

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНТЕЛЕКТУАЛЬНИХ
ТЕХНОЛОГІЙ І ЗВ'ЯЗКУ

КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ АВТОМАТИЗОВАНИХ
СИСТЕМ

Методичні вказівки
до індивідуальної роботи
для підготовки бакалаврів

Спеціальність:

151 – Автоматизація та комп'ютерно-інтегровані технології

Укладач: доц. Андрій СТОПАКЕВИЧ

Розглянуто та схвалено на засіданні кафедри
Автоматики та комп'ютерно-інтегрованих технологій

Протокол від “29” серпня 2022 року № 1

В.о. завідувача кафедри _____ Олена ВОРОБІОВА
(підпис)

ЗМІСТ

Вступ	5
Основні визначення	6
Застосовані аббревіатури	7
ЗАВДАННЯ ТА ВИМОГИ ДО ВИКОНАННЯ ІНДИВІДУАЛЬНОЇ РОБОТИ	8
1 ПРАВИЛА АНАЛІЗУ БАЗОВИХ СХЕМ АВТОМАТИЗАЦІЇ (БСА) ПРОМИСЛОВИХ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ (ПТП)	10
1.1 Основні типи параметрів ПТП	10
1.2 Типові позначення автоматичних регуляторів ПТП	10
1.3 Типові позначення параметрів технологічних потоків ПТП	11
1.4 Приклад аналізу БСА ПТП	11
2 ПРОЦЕДУРА РОЗРОБКИ СИСТЕМИ АВТОМАТИЧНОГО КЕРУВАННЯ (САК) ПТП	13
2.1 Математичне моделювання ПТП	13
2.2 Розрахунок параметрів ПІ-регуляторів	15
2.3 Моделювання поведінки(динаміки) САК	18
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МОДЕЛЮВАННЯ ПТП В РЕАЛЬНОМУ ЧАСІ	18
3.1 Аналіз БСА ПТП та визначення тегів для OPC-сервера	20
3.2 Розробка скрипта для моделювання ОК	20
4 РОЗРОБКА МАКЕТА ЛЮДИНО-МАШИННОГО ІНТЕРФЕЙСУ (ЛМІ) СИСТЕМИ АВТОМАТИЗАЦІЇ	25
4.1 Визначення основних вимог до функціональності ЛМІ	25
4.2 Розробка макета екрана I рівня	27
4.3 Розробка макета екрана II рівня	29
4.4 Розробка макета екрана III рівня	29
5 РЕАЛІЗАЦІЯ ЛЮДИНО-МАШИННОГО ІНТЕРФЕЙСУ (ЛМІ) В SCADA-СИСТЕМІ SIMPLE SCADA	32
5.1 Налаштування переліку змінних проекту	32
5.2 Розробка програми для ПІД-регулювання	35
5.3 Підготовка зображень	37
5.4 Розробка екрана I рівня	39
5.5 Розробка екрана II рівня	44

5.6 Розробка екрана III рівня	45
ДОДАТКИ	46
Додаток А. CSV файл з переліком тегів	46
Додаток Б. Код скрипта ПІД-регулятора	48
Додаток В. Код скрипта для відображення ПС	51
Додаток Г. Код скрипта для сигналізації параметрів	53
ВАРІАНТИ ЗАВДАНЬ	56

Вступ

Людино-машинні інтерфейси (ЛМІ) операторів систем автоматизації технологічних процесів мають відповідати стандартам розробки комп'ютерних інтерфейсів людина-машина (ISO 9241 та ін.). Актуальним завданням для комп'ютерно-інтегрованих систем автоматизації є впровадження рекомендацій стандарту ANSI/ISA-101.01-2015 в процедуру розробки та контролю якості ЛМІ.

Розробка сучасних комп'ютерно-інтегрованих системи автоматизації вимагає від розробників значної кваліфікації в галузі інформаційних технологій. Для розробки ЛМІ систем автоматизації найчастіше використовують SCADA-системи, що підтримують стандартизовані інтерфейси передачі даних реального часу. Оскільки розробка та відлагодження графічних інтерфейсів проводиться автономно, то працездатність інтерфейсу необхідно перевіряти за допомогою комп'ютерних моделей технологічних процесів. Для розробки таких моделей доцільно застосовувати пакет Matlab. За Matlab можливо швидко розробляти імітаційні моделі динаміки технологічних об'єктів, що мають працювати в реальному часі і обмінюватись даними за промисловим протоколом OPC.

Основні визначення

Комп'ютерно-інтегрована промислова система автоматизації (СА) – система, що передбачає обов'язкове використання комп'ютера для задач управління елементами СА та обміну інформацією з програмним забезпеченням інформаційних систем промислового підприємства. З розвитком інформаційних та комп'ютерних технологій в СА стає можливим зменшити кількість обов'язкових елементів: засоби автоматизації (датчики і виконавчі механізми), що під'єднані до промислової мережі (в тому числі бездротової) керуються без додаткових перетворювачів сигналу за допомогою промислового комп'ютера чи мікроконтролера. Для управління щити і пульти все частіше замінюються на комп'ютери з сенсорними екранами та на промислові панелі.

