкта і спеціалізованого оптимізувального пристрою, що підвищує складність їх практичного застосування.

Статистичні методи ідентифікації, у свою чергу, ґрунтуються на використанні статистичних характеристик сигналів у процесі оцінювання параметрів системи. Міра якості (J) у таких підходах, як правило, виражається через середній ризик або середньоквадратичну похибку. Використання статистичного підходу є особливо ефективним у присутності випадкових збурень, оскільки дає змогу зменшити вплив стохастичних факторів і підвищити достовірність результатів ідентифікації.

Однак недолік статистичних методів ґрунтується на тому, що вони потребують або заздалегідь відомих статистичних характеристик, або витрат на час для їх визначення (у випадку часового усереднення). Похибки статистичних методів в основному пов'язані з неточністю визначення або знання статистичних характеристик, обумовленою, наприклад, кінцевістю інтервалів усереднення.

Нестатистичні (детерміновані) методи ідентифікації базуються на встановленні функціональних залежностей, що визначають взаємозв'язок між параметрами вхідних та вихідних сигналів і параметрами об'єкта. Такі методи застосовуються переважно у випадках, коли відсутні випадкові збурення або перешкоди, а система може бути описана детермінованими рівняннями. Основним недоліком цих підходів є їх низька завадостійкість, що обмежує сферу практичного використання в умовах реальних технологічних процесів.

Градiєнтні методи ідентифікації ґрунтуються на принципах оптимізації, де зміна параметрів моделі здійснюється у напрямку градієнта міри якості (I). Оптимізаційний процес реалізується поступово — на кожному кроці параметри коригуються відповідно до величини та напрямку градієнта з певною швидкістю збіжності. Цей підхід є ефективним у випадках, коли функція якості має строго унімодальний характер, що забезпечує стабільну та однозначну збіжність до оптимального рішення.

Основним обмеженням градієнтних методів є недосконалість вибраної міри якості, оскільки між окремими параметрами об'єкта можуть існувати кореляційні зв'язки, що зумовлює зміщення отриманих оцінок. Крім того, практична реалізація градієнтних систем ідентифікації є технічно складною, оскільки потребує спеціалізованих засобів для обчислення компонентів градієнта функціоналу (I).

На відміну від них, неградієнтні методи ґрунтуються на пошуку екстремуму функціоналу (I) без безпосереднього визначення його градієнта. Рух до оптимального значення виконується за допомогою евристичних або наближених алгоритмів, які орієнтуються лише на напрямок зменшення функціоналу якості. При цьому вибір кроку зміни параметрів здійснюється таким чином, щоб

забезпечити необхідні показники точності та швидкодії процесу ідентифікації.

Як правило, швидкодія неградієнтних методів є нижчою порівняно з градієнтними, а точність їх результатів залежить від величини кроку наближення до оптимального значення функціоналу (I). Це може спричиняти коливання оцінок навколо оптимальної точки або призводити до їх зміщення. У пошукових методах ідентифікації для досягнення оптимального режиму використовуються спеціальні пробні параметричні коливання, що дозволяють оцінити напрямок зміни функціоналу якості.

Основним недоліком таких методів є обмежена швидкодія та наявність коливального режиму поблизу точки оптимуму. Крім того, їх стійкість до шумів і перешкод є невисокою, оскільки значення міри якості (I) змінюється під впливом пробних сигналів і випадкових збурень. Вибір конкретного методу ідентифікації визначається особливостями об'єкта, умовами його функціонування, вимогами до точності, а також простотою і доцільністю практичної реалізації.

4.3. Ідентифікація та параметричне оцінювання режимів функціонування відцентрових нагнітачів у складі газоперекачувальних агрегатів

Ідентифікація помпажних характеристик відцентрових нагнітачів газоперекачувальних агрегатів із газотурбінним приводом компресорних станцій становить важливе та актуальне завдання як у науковому, так і в прикладному аспектах. Це зумовлено тим, що за результатами аналізу цих характеристик визначається технічний стан нагнітача і двигуна, встановлюється необхідність проведення профілактичного обслуговування або оновлення помпажних характеристик. До того ж, ефективне керування роботою газоперекачувального агрегату неможливе без використання у його контурах керування не лише статичних еталонних моделей, але й адаптивних, здатних враховувати зміну параметрів процесу транспортування газу.

Незважаючи на наявність значного обсягу досліджень, присвячених

загальній теорії існуючих систем, питання ідентифікації помпажних характеристик відцентрових нагнітачів газоперекачувальних агрегатів із газотурбінним приводом за умов штатної експлуатації потребує подальшого наукового опрацювання та систематизації.

Основною метою цієї роботи є розроблення процесу ідентифікації та створення методу автоматизації визначення реальних приведених витрато-напірних характеристик нагнітачів. Отримані результати можуть бути використані як науково-практична основа для вдосконалення програмного забезпечення підсистем антипомпажного регулювання та захисту.

Відповідно до міжнародної інженерної практики, регулярне визначення межі помпажу на основі індивідуальних витрато-напірних характеристик кожного нагнітача є обов'язковою процедурою, оскільки точка виникнення помпажу змінюється залежно від умов роботи агрегату.

Більш раціональним підходом до вирішення цього завдання є застосування компенсаційних методів, що передбачають використання моделі об'єкта, зв'язаної з реальним об'єктом як послідовно, так і паралельно (рис. 4.3 та 4.4).

Процеси, що відбуваються в об'єкті, можна формалізувати за допомогою оператора $G_0(a, x, t)$, де $a(a_1, a_2, \dots, a_n)$ - вектор параметрів об'єкта, x - вхідний сигнал, а t - час. Оскільки характеристики газоперекачувальних агрегатів (ГПА) мають помпажний та нелінійний характер, оператор G_0 безпосередньо залежить від вхідного сигналу (x).

Математична модель G_m , що описує вихідні процеси об'єкта при відомих значеннях (x) та (t) і сформована в результаті проведення ідентифікації, може розглядатися як апроксимаційна оцінка оператора G_0 . Аналогічно, вона забезпечує оцінку кожного окремого параметра об'єкта.

Архітектура моделі формується на основі наявних архівних даних щодо помпажних характеристик та враховує вимоги до точності і рівня складності опису. При цьому структура послідовної моделі визначається як обернена до структури гіпотетичної моделі об'єкта, тоді як паралельна модель відтворює гіпотетичну структуру об'єкта без змін.

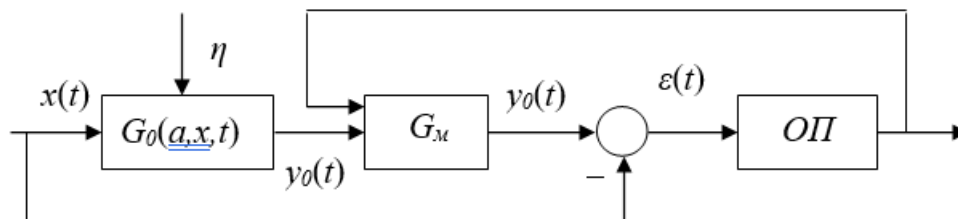


Рис. 4.3. Структура системи компенсаційної ідентифікації помпажних характеристик, яка включає в себе послідовну модель, може бути представлена в схемі: η – завади; G_M - оператор моделі; ОП - обчислювальний пристрій, котрий виконує оптимізацію функціоналу якості I ; $x(t)$ - вхідний сигнал; $y_0(t)$ - вихідний сигнал об'єкту; G_M^* - певне значення оператора G_M , котре являється оцінкою оператора об'єкта G_0

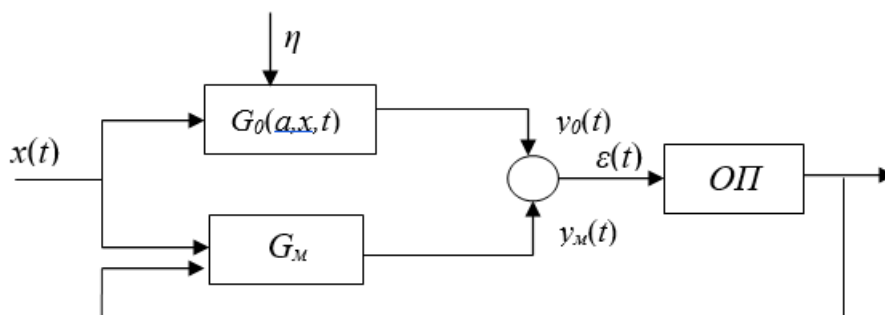


Рис. 4.4. Структура системи компенсаційної ідентифікації з паралельною моделлю

Оптимальні параметри моделі G_M , при яких показник якості ідентифікації I набуває свого максимального значення, використовуються як оцінки параметрів помпажних характеристик відцентрового газового нагнітача.

Наступним кроком є визначення передатних функцій для моделей G_0 та G_M .

У сучасній практиці розробки промислових систем автоматичного керування основний акцент робиться на використанні передавальної функції для опису динамічних характеристик об'єкта.

$$W(p) = \frac{k \cdot e^{-p\tau}}{a_2 \cdot p^2 + a_1 \cdot p + 1,0},$$

де коефіцієнт підсилення k встановлюється шляхом аналізу статичних характеристик об'єкта. Процес визначення включає в себе ідентифікацію сталої часу чистого запізнення τ , і відповідно коефіцієнтів a_1 та a_2 .

Функція передачі ВН, спільно з газотурбінним устаткуванням (для крейсерського або середнього режиму роботи), може бути представлена у наступному вигляді:

$$W(p) = \frac{k}{p^2 + 7,450 \cdot p + 13,6510}.$$

Для визначення коефіцієнта підсилення k скористаємося відомою еталонною характеристикою $E(Q_{\text{пр}})$ ВН, яка має тип 370 (рис. 1.4) за умови, що $n_{\text{пр}}=1$. Використовуючи інтерполяційний многочлен Лагранжа, наміримо криву та отримаємо рівняння статичної характеристики. Розрахунки виконаємо за допомогою програми Mathcad.

Актуальні значення параметрів для наявного рівняння $E = e(q)$:

▸ 0-а позиція: $e_0 := 1.275$; $q_0 := 250$.