Людино-машинний інтерфейс (ЛМІ) комп'ютерно-інтегрованої СА – це сукупність технічних, програмних засобів та правил взаємодії людини з технічною системою.

Технічні засоби ЛМІ сучасної промислової СА – промислові та персональні комп'ютери, мобільні пристрої, сенсорні панелі.

Програмні засоби ЛМІ – системи візуалізації і управління, в яких реалізований зв'язок зі стандартними інтерфейсами систем автоматизації. Для комп'ютерних ЛМІ це системи HMI/SCADA, що мають зрозумілі та порівняно прості в застосуванні засоби розробки (програмування) ЛМІ.

Правила взаємодії людини з технічною системою формуються виходячи з вимог виробництва та стандартів промислової ергономіки. Процес розробки таких правил найчастіше ітераційний і має закінчитись розробкою докладної інструкції для оператора-технолога. Сучасним стандартом промислової ергономіки є ANSI/ISA-101.01-2015, що є оптимальним для промислових процесів, які не є повністю автоматизованими. На теперішній час повністю автоматизовані процеси розробляються в економічно розвинених країнах тільки для галузей, в яких людина є ключовим елементом процесу і може бути замінена роботом (збірка обладнання та автомобілів, автоматизація складів, сфера обслуговування, тощо). В галузях, в яких використовуються вибухонебезпечні технологічні процеси досягнути високого ступеню автоматизації на разі майже не можливо, тому їх СА вимагають постійного нагляду та періодичного втручання людини. Таке втручання реалізується за допомогою ЛМІ і розробка таких ЛМІ є дуже відповідальною справою, оскільки ціна помилки людини-оператора чи розробника може бути далеко не тільки економічною.

Застосовані аббревіатури

ПТП – промисловий технологічний процес

СА – система автоматизації

ЛМІ – людино-машинний інтерфейс

КЗ – керована змінна ПТП

КВ – керуючий вплив ПТП

ПС – параметр стану ПТП

БСА – базова схема автоматизації

САК – система автоматичного керування

ПД – пропорційно-інтегруючий-диференціюючий

ОК – об'єкт керування

OPC – OLE for PC Control

SCADA – Supervisory Control and Data Acquisition

ЗАВДАННЯ ТА ВИМОГИ ДО ВИКОНАННЯ ІНДИВІДУАЛЬНОЇ РОБОТИ

Результати виконання індивідуальної роботи оформлюються письмово на аркушах А4 14 шрифтом через 1,5 інтервала.

В першому семестрі необхідно представити першу частину, яка містить 1-4 пункти, в другому 5-7.

Текст індивідуальної роботи має складатись з наступних *обов'язкових* розділів.

1. Вступ. Включає опис мети роботи та засобів, що використовувались для її виконання.
2. Опис технологічного процесу. В розділі необхідно навести:
 - невелику (до 1-2 стр.) характеристику промислового технологічного процесу за варіантом (скористатись пошуком в інтернеті та обробити його результати);
 - базову схему автоматизації процесу;
 - результати аналізу параметрів процесу: провести класифікацію та припустимі відхилення його керованих змінних, керуючих впливів, параметрів для аналізу стану.
3. Розробка децентралізованої системи автоматичного керування. В розділі необхідно навести:
 - розгінні характеристики моделі об'єкта керування;
 - отримані параметри ПІ-регуляторів та опис методики їх отримання;
 - перехідні процеси системи автоматичного керування з обґрунтуванням її працездатності;
 - код програми, за допомогою якої були отримані перехідні процеси.
4. Розробка програмної імітаційної моделі промислового технологічного процесу. В розділі необхідно навести:
 - коротку характеристику ОРС-сервера, що використовується;
 - перелік змінних ОРС-сервера, що обрані для застосування;
 - програмний код, що реалізує імітаційну модель
 - протокол роботи, що демонструє працездатність розробленої програмної імітаційної моделі.
5. Розробка макета людино-машинного інтерфейсу. В розділі необхідно навести:
 - сформовану кольорову карту ЛМІ в вигляді таблиці;
 - сформовану елементну базу ЛМІ в вигляді таблиці;