▸ 1-а позиція: $e_1 := 1.248$; $q_1 := 350$.

▸ 2-а позиція: $e_2 := 1.125$; $q_2 := 500$.

Здійснимо розрахунок елементів рівняння $E = e(q)$:

$$L_0(q) := e_0 \cdot (q - q_1) \cdot (q - q_2) / (q_0 - q_1) \cdot (q_0 - q_2),$$

$$L_1(q) := e_1 \cdot (q - q_0) \cdot (q - q_2) / (q_1 - q_0) \cdot (q_1 - q_2),$$

$$L_2(q) := e_2 \cdot (q - q_0) \cdot (q - q_1) / (q_2 - q_0) \cdot (q_2 - q_1),$$

$$E(q) := L_1(q) + L_0(q) + L_2(q).$$

$$E(q) \Big|_{\text{float,4}}^{\text{expand}} \rightarrow -0,22 \cdot 10^{-7} \cdot q^2 + 0,105 \cdot 10^{-4} \cdot q + 1,150$$

Одеримо:

$$E(q) = -0,22 \cdot 10^{-7} \cdot q^2 + 0,105 \cdot 10^{-4} \cdot q + 1,150$$

Величина похибки:

$$q := \begin{pmatrix} 250 \\ 350 \\ 425 \\ 500 \end{pmatrix} \quad e(q) := \begin{pmatrix} 1.275 \\ 1.248 \\ 1.2 \\ 1.125 \end{pmatrix}. \quad Y := e(q). \quad T := q.$$

Оброблення експериментально отриманих даних:

$$k1 := \text{rows}(Y).$$

$$k1 = 4.$$

$$tk := T_{k1-1}. \quad S := \text{cspline}(T, Y). \quad q := 250, 265..500. \quad h(t) := \text{interp}(Y, t, S, T).$$

$$\Delta q := (100 \cdot (E(q) - h(q))) / h(tk).$$

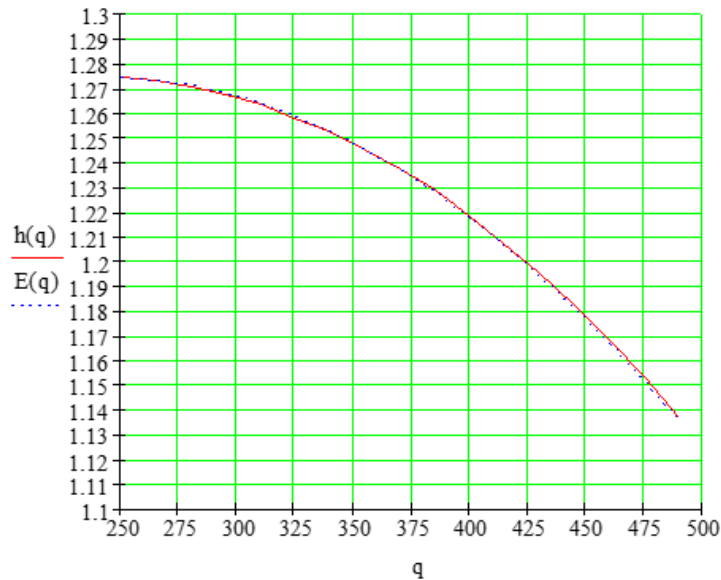


Рис. 4.5. Графік, що ілюструє відношення ступеня стиску (E) до приведеної витрати ($Q_{пр}$) на координатній площині

Для виконання порівняння відобразимо дві характеристики відцентрового нагнітача: еталонну та отриману в реальних умовах.

Передавальний коефіцієнт k визначимо як деяке значення похідної у робочій точці ($Q_{пр}=300$):

$$k = \frac{\partial \varepsilon}{\partial Q_{пр}} \Big|_{Q_{пр}=300} = (-2.200 \cdot 10^{-6} \cdot q^2 + 1.050 \cdot 10^{-3} \cdot q + 1.15)'_{Q_{пр}=300},$$

$$k = -0,27 \cdot 10^{-5}.$$

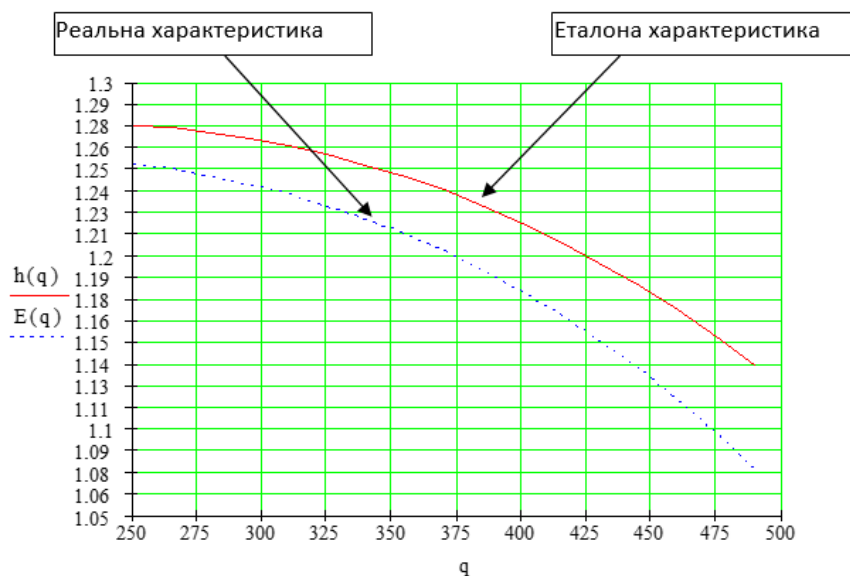


Рис. 4.6. Характеристики нагнітача, які порівнюються, включають еталонну та отриману в реальних умовах характеристики нагнітача

Таким чином, функція передачі об'єкта буде виглядати наступним чином:

$$W_0(p) = \frac{-0,27 \cdot 10^{-5}}{p^2 + 7,450p + 13,6510}$$

Функція передачі для об'єкта із наявними характеристиками матиме вигляд такий вигляд:

$$k = (-0,3069 \cdot 10^{-10} \cdot q^3 + 0,7684 \cdot 10^{-8} \cdot q + 1,253)'_{Q_{np}=300}$$

$$k = -0,3669 \cdot 10^{-5}$$

$$W_0(p) = \frac{-0,3669 \cdot 10^{-5}}{p^2 + 7,450p + 13,6510}$$

Створимо модель ідентифікації з паралельною моделлю у середовищі MATLAB.

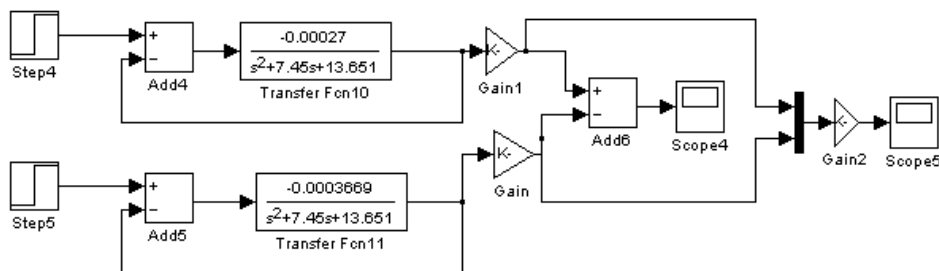


Рис. 4.7. Модель ідентифікації з паралельною структурою, реалізована у середовищі Simulink

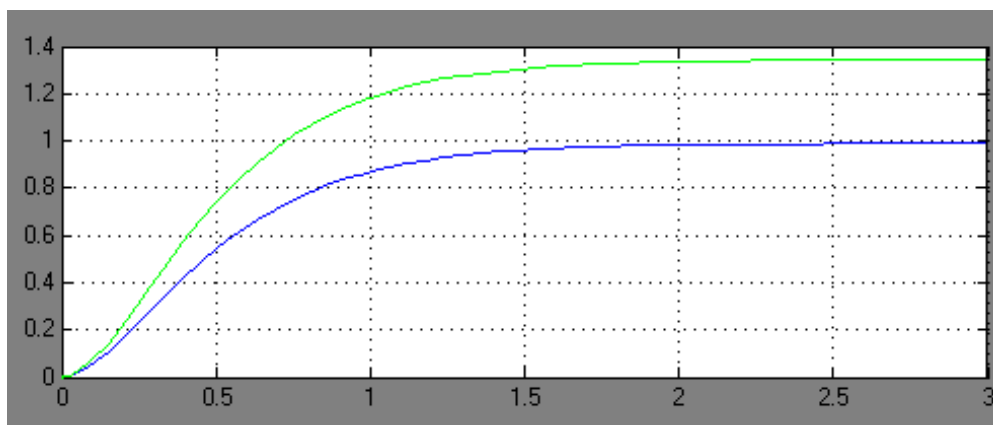


Рис. 4.8. Зміни динамічних властивостей системи залежно від різних значень коефіцієнта підсилення k

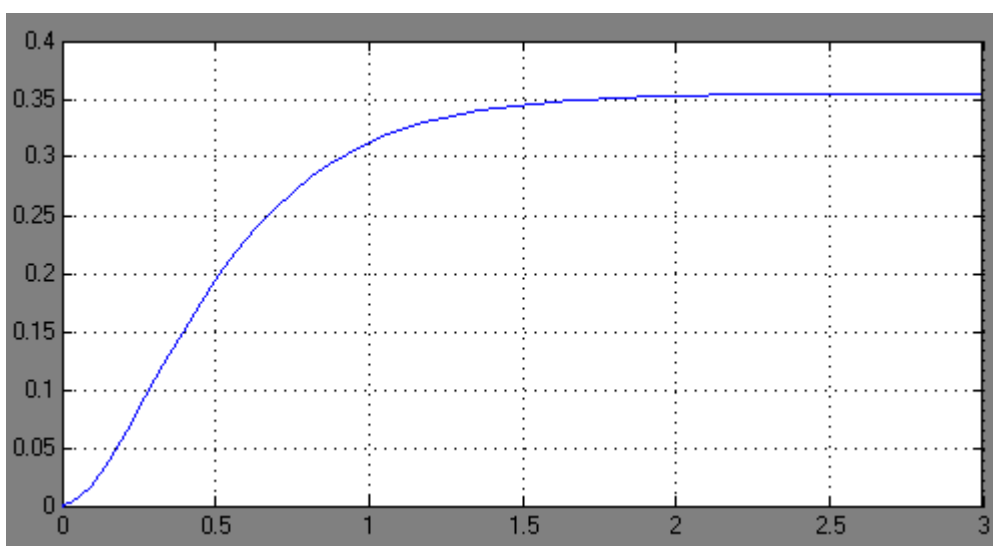


Рис. 4.9. Розмах або значення відхилень ϵ

Перевірка роботи програми здійснюється через перехід у друге вікно інтерфейсу, де відбувається безпосередня апроксимація еталонної характеристики та визначення відхилення реальної характеристики. Для доступу до цього вікна (рис. 4.10) слід натиснути кнопку «Наступний крок» (рис. 3.4). У панелі «Еталонна характеристика» необхідно задати три точки графіка характеристики для її апроксимації та ввести передавальну функцію об'єкта. Після натискання кнопки «Побудова графіка» на екрані будуть відображені дві приведені характеристики: еталонна – у верхній частині, та реальна – у нижній (рис. 4.11). Повторне натискання кнопки «Побудова графіка» дозволяє визначити величину

відхилення ε (рис. 4.12).

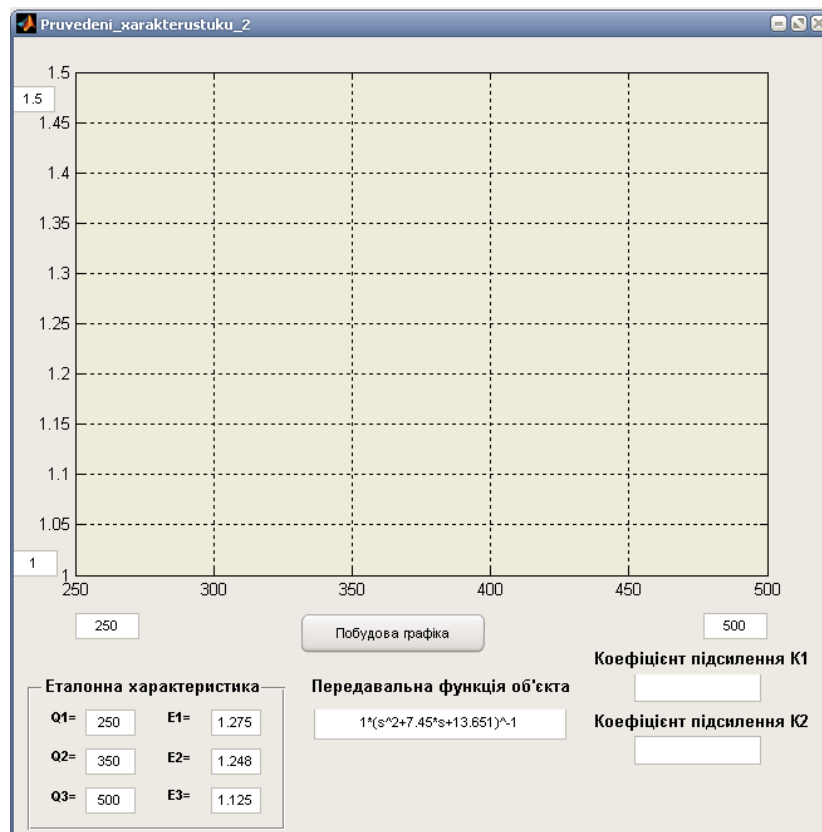


Рис. 4.10. Вигляд вікна №2

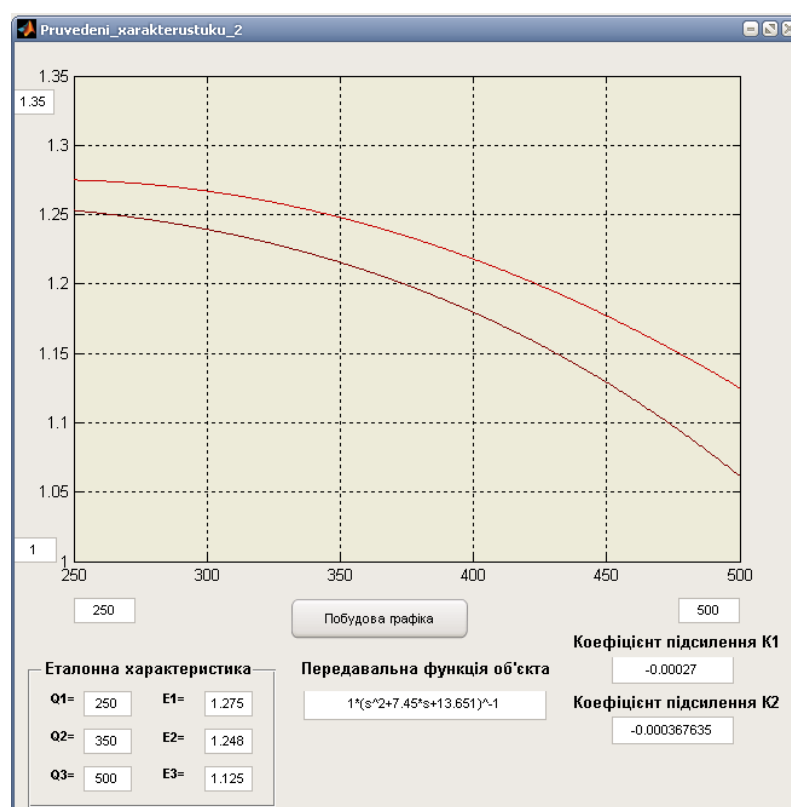


Рис. 4.11. Результати визначених приведених характеристик

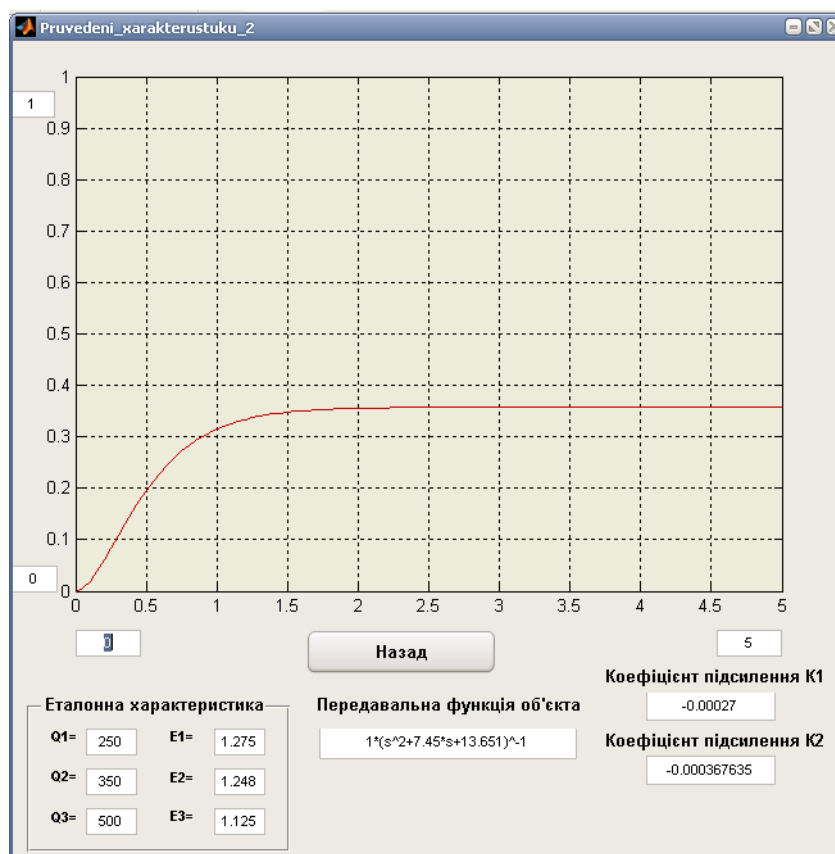


Рис. 4.12. Вікно відображення відхилення ε

Висновки до розділу 4

Метод ідентифікації характеристик відцентрових нагнітачів газоперекачувальних агрегатів із застосуванням паралельної моделі був розроблений та досліджений. Запропонований підхід забезпечує автоматизоване визначення приведених характеристик нагнітача та обчислення величини відхилення ε . Отримані значення ε використовуються для корекції антипомпажної системи або для безпосереднього регулювання байпасного клапана з метою контролю витрати, що дозволяє ефективно запобігати виникненню аварійних ситуацій.

ЗАГАЛЬНІ ВИСНОВКИ

У межах магістерської роботи проведено дослідження, присвячене ідентифікації та автоматизації процесу визначення реальних приведених характеристик відцентрових нагнітачів. Основні результати дослідження включають:

1. На основі аналізу структури компресорної станції, конструкції відцентрових нагнітачів газоперекачувальних агрегатів та їх приведених характеристик обґрунтовано наукову та практичну актуальність роботи.

2. Удосконалено методику розрахунку об'ємної витрати газу через перепад тиску в конфузори відцентрового нагнітача, що дозволило визначити нормовані значення $Q_{пр}$ і $E_{пр}$. На основі отриманих даних були побудовані графіки актуальних нормованих параметрів відцентрових нагнітачів ГПА в режимі реального часу.

3. Розроблено алгоритм створення програмного забезпечення для автоматизованого визначення фактичних нормованих параметрів відцентрового нагнітача, що забезпечує організацію збору даних з об'єкта та їх передачу на персональний комп'ютер оператора.

4. Удосконалено методику ідентифікації характеристик відцентрових нагнітачів газоперекачувальних агрегатів з газотурбінним приводом, що є важливим науково-практичним завданням. Це зумовлено тим, що на основі отриманих характеристик приймаються рішення щодо необхідності профілактичного обслуговування двигуна та нагнітача або актуалізації їхніх характеристик. Використання визначеного відхилення ε дозволяє коригувати антипомпажну систему або безпосередньо керувати байпасним краном для регулювання витрати газу, що сприяє запобіганню аварійних режимів роботи агрегату.

5. Створено програмний комплекс у середовищі MATLAB для автоматизованого визначення нормованих характеристик відцентрових нагнітачів газоперекачувальних агрегатів. На підставі отриманих даних здійснюється оцінка

технічного стану двигуна та нагнітача, а також визначається потреба у корекції програмного забезпечення підсистеми антипомпажного регулювання та захисту агрегату.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

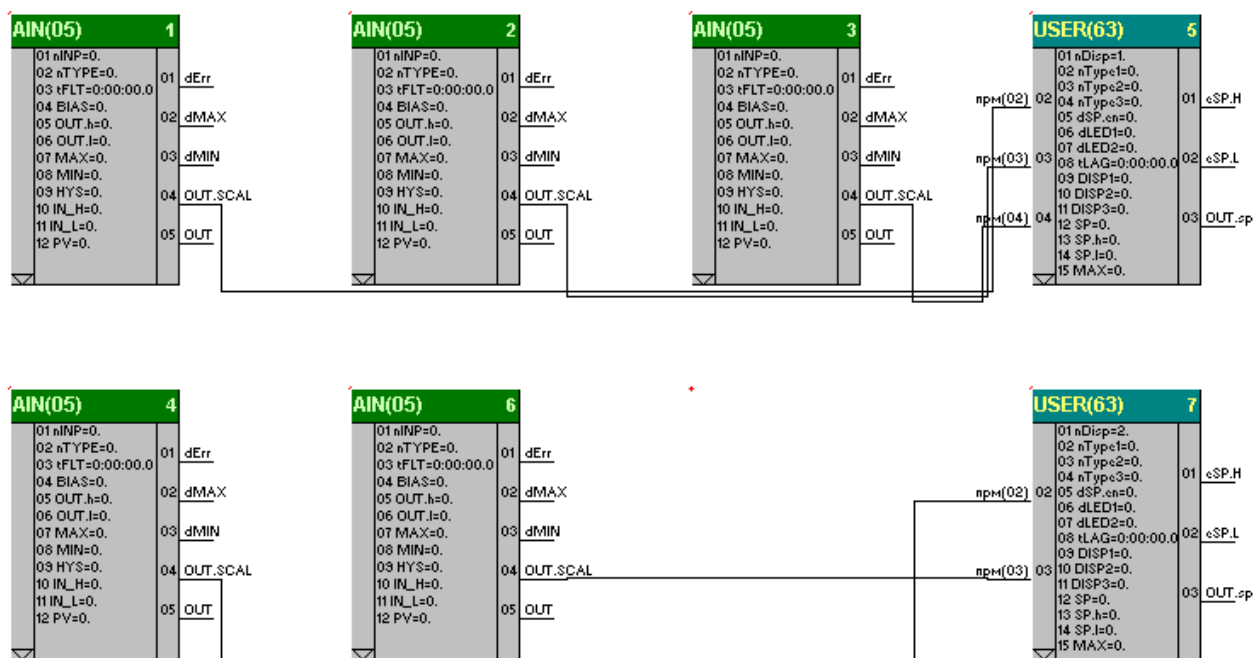
1. Тимків Д. Ф., Грудз В. Я., Ковалко М. П., Михалків В. Б., Ковалко О. М., Шлапак Л. С. Трубопровідний транспорт газу. К.: Агенство із раціонального використання енергії та екології, 2002. 600 с.
2. Горбійчук, М. І. Теорія автоматичного керування. Лінійні системи. [Текст]: конспект лекцій / М. І. Горбійчук. – Івано-Франківськ: ІФНТУНГ, 2024. – 159 с.
3. Семенцов, Г. Н. Теорія автоматичного управління (нелінійні та дискретні системи) [Текст]: навч. посіб. / Г. Н. Семенцов, І. І. Чигур. – Івано-Франківськ: ІФНТУНГ, 2014. – 96 с.
4. Горбійчук, М. І. Теорія автоматичного керування [Текст]: метод. вказівки до виконання курс. роботи / М. І. Горбійчук. – Івано-Франківськ: ІФНТУНГ, 2021. – 57 с.
5. Руднік А. А., Розгонюк В. В., Григіль М. А., Коломєєв В. М.. Довідник працівника газотранспортного підприємства. К.: Росток, 2001. 1091 с.
6. Автоматизація виробничих процесів в енергетиці [Текст]: навч. посіб. / Г. Н. Семенцов, Я. Р. Когуч, М. В. Шавранський, М. М. Дранчук. – Івано-Франківськ : ІФНТУНГ, 2003. – 372 с.
7. «Контролер Малоканалний Багатофункціональний Мікропроцес-сміттевий МІК-51» Посібник з експлуатації ПРМК.421457.005 РЕ1,2. Івано-Франківськ, 2006. 87 с.
8. "Блоки перетворення сигналів інтерфейсів БПІ-52 (USB2.0 RS-485)" Посібник з експлуатації ПРМК.426449.003 РЕ. Івано-Франківськ, 2006. 17 с.
9. «MODBUS OPC Server Версія 1.0.1» Посібник з експлуатації ПРМК.426000.004 РЕ. Івано-Франківськ, 2006. 8 с.
10. Макаров А., Продовіков С., Черніков А., Бунін В. Досвід автоматизації складних промислових об'єктів на прикладі газокомпресорних станцій. Системна інтеграція. Нафтогазова промисловість, 1999. №2. С.16-25.

11. Остапенко Ю. О. Ідентифікація та моделювання технологічних об'єктів керування. К.: Задруга, 1999. 424 с.
12. Семенцов Г. Н. Теорія автоматичного керування. Івано-Франківськ: Факел. 1999. 611 с.
13. Костюк В. І., Кіху А. Г., Краскевич В. Є., Шпіт С. В., Сильвестров А. Н. Адаптивні системи ідентифікації. К.: Техніка, 1975. 288с.
14. Гостєв В. І. Синтез нечітких регуляторів систем автоматичного управління. К.: Радіоаматор, 2003. 510 с.
15. Кондрат Р. М., Яремійчук Р.С., Бойко В. С. Довідник з нафтогазової справи. К.: Львів, 1996. 620 с.
16. Давиденко Л. І., Бляут Ю. Є. Давиденко Л. І. Автоматична ідентифікація реальних приведених витратно-напірних характеристик відцентрових нагнітачів. Нафтогазова енергетика. 2012. № 1. С. 60-69.

ДОДАТКИ

Додаток А

FBD-програма Альфа для контролера МІК-51, в якій 5 вхідних сигналів з'єднуються з панелью користувача USER(63). Функціональний блок USER(63) застосовується для індикації та зміни сигналів, які потрібно бачити користувачеві одночасно на передній панелі контролера.



Додаток Б

Код програми (M-file) визначення приведених характеристик відцентрових нагнітачів газоперекачувальних агрегатів створена в програмному пакеті MATLAB.

```
function varargout = Pruverdeni_xarakterustuku_2(varargin)
%PRUVEDENI_XARAKTERUSTUKU_2 M-file for Pruverdeni_xarakterustuku_2.fig
%   PRUVEDENI_XARAKTERUSTUKU_2, by itself, creates a new
PRUVEDENI_XARAKTERUSTUKU_2 or raises the existing
%   singleton*.
%
%   H = PRUVEDENI_XARAKTERUSTUKU_2 returns the handle to a new
PRUVEDENI_XARAKTERUSTUKU_2 or the handle to
%   the existing singleton*.
%
%   PRUVEDENI_XARAKTERUSTUKU_2('Property','Value',...) creates a new
PRUVEDENI_XARAKTERUSTUKU_2 using the
%   given property value pairs. Unrecognized properties are passed via
%   varargin to Pruverdeni_xarakterustuku_2_OpeningFcn. This calling syntax
produces a
%   PRUVEDENI_XARAKTERUSTUKU_2('CALLBACK') and
PRUVEDENI_XARAKTERUSTUKU_2('CALLBACK',hObject,...) call the
%   local function named CALLBACK in PRUVEDENI_XARAKTERUSTUKU_2.M with the
given input
%   arguments.
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Begin initialization code - DO NOT EDIT
gui_Singleton = 0;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Pruverdeni_xarakterustuku_2_OpeningFcn, ...
                  'gui_OutputFcn',  @Pruverdeni_xarakterustuku_2_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

function Pruverdeni_xarakterustuku_2_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
guidata(hObject, handles);
global serv grp_51 Pvx Pvux dP n T

clc;
serv=opcda('localhost','Microl.OPC.ModBus Server');
connect(serv);
serv;
grp_51=addgroup(serv);
grp_51.UpdateRate=0.1;
```

```

Pvx=additem(grp_51, 'Pvx');
dP=additem(grp_51, 'dP');
n=additem(grp_51, 'n');
T=additem(grp_51, 'T');
Pvux=additem(grp_51, 'Pvux');
read(Pvx);
read(dP);
read(n);
read(T);
read(Pvux);
set(handles.Q1, 'Visible', 'off');
set(handles.Q2, 'Visible', 'off');
set(handles.Q3, 'Visible', 'off');
set(handles.E1, 'Visible', 'off');
set(handles.E2, 'Visible', 'off');
set(handles.E3, 'Visible', 'off');
set(handles.text80, 'Visible', 'off');
set(handles.text81, 'Visible', 'off');
set(handles.text82, 'Visible', 'off');
set(handles.text84, 'Visible', 'off');
set(handles.text86, 'Visible', 'off');
set(handles.text87, 'Visible', 'off');
set(handles.Per, 'Visible', 'off');
set(handles.K1, 'Visible', 'off');
set(handles.K2, 'Visible', 'off');
set(handles.uipanel21, 'Visible', 'off');
set(handles.pobudov, 'Visible', 'off');

function varargout = Pruvedeni_xarakterustuku_2_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function Reg_1_P_Callback(hObject, eventdata, handles)

function Reg_1_P_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function Reg_2_P_Callback(hObject, eventdata, handles)

function Reg_2_P_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function Reg_3_P_Callback(hObject, eventdata, handles)

function Reg_3_P_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

```

function Vidn_gyst_Callback(hObject, eventdata, handles)

function Vidn_gyst_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Koef_stusl_Callback(hObject, eventdata, handles)

function Koef_stusl_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Temp_vxid_Callback(hObject, eventdata, handles)

function Temp_vxid_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Chast_obert_Callback(hObject, eventdata, handles)

function Chast_obert_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Obem_vutrat_Callback(hObject, eventdata, handles)

function Obem_vutrat_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Tusk_vux_Callback(hObject, eventdata, handles)

function Tusk_vux_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```

        set(hObject,'BackgroundColor','white');
end

function Zpr_Callback(hObject, eventdata, handles)

function Zpr_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Rpr_Callback(hObject, eventdata, handles)

function Rpr_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Tpr_Callback(hObject, eventdata, handles)

function Tpr_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function N_nom_Callback(hObject, eventdata, handles)

function N_nom_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Graf_pbt_Callback(hObject, eventdata, handles)
global A Qpr
Gn=str2num(get(handles.Vidn_gyst,'String'));
Z=str2num(get(handles.Koef_stusl,'String'));
Tvx=str2num(get(handles.Temp_vxid,'String'));
Pa=str2num(get(handles.Tusk_atm,'String'));
P(1)=str2num(get(handles.Reg_1_P,'String'));
P(2)=str2num(get(handles.Reg_2_P,'String'));
P(3)=str2num(get(handles.Reg_3_P,'String'));
dP(1)=str2num(get(handles.Reg_1_dP,'String'));
dP(2)=str2num(get(handles.Reg_2_dP,'String'));
dP(3)=str2num(get(handles.Reg_3_dP,'String'));
Pvux(1)=str2num(get(handles.Tusk_vux,'String'));
Pvux(2)=str2num(get(handles.Tusk_vux,'String'));
Pvux(3)=str2num(get(handles.Tusk_vux,'String'));
N(1)=str2num(get(handles.Chast_obert_1,'String'));
N(2)=str2num(get(handles.Chast_obert_2,'String'));

```

```

N(3)=str2num(get(handles.Chast_obert_3,'String'));
Zpr=str2num(get(handles.Zpr,'String'));
Tpr=str2num(get(handles.Tpr,'String'));
Rpr=str2num(get(handles.Rpr,'String'));
Nn=str2num(get(handles.N_nom,'String'));
A=str2num(get(handles.A,'String'));
L=str2num(get(handles.L,'String'));
set(handles.X_start,'String',250);
set(handles.X_end,'String',500);
set(handles.Y_start,'String',1);
set(handles.Y_end,'String',1.35);
set(handles.axes1,'XLim',[fix(str2num(get(handles.X_start,'String')))...
    fix(str2num(get(handles.X_end,'String'))]);
set(handles.axes1,'YLim',[str2double(get(handles.Y_start,'String'))...
    str2double(get(handles.Y_end,'String'))]);
T=273+Tvx;
Pa=Pa/735.5;
Pvx=Pa+P;
d=Gn/1.206;
R=288/d;
Gyst=Gn/1.206;
G=(Pvx*10^5)/(Z*R*T);
l(1)=1-L*dP(1)/Pvx(1);
l(2)=1-L*dP(2)/Pvx(2);
l(3)=1-L*dP(3)/Pvx(3);
Qv(1)=A*l(1)*sqrt(dP(1)/G(1));
Qv(2)=A*l(2)*sqrt(dP(2)/G(2));
Qv(3)=A*l(3)*sqrt(dP(3)/G(3));
Npr=(N/Nn)*sqrt((Zpr*Rpr*Tpr)/(Z*T*R));
Nv(1)=Nn/N(1);
Nv(2)=Nn/N(2);
Nv(3)=Nn/N(3);
E(1)=Pvux(1)/P(1);
E(2)=Pvux(2)/P(2);
E(3)=Pvux(3)/P(3);
Qpr(1)=Qv(1)*Nv(1);
Qpr(2)=Qv(2)*Nv(2);
Qpr(3)=Qv(3)*Nv(3);
Epr(1)=E(1)^(Npr(1)^-2);
Epr(2)=E(2)^(Npr(2)^-2);
Epr(3)=E(3)^(Npr(3)^-2);
X=[1,Qpr(2),Qpr(2)^2,Qpr(2)^3; 1,Qpr(1),Qpr(1)^2,Qpr(1)^3;
1,Qpr(3),Qpr(3)^2,Qpr(3)^3; 0,1,0,0];
B=[Epr(2);Epr(1);Epr(3); 0];
format short e
A=X\B;
set(handles.A1,'String',A(1));
set(handles.A2,'String',A(2));
set(handles.A3,'String',A(3));
set(handles.A4,'String',A(4));
set(handles.Graf_pbt,'Enable','on');
set(handles.Clear_pbt,'Enable','on');
a=str2double(get(handles.X_start,'String'));
b=str2double(get(handles.X_end,'String'));
Qpr=str2double(get(handles.X_start,'String'));
while Qpr<str2double(get(handles.X_end,'String'))
    y=(A(1)+A(2)*Qpr+A(3)*Qpr^2+A(4)*Qpr^3);
    y1=y^1.21;
    y2=y^1.1025;
    y3=y^0.9025;
    y5=y^0.81;
    y6=y^0.7225;
    y7=y^0.64;
    y8=y^0.5625;

```

```

    y9=y^0.49;
    if y>1
        plot(Qpr,y1,'Parent',handles.axes1,'EraseMode','none','Color',[0.785 0
0]);
        plot(Qpr,y2,'Parent',handles.axes1,'EraseMode','none','Color',[0.5 0 0]);
        plot(Qpr,y3,'Parent',handles.axes1,'EraseMode','none','Color',[0.4 0 0]);
        plot(Qpr,y,'Parent',handles.axes1,'EraseMode','none','Color',[0.3 0 0]);
        plot(Qpr,y5,'Parent',handles.axes1,'EraseMode','none','Color',[0.2 0 0]);
        plot(Qpr,y6,'Parent',handles.axes1,'EraseMode','none','Color',[0.1 0 0]);
        plot(Qpr,y7,'Parent',handles.axes1,'EraseMode','none','Color',[0.345 0.5
0]);
        plot(Qpr,y8,'Parent',handles.axes1,'EraseMode','none','Color',[0.785 0.6
0]);
        plot(Qpr,y9,'Parent',handles.axes1,'EraseMode','none','Color',[0 0.7 0]);
        Qpr=Qpr+0.1;
        drawnow;
    else
        Qpr=Qpr+0.1;
    end
end

function Clear_pbt_Callback(hObject, eventdata, handles)
global Qpr

set(handles.axes1,'NextPlot','replace');
plot(0,0,'Parent',handles.axes1);
set(handles.axes1,'NextPlot','add','XGrid','on','YGrid','on','Color',[0.931 0.92
0.851]);
Qpr=str2double(get(handles.X_end,'String'));
Privedeni_xarakterustuku_2('X_start_Callback',gcbo,[],guidata(gcbo));
Privedeni_xarakterustuku_2('Y_start_Callback',gcbo,[],guidata(gcbo));

function Regum_menu_Callback(hObject, eventdata, handles)
global Pvx Pvux dP n T num
num=get(hObject,'Value');
switch num
    case 1
        set(handles.Reg_1_P,'String',Pvx.Value);
        set(handles.Reg_1_dP,'String',dP.Value);
        set(handles.Chast_obert_1,'String',n.Value);
    case 2
        set(handles.Reg_2_P,'String',Pvx.Value);
        set(handles.Reg_2_dP,'String',dP.Value);
        set(handles.Chast_obert_2,'String',n.Value);
    case 3
        set(handles.Reg_3_P,'String',Pvx.Value);
        set(handles.Reg_3_dP,'String',dP.Value);
        set(handles.Chast_obert_3,'String',n.Value);
        set(handles.Temp_vxid,'String',T.Value);
        set(handles.Tusk_vux,'String',Pvux.Value);
end

function Regum_menu_CreateFcn(hObject, eventdata, handles)

% Hint: popmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function A1_Callback(hObject, eventdata, handles)

```

```

function A1_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function A2_Callback(hObject, eventdata, handles)

function A2_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function A3_Callback(hObject, eventdata, handles)

function A3_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function A4_Callback(hObject, eventdata, handles)

function A4_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Zmax_A_Callback(hObject, eventdata, handles)
global A
Gn=str2num(get(handles.Vidn_gyst,'String'));
Z=str2num(get(handles.Koef_stusl,'String'));
Tvx=str2num(get(handles.Temp_vxid,'String'));
Pa=str2num(get(handles.Tusk_atm,'String'));
P(1)=str2num(get(handles.Reg_1_P,'String'));
P(2)=str2num(get(handles.Reg_2_P,'String'));
P(3)=str2num(get(handles.Reg_3_P,'String'));
dP(1)=str2num(get(handles.Reg_1_dP,'String'));
dP(2)=str2num(get(handles.Reg_2_dP,'String'));
dP(3)=str2num(get(handles.Reg_3_dP,'String'));
Pvux(1)=str2num(get(handles.Tusk_vux,'String'));
Pvux(2)=str2num(get(handles.Tusk_vux,'String'));
Pvux(3)=str2num(get(handles.Tusk_vux,'String'));
N(1)=str2num(get(handles.Chast_obert_1,'String'));
N(2)=str2num(get(handles.Chast_obert_2,'String'));
N(3)=str2num(get(handles.Chast_obert_3,'String'));
Zpr=str2num(get(handles.Zpr,'String'));
Tpr=str2num(get(handles.Tpr,'String'));
Rpr=str2num(get(handles.Rpr,'String'));
Nn=str2num(get(handles.N_nom,'String'));

```

```

A=str2num(get(handles.A, 'String'));
L=str2num(get(handles.L, 'String'));
T=273+Tvx;
Pa=Pa/735.5;
Pvx=Pa+P;
d=Gn/1.206;
R=288/d;
Gyst=Gn/1.206;
G=(Pvx*105)/(Z*R*T);
l(1)=1-L*dP(1)/Pvx(1);
l(2)=1-L*dP(2)/Pvx(2);
l(3)=1-L*dP(3)/Pvx(3);
Qv(1)=A*l(1)*sqrt(dP(1)/G(1));
Qv(2)=A*l(2)*sqrt(dP(2)/G(2));
Qv(3)=A*l(3)*sqrt(dP(3)/G(3));
Npr=(N/Nn)*sqrt((Zpr*Rpr*Tpr)/(Z*T*R));
Nv(1)=Nn/N(1);
Nv(2)=Nn/N(2);
Nv(3)=Nn/N(3);
E(1)=Pvux(1)/P(1);
E(2)=Pvux(2)/P(2);
E(3)=Pvux(3)/P(3);
Qpr(1)=Qv(1)*Nv(1);
Qpr(2)=Qv(2)*Nv(2);
Qpr(3)=Qv(3)*Nv(3);
Epr(1)=E(1)^(Npr(1)^-2);
Epr(2)=E(2)^(Npr(2)^-2);
Epr(3)=E(3)^(Npr(3)^-2);
X=[1, Qpr(2), Qpr(2)^2, Qpr(2)^3; 1, Qpr(1), Qpr(1)^2, Qpr(1)^3;
1, Qpr(3), Qpr(3)^2, Qpr(3)^3; 0, 1, 0, 0];
B=[Epr(2); Epr(1); Epr(3); 0];
format short e;
A=X\B;
set(handles.A1, 'String', A(1));
set(handles.A2, 'String', A(2));
set(handles.A3, 'String', A(3));
set(handles.A4, 'String', A(4));
set(handles.Graf_pbt, 'Enable', 'on');
set(handles.Clear_pbt, 'Enable', 'on');

function X_start_Callback(hObject, eventdata, handles)
% hObject    handle to X_start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of X_start as text
%         str2double(get(hObject, 'String')) returns contents of X_start as a double
set(handles.axes1, 'XLim', [fix(str2num(get(handles.X_start, 'String')))...
fix(str2num(get(handles.X_end, 'String')))]);

% --- Executes during object creation, after setting all properties.
function X_start_CreateFcn(hObject, eventdata, handles)
% hObject    handle to X_start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function X_end_Callback(hObject, eventdata, handles)

```

```

% hObject      handle to X_end (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of X_end as text
%          str2double(get(hObject,'String')) returns contents of X_end as a double
set(handles.axes1,'XLim',[fix(str2num(get(handles.X_start,'String')))...
    fix(str2num(get(handles.X_end,'String')))]);

% --- Executes during object creation, after setting all properties.
function X_end_CreateFcn(hObject, eventdata, handles)
% hObject      handle to X_end (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Y_end_Callback(hObject, eventdata, handles)
% hObject      handle to Y_end (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Y_end as text
%          str2double(get(hObject,'String')) returns contents of Y_end as a double
set(handles.axes1,'YLim',[str2num(get(handles.Y_start,'String'))...
    str2num(get(handles.Y_end,'String'))]);

% --- Executes during object creation, after setting all properties.
function Y_end_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Y_end (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Y_start_Callback(hObject, eventdata, handles)
% hObject      handle to Y_start (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Y_start as text
%          str2double(get(hObject,'String')) returns contents of Y_start as a double
set(handles.axes1,'YLim',[str2num(get(handles.Y_start,'String'))...
    str2num(get(handles.Y_end,'String'))]);

% --- Executes during object creation, after setting all properties.
function Y_start_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Y_start (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Chast_obert_1_Callback(hObject, eventdata, handles)
% hObject    handle to Chast_obert_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Chast_obert_1 as text
%         str2double(get(hObject,'String')) returns contents of Chast_obert_1 as a
double

% --- Executes during object creation, after setting all properties.
function Chast_obert_1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Chast_obert_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Reg_1_dP_Callback(hObject, eventdata, handles)
% hObject    handle to Reg_1_dP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Reg_1_dP as text
%         str2double(get(hObject,'String')) returns contents of Reg_1_dP as a
double

% --- Executes during object creation, after setting all properties.
function Reg_1_dP_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Reg_1_dP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Chast_obert_2_Callback(hObject, eventdata, handles)
% hObject    handle to Chast_obert_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Chast_obert_2 as text
%         str2double(get(hObject,'String')) returns contents of Chast_obert_2 as a
double

% --- Executes during object creation, after setting all properties.
function Chast_obert_2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Chast_obert_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Reg_2_dP_Callback(hObject, eventdata, handles)
% hObject     handle to Reg_2_dP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Reg_2_dP as text
%     str2double(get(hObject,'String')) returns contents of Reg_2_dP as a
double

% --- Executes during object creation, after setting all properties.
function Reg_2_dP_CreateFcn(hObject, eventdata, handles)
% hObject     handle to Reg_2_dP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Chast_obert_3_Callback(hObject, eventdata, handles)
% hObject     handle to Chast_obert_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Chast_obert_3 as text
%     str2double(get(hObject,'String')) returns contents of Chast_obert_3 as a
double

% --- Executes during object creation, after setting all properties.
function Chast_obert_3_CreateFcn(hObject, eventdata, handles)
% hObject     handle to Chast_obert_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Reg_3_dP_Callback(hObject, eventdata, handles)
% hObject     handle to Reg_3_dP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Reg_3_dP as text
%     str2double(get(hObject,'String')) returns contents of Reg_3_dP as a
double

% --- Executes during object creation, after setting all properties.
function Reg_3_dP_CreateFcn(hObject, eventdata, handles)
% hObject     handle to Reg_3_dP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function A_Callback(hObject, eventdata, handles)
% hObject      handle to A (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of A as text
%       str2double(get(hObject,'String')) returns contents of A as a double

% --- Executes during object creation, after setting all properties.
function A_CreateFcn(hObject, eventdata, handles)
% hObject      handle to A (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Tusk_atm_Callback(hObject, eventdata, handles)
% hObject      handle to Tusk_atm (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Tusk_atm as text
%       str2double(get(hObject,'String')) returns contents of Tusk_atm as a
double

% --- Executes during object creation, after setting all properties.
function Tusk_atm_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Tusk_atm (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Next.
function Next_Callback(hObject, eventdata, handles)
global A B
% hObject      handle to Next (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
set(handles.uipanel1,'Visible','off');
set(handles.uipanel3,'Visible','off');
set(handles.uipanel9,'Visible','off');
set(handles.Graf_pbt,'Visible','off');
set(handles.Clear_pbt,'Visible','off');
set(handles.uipanel5,'Visible','off');

```

```

set(handles.text21,'Visible','off');
set(handles.text22,'Visible','off');
set(handles.text23,'Visible','off');
set(handles.text24,'Visible','off');
set(handles.A1,'Visible','off');
set(handles.A2,'Visible','off');
set(handles.A3,'Visible','off');
set(handles.A4,'Visible','off');
set(handles.Q1,'Visible','on');
set(handles.Q2,'Visible','on');
set(handles.Q3,'Visible','on');
set(handles.E1,'Visible','on');
set(handles.E2,'Visible','on');
set(handles.E3,'Visible','on');
set(handles.text80,'Visible','on');
set(handles.text81,'Visible','on');
set(handles.text82,'Visible','on');
set(handles.text84,'Visible','on');
set(handles.text86,'Visible','on');
set(handles.text87,'Visible','on');
set(handles.text88,'Visible','on');
set(handles.text89,'Visible','on');
set(handles.text90,'Visible','on');
set(handles.Per,'Visible','on');
set(handles.K1,'Visible','on');
set(handles.K2,'Visible','on');
set(handles.pobudov,'Visible','on');
set(handles.pobudov,'Visible','on');
set(handles.uipanel21,'Visible','on');
set(handles.figure1,'Position',[71.8 5 130 49]);
set(handles.axes1,'NextPlot','replace');
plot(0,0,'Parent',handles.axes1);
set(handles.axes1,'NextPlot','add','XGrid','on','YGrid','on','Color',[0.931 0.92
0.851]);
Qpr=str2double(get(handles.X_end,'String'));
Pravedeni_xarakterustuku_2('X_start_Callback',gcbo,[],guidata(gcbo));
Pravedeni_xarakterustuku_2('Y_start_Callback',gcbo,[],guidata(gcbo));

function L_Callback(hObject, eventdata, handles)
% hObject    handle to L (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of L as text
%        str2double(get(hObject,'String')) returns contents of L as a double

% --- Executes during object creation, after setting all properties.
function L_CreateFcn(hObject, eventdata, handles)
% hObject    handle to L (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function figure1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% --- Executes when figure1 is resized.
function figure1_ResizeFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

function rivnjanja_Callback(hObject, eventdata, handles)
% hObject    handle to rivnjanja (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of rivnjanja as text
%        str2double(get(hObject,'String')) returns contents of rivnjanja as a
double

% --- Executes during object creation, after setting all properties.
function rivnjanja_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rivnjanja (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Q2_Callback(hObject, eventdata, handles)
% hObject    handle to Q2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Q2 as text
%        str2double(get(hObject,'String')) returns contents of Q2 as a double

% --- Executes during object creation, after setting all properties.
function Q2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Q2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Q3_Callback(hObject, eventdata, handles)
% hObject    handle to Q3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Q3 as text
%        str2double(get(hObject,'String')) returns contents of Q3 as a double

```

```

% --- Executes during object creation, after setting all properties.
function Q3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Q3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Q4_Callback(hObject, eventdata, handles)
% hObject    handle to Q4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Q4 as text
%         str2double(get(hObject,'String')) returns contents of Q4 as a double

% --- Executes during object creation, after setting all properties.
function Q4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Q4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function E1_Callback(hObject, eventdata, handles)
% hObject    handle to E1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of E1 as text
%         str2double(get(hObject,'String')) returns contents of E1 as a double

% --- Executes during object creation, after setting all properties.
function E1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to E1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function E3_Callback(hObject, eventdata, handles)
% hObject    handle to E3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of E3 as text
%         str2double(get(hObject,'String')) returns contents of E3 as a double

```

```

% --- Executes during object creation, after setting all properties.
function E3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to E3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function E2_Callback(hObject, eventdata, handles)
% hObject    handle to E2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of E2 as text
%        str2double(get(hObject,'String')) returns contents of E2 as a double

% --- Executes during object creation, after setting all properties.
function E2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to E2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function E4_Callback(hObject, eventdata, handles)
% hObject    handle to E4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of E4 as text
%        str2double(get(hObject,'String')) returns contents of E4 as a double

% --- Executes during object creation, after setting all properties.
function E4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to E4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Q1_Callback(hObject, eventdata, handles)
% hObject    handle to Q1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Q1 as text
%        str2double(get(hObject,'String')) returns contents of Q1 as a double

```

```

% --- Executes during object creation, after setting all properties.
function Q1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Q1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function K1_Callback(hObject, eventdata, handles)
% hObject    handle to K1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of K1 as text
%        str2double(get(hObject,'String')) returns contents of K1 as a double

% --- Executes during object creation, after setting all properties.
function K1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to K1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function K2_Callback(hObject, eventdata, handles)
% hObject    handle to K2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of K2 as text
%        str2double(get(hObject,'String')) returns contents of K2 as a double

% --- Executes during object creation, after setting all properties.
function K2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to K2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Per_Callback(hObject, eventdata, handles)
% hObject    handle to Per (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Per as text
%        str2double(get(hObject,'String')) returns contents of Per as a double

% --- Executes during object creation, after setting all properties.

```

```

function Per_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Per (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit96_Callback(hObject, eventdata, handles)
% hObject    handle to edit96 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit96 as text
%         str2double(get(hObject,'String')) returns contents of edit96 as a double

% --- Executes during object creation, after setting all properties.
function edit96_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit96 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit97_Callback(hObject, eventdata, handles)
% hObject    handle to edit97 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit97 as text
%         str2double(get(hObject,'String')) returns contents of edit97 as a double

% --- Executes during object creation, after setting all properties.
function edit97_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit97 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit98_Callback(hObject, eventdata, handles)
% hObject    handle to edit98 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit98 as text
%         str2double(get(hObject,'String')) returns contents of edit98 as a double

% --- Executes during object creation, after setting all properties.
function edit98_CreateFcn(hObject, eventdata, handles)

```

```

% hObject      handle to edit98 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit99_Callback(hObject, eventdata, handles)
% hObject      handle to edit99 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit99 as text
%         str2double(get(hObject,'String')) returns contents of edit99 as a double

% --- Executes during object creation, after setting all properties.
function edit99_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit99 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit100_Callback(hObject, eventdata, handles)
% hObject      handle to edit100 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit100 as text
%         str2double(get(hObject,'String')) returns contents of edit100 as a double

% --- Executes during object creation, after setting all properties.
function edit100_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit100 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit101_Callback(hObject, eventdata, handles)
% hObject      handle to edit101 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit101 as text
%         str2double(get(hObject,'String')) returns contents of edit101 as a double

% --- Executes during object creation, after setting all properties.
function edit101_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit101 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pobudov.
function pobudov_Callback(hObject, eventdata, handles)
global A B
% hObject handle to pobudov (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

q(1)=str2num(get(handles.Q1,'String'));
q(2)=str2num(get(handles.Q2,'String'));
q(3)=str2num(get(handles.Q3,'String'));
e(1)=str2num(get(handles.E1,'String'));
e(2)=str2num(get(handles.E2,'String'));
e(3)=str2num(get(handles.E3,'String'));
B=polyfit(q,e,2);
syms t
E1=inline(diff(B(1)*t^2+B(2)*t+B(3)));
K1=E1(300);
E2=inline(diff(A(4)*t^3+A(3)*t^2+A(2)*t+A(1)));
K2=E2(300);
set(handles.K1,'String',K1);
set(handles.K2,'String',K2);
set(handles.X_start,'String',250);
set(handles.X_end,'String',500);
set(handles.Y_start,'String',1);
set(handles.Y_end,'String',1.35);
set(handles.axes1,'XLim',[fix(str2num(get(handles.X_start,'String')))...
    fix(str2num(get(handles.X_end,'String'))]);
set(handles.axes1,'YLim',[str2double(get(handles.Y_start,'String'))...
    str2double(get(handles.Y_end,'String'))]);
e1=inline(B(1)*t^2+B(2)*t+B(3));
e2=inline(A(4)*t^3+A(3)*t^2+A(2)*t+A(1));
t=[str2num(get(handles.X_start,'String')):0.5:str2num(get(handles.X_end,'String'))
];
plot(t,e1(t),'Parent',handles.axes1,'EraseMode','none','Color',[0.78 0 0]);
plot(t,e2(t),'Parent',handles.axes1,'EraseMode','none','Color',[0.5 0 0]);
set(handles.pobudov,'Visible','off');
set(handles.pobudov1,'Visible','on');

% --- Executes on button press in pobudov1.
function pobudov1_Callback(hObject, eventdata, handles)
global A B
% hObject handle to pobudov1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
syms t s
format short
E1=inline(diff(B(1)*t^2+B(2)*t+B(3)));
K1=E1(300);
E2=inline(diff(A(4)*t^3+A(3)*t^2+A(2)*t+A(1)));
K2=E2(300);
set(handles.K1,'String',K1);
set(handles.K2,'String',K2);
Per1=inline(get(handles.Per,'String'));
p1=Per1(0)^-1;

```

```

Per2=inline(diff(Per1(s)^-1));
p2=Per2(0);
Per3=inline(diff(Per2(s)));
p3=Per3(0)/2;
Per=tf([1],[p3 p2 p1]);
per1=Per*K1;
per2=Per*K2;
dper=per1-per2;
set(handles.X_start,'String',0);
set(handles.X_end,'String',5);
set(handles.Y_start,'String',0);
set(handles.Y_end,'String',10^-5);
set(handles.axes1,'XLim',[fix(str2num(get(handles.X_start,'String')))...
    fix(str2num(get(handles.X_end,'String'))]);
set(handles.axes1,'YLim',[str2double(get(handles.Y_start,'String'))...
    str2double(get(handles.Y_end,'String'))]);
d=[0:0.05:str2num(get(handles.X_end,'String'))];
[y,d]=step(dper,d);
plot(d,y,'Parent',handles.axes1,'EraseMode','none','Color',[0.78 0 0]);
set(handles.pobudov1,'Visible','off');
set(handles.back,'Visible','on');
% --- Executes on button press in back.
function back_Callback(hObject, eventdata, handles)
% hObject    handle to back (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.uipanel1,'Visible','on');
set(handles.uipanel3,'Visible','on');
set(handles.uipanel9,'Visible','on');
set(handles.Graf_pbt,'Visible','on');
set(handles.Clear_pbt,'Visible','on');
set(handles.uipanel5,'Visible','on');
set(handles.text21,'Visible','on');
set(handles.text22,'Visible','on');
set(handles.text23,'Visible','on');
set(handles.text24,'Visible','on');
set(handles.A1,'Visible','on');
set(handles.A2,'Visible','on');
set(handles.A3,'Visible','on');
set(handles.A4,'Visible','on');
set(handles.Q1,'Visible','off');
set(handles.Q2,'Visible','off');
set(handles.Q3,'Visible','off');
set(handles.E1,'Visible','off');
set(handles.E2,'Visible','off');
set(handles.E3,'Visible','off');
set(handles.text80,'Visible','off');
set(handles.text81,'Visible','off');
set(handles.text82,'Visible','off');
set(handles.text84,'Visible','off');
set(handles.text86,'Visible','off');
set(handles.text87,'Visible','off');
set(handles.text88,'Visible','off');
set(handles.text89,'Visible','off');
set(handles.text90,'Visible','off');
set(handles.Per,'Visible','off');
set(handles.K1,'Visible','off');
set(handles.K2,'Visible','off');
set(handles.pobudov,'Visible','off');
set(handles.pobudov,'Visible','off');
set(handles.uipanel21,'Visible','off');
set(handles.back,'Visible','off');
set(handles.figure1,'Position',[40 5 207.6 48.231]);

```

Додаток В

Опис програми Pruvereni_harakterustuku_2

Програму Pruvereni_harakterustuku_2 створено для визначення приведених характеристик відцентрового нагнітача газоперекачуючого агрегата в реальному часі та отримання відхилення ε , за допомогою якого відбуватиметься аналізування технічного стану газоперекачувального агрегату та корекція антипомпажної системи. Таким чином запобігатиметься виникнення аварійних ситуацій.

Програма складено в програмному пакеті MATLAB. В середовищі GUIDE створено графічну оболонку програми. Сама програма зберігається в М-файлі. В ньому і проводиться програмування кожного елемента графічного редактора GUIDE. Вигляд програми подано в додатку Б.

Потрібно звернути увагу на те, що файл Pruvereni_harakterustuku_2.m може знаходитися в будь-якому каталозі, але перед запуском програми потрібно вказати пакету MATLAB шлях до цього файлу. Це можна зробити з командного меню (File | Set Path) або безпосередньо в командному вікні (addpath шлях_до_файлу).

В панелі «Виміряні дані» розміщені три панелі режимів роботи нагнітача. В кожному з них будуть виводитись дані, які зчитуються з давачів, для кожного із режимів роботи, а саме: тиск перед всмоктуючою камерою - $P_{вх}$ (кгС/см³), фактична частота обертів - N (об/хв), перепад тиску на всмоктуючій камері - dP_k (мм. рт. ст). Зчитування здійснюється при виборі режиму в розкриваючому списку. Перед побудовою графіка потрібно ввести дані лабораторії – густину газу (кг/м³), атмосферний тиск (мм. рт. ст.), коефіцієнт стисливості, та характеристику нагнітача: номінальні оберти (об/хв), коефіцієнт витрати, поправку на стиснення газу в камері; умови проведення – Z_{pr} , N_{pr} , R_{pr} .

При отриманні всіх потрібних даних можна приступити до побудови графіка. Для цього потрібно натиснути кнопку «Побудова графіка». Після цього будуть виведені коефіцієнт A , A_1 , A_2 , A_3 та криві реальної приведеної характеристики відцентрового нагнітача. Для знаходження нових кривих при нових даних потрібно натиснути кнопку «Очистити» та повторити введення та зчитування

даних (при необхідності).

Для отримання відхилення ε потрібно перейти у вікно №2, де проводиться апроксимація еталонної характеристики та ідентифікація її з отриманою. Щоб перейти у вікно №2 потрібно натиснути кнопку «Наступний крок». В цьому вікні в панелі «Еталонна характеристика» необхідно ввести 3 точки графіка характеристики для її апроксимації та ввести передавальну функцію об'єкта. Тоді натискаємо кнопку «Побудова графіка» і на екрані виводиться дві приведені характеристики – еталонна(верхня) та реальна(нижня). Також виводяться обраховані по статичній характеристиці коефіцієнти підсилення K_1 та K_2 . Натискаємо кнопку «Побудова графіка» ще раз і одержуємо відхилення ε .

Додаток Г

ІВАНО-ФРАНКІВСЬКИЙ НАЦІОНАЛЬНИЙ
ТЕХНІЧНИЙ УНІВЕРСИТЕТ НАФТИ І ГАЗУ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ОСВІТІ, ТЕХНІЦІ ТА ПРОМИСЛОВОСТІ

МАТЕРІАЛИ
НАУКОВО-ПРАКТИЧНОЇ
КОНФЕРЕНЦІЇ

Івано-Франківськ

2025

<i>М. І. Горбійчук, І. С. Єднак, Ю. В. Пилипчук</i>	Синтез оптимальної системи автоматичного керування процесом низькотемпературної сепарації природного газу	73
<i>В. О. Лось, О. В. Кучмистенко</i>	Моделювання та дослідження системи регулювання температурних режимів установки гідроочищення гасу в середовищі Aspen Hysys	75
<i>Є. Т. Боднарук, О. В. Кучмистенко</i>	Дослідження автоматизованої системи керування установки атмосферної перегонки нафти	77
<i>Н. М. Паньків, О. В. Кучмистенко</i>	Опис структурних зв'язків системи автоматичного регулювання барабанного котла	79
<i>Д. О. Бабенко, А. І. Лагойда</i>	Інтелектуалізація системи керування процесом хлорування бензолу з використанням цифрового двійника	81
<i>В. К. Калініченко, А. І. Лагойда</i>	Створення системи керування технологічним процесом виробництва технічного вуглецю шляхом термічного розкладання	83
<i>А. І. Яблонський, А. І. Лагойда</i>	Розроблення системи керування реактором у технології каталітичного крекінгу з псевдозрідженим шаром	85
<i>І. В. Савчин, А. І. Лагойда</i>	Модернізація системи автоматизованого керування процесом гідравлічного розриву пласта на нафтових і газових родовищах із використанням нечіткої логіки	87
<i>В. О. Ткаченко, А. І. Лагойда</i>	Синтез автоматизованої нейромережевої системи керування скруббером для депарафінації дизельного палива	90
<i>М. К. Калініченко, А. І. Лагойда</i>	Оптимізація багаторівневої системи автоматизованого керування процесом переробки вторинного нафтового дистилляту	92
<i>М. О. Дідоха, Л. І. Лагойда</i>	Розробка автоматизованої системи керування газоперекачувальним агрегатом із застосуванням нечіткого регулятора	94
<i>Н. С. Шеремета, Л. І. Лагойда</i>	Автоматизоване визначення та ідентифікація реальних приведених характеристик відцентрових нагнітачів	96

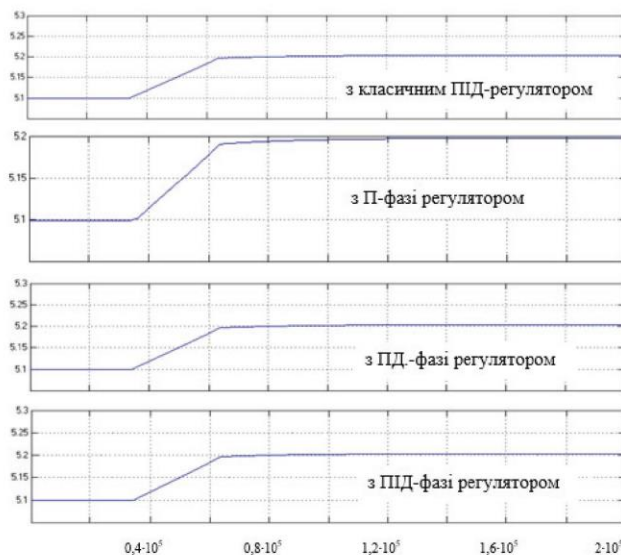


Рисунок 2 - Динаміка зміни тиску перед газоперекачувальним агрегатом з різними регуляторами

Літературні джерела

1 Семенцов, Г. Н. Теорія автоматичного управління (нелінійні та дискретні системи) [Текст] : навч. посіб. / Г. Н. Семенцов, І. І. Чигур. – Івано-Франківськ : ІФНТУНГ, 2014. – 96 с.

2 Горбійчук, М. І. Математичне моделювання на ЕОМ технологічних об'єктів [Текст] : навч. посіб. / М. І. Горбійчук. – Івано-Франківськ : Факел, 2001. – 240 с.

УДК 681.5

АВТОМАТИЗОВАНЕ ВИЗНАЧЕННЯ ТА ІДЕНТИФІКАЦІЯ РЕАЛЬНИХ ПРИБЕДЕНИХ ХАРАКТЕРИСТИК ВІДЦЕНТРОВИХ НАГНІТАЧІВ

Н. С. Шеремета, Л. І. Лагойда

*ІФНТУНГ, 76019, м. Івано-Франківськ, вул. Карпатська, 15, тел.(0342)727167,
e-mail: liudmyla.lahoida@nung.edu.ua*

У вітчизняній технічній літературі термін «нагнітач» традиційно використовується для позначення лопатевих компресорних машин, що забезпечують відношення тиску понад 1,1 та не оснащені пристроями для охолодження робочого середовища під час процесу стискання. У зарубіжних джерелах, зокрема англомовних, подібні машини здебільшого класифікують як компресори, а іноді застосовують термін бустери.

У ході проведених досліджень було визначено приведені характеристики

АВТОМАТИЗАЦІЯ ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ

E_{np} та Q_{np} , а також побудовано графіки реальних приведених характеристик відцентрових нагнітачів (ВН), що входять до складу газоперекачувальних агрегатів (ГПА). Це дало можливість здійснювати оперативний аналіз технічного стану ГПА та прогнозувати виникнення передаварійних ситуацій.

Крім того, розроблено алгоритм, на основі якого створено програмне забезпечення для визначення реальних приведених характеристик відцентрових нагнітачів.

З метою реалізації цього підходу запропоновано систему збору та передачі даних з об'єкта на персональний комп'ютер оператора. Вона включає: первинні вимірювальні перетворювачі, програмований логічний контролер, блок перетворення інтерфейсів та робочу станцію з інстальованим прикладним програмним забезпеченням (рис. 1).



Рисунок 1 - Блочна схема запропонованої системи

У середовищі MATLAB розроблено програму для визначення приведених характеристик відцентрових нагнітачів, яка забезпечує можливість відстеження їх змін у реальному часі (рис. 2).

У панелі «Виміряні дані» розташовано три секції, що відповідають режимам роботи нагнітача. Для кожного режиму відображаються показники, зчитані з датчиків: тиск перед всмоктувальною камерою – $P_{вх}$ (кгс/см²),

фактична частота обертання – N (об/хв), перепад тиску у всмоктувальній камері – dP_k (мм рт. ст.). Зчитування здійснюється після вибору відповідного режиму у розкритому списку.

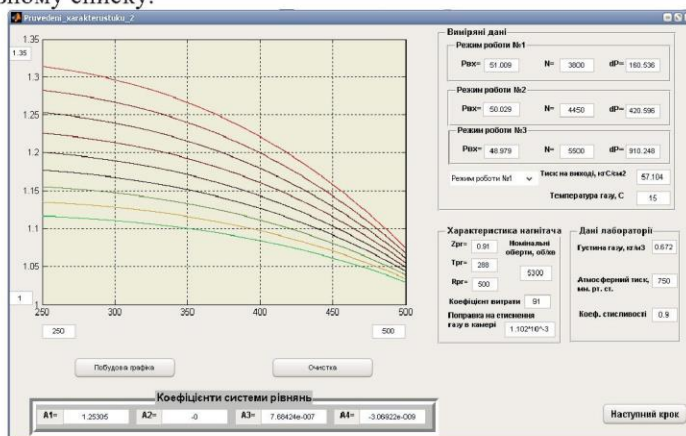


Рисунок 2 - Графічна оболонка програми в середовищі GUIDE

Перед побудовою графіка необхідно ввести лабораторні дані: густину газу ($\text{кг}/\text{м}^3$), атмосферний тиск (мм рт. ст.), коефіцієнт стисливості, а також параметри нагнітача: номінальну частоту обертів (об/хв), коефіцієнт витрати, поправку на стиснення газу в камері та умови проведення – Z_{pr} , N_{pr} , R_{pr} .

Після введення всіх необхідних даних користувач може перейти до побудови графіка, натиснувши кнопку «Побудова графіка». У результаті відображаються коефіцієнти A , $A1$, $A2$, $A3$ та криві реальної приведені характеристики відцентрового нагнітача. Для побудови нових кривих при зміні вхідних параметрів необхідно скористатися кнопкою «Очистити» та повторити процедуру введення і зчитування даних (за потреби).

Програма зберігається у вигляді М-файлу, де реалізовано програмування кожного елемента графічного редактора GUIDE.

Літературні джерела

- 1 Ковалко М. П., Грудз В. Я., Михалків В. Б., Тимків Д. Ф., Шлапак Л. С., Ковалко О. М. Трубопровідний транспорт газу: за редакцією М.П.Ковалка. К.: Агенство з раціонального використання енергії та екології, 2002. 600 с.
- 2 Остапенко Ю.О. Ідентифікація та моделювання технологічних об'єктів керування. К.: Задруга, 1999. 424 с.
- 3 Семенцов, Г. Н. Планування та обробка результатів експерименту [Текст]: навч. посіб. / Г. Н. Семенцов, Я. Р. Когуч, М. М. Дранчук. – Івано-Франківськ : ІФНТУНГ, 2003. – 222 с.



ІТЕТІ20250127

СЕРТИФІКАТ

УЧАСНИКА КОНФЕРЕНЦІЇ

0,15 кредити ECTS

ШЕРЕМЕТА Н. С.

Всеукраїнська Інтернет-конференція молодих учених і студентів
«Інформаційні технології в освіті, техніці та промисловості»



09 жовтня 2025

Голова організаційного комітету Сергій ЗІКРАТИЙ

Івано-Франківськ