- формулу розрахунку критерію економічної ефективності та її обґрунтування;
 - обґрунтування сформованих вимог для екранів ЛМІ;
 - зовнішній вигляд макетів екранів 3 рівнів ЛМІ.
6. Розробка людино-машинного інтерфейса в Simple SCADA. В розділі необхідно навести:
- сформований перелік шкал Simple SCADA з межами та одиницями вимірювання в вигляді таблиці;
 - сформований перелік змінних Simple SCADA (зовнішніх та внутрішніх) з їх вказаними параметрами в вигляді таблиці;
 - програмний код для реалізації ПІД-регулювання;
 - зовнішній вигляд екрана відлагодження регулятора з перехідними процесами по завданню та збуренню, що демонструють працездатність регулятора;
 - зовнішній вигляд працюючого екрана І рівня в певній технологічній ситуації та необхідний програмний код для його реалізації;
 - зовнішній вигляд працюючого екрана ІІ рівня в певній технологічній ситуації та необхідний програмний код для його реалізації.
7. Висновки. Включає опис основних результатів, що досягнуто. Також бажано навести шляхи вдосконалення розробленого ЛМІ.

1. ПРАВИЛА АНАЛІЗУ БАЗОВИХ СХЕМ АВТОМАТИЗАЦІЇ (БСА) ПРОМИСЛОВИХ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ (ПТП)

1.1 Основні типи параметрів ПТП

Параметри виробничих процесів розділимо на три основні типи:

1) керовані змінні (КЗ) – параметри, що підлягають автоматичному регулюванню (підтриманню постійного значення);

2) керуючі впливи (КВ) – параметри, що підлягають зміні для досягнення постійних значень керованих змінних.

3) параметри для аналізу стану виробництва (ПС) – параметри, відхилення яких від норми свідчить про порушення ходу технологічного процесу, що може призвести до аварійної ситуації, зниження якості продукції чи збільшення собівартості виробництва.

Параметр виробничого процесу може належати як до одного, так і до декількох типів одночасно. Наприклад, витрата газу в котел – це одночасно КВ і ПС.

1.2 Типові позначення автоматичних регуляторів ПТП

Поточні значення керованих змінних отримуються з датчиків і передаються в керуючий пристрій (як правило контролер чи комп'ютер). В цьому пристрої реалізуються алгоритми регулювання, що розраховують нові значення КВ.

Для більшості ПТП можливо використовувати алгоритми регулювання, що використовують ПД-закон. В такому випадку досягнення бажаного значення однієї КЗ досягається зміною одного КВ. Зв'язок між однією КЗ і одним КВ називають контуром керування.

Найпростіший контур керування показано на рисунку 1.1. В цьому контурі КЗ – концентрація певного компонента в суміші, що виходить з ребойлера (кип'ятильника), КВ – витрата пари в ребойлер.

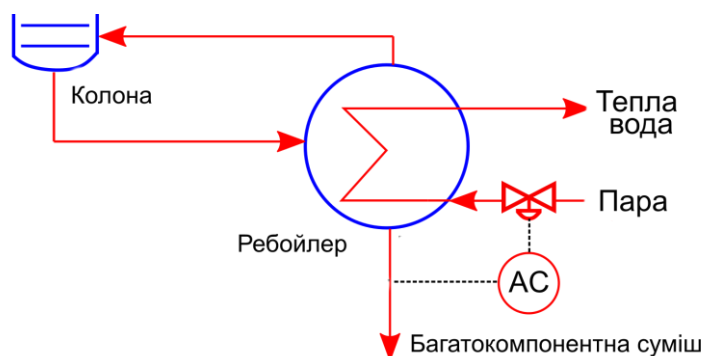


Рисунок 1.1 – Приклад зображення на БСА ПТП контуру керування

Алгоритм регулювання в контурі технічно змінює кількість витрати пари шляхом зміни ступеню відкриття положення регулюючого клапану. Така зміна проводиться алгоритмом в випадку якщо значення концентрації заданого компоненту відхилюється від бажаного (заданого).

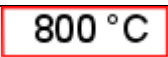
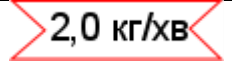
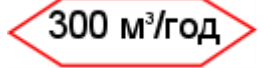
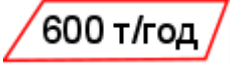
Позначення регуляторів на БСА проводиться за допомогою незафарбованого кола, в середині якого записуються дві літери, друга з яких обов'язково літера «С». КЗ та КВ контура показуються за допомогою пунктирної лінії, що веде до кола.

Перша літера в тексті кола описує тип параметра КЗ з фізичної точки зору. Перелік можливих перших літер наступний: «Т» – температура, «Р» – тиск, «F» – витрата, «L» – рівень, «S» – швидкість, «Е» – електричний параметр (напруга, потужність, сила тока, тощо), «М» – вологість, «А» – параметр, визначений за допомогою інтелектуального датчика за заданим алгоритмом (наприклад, концентрація компоненту в суміші чи газі, значення рН, тощо). Букви «Х», «У», «Z» зарезервовані для параметрів інших типів.

1.3 Типові позначення параметрів технологічних потоків ПТП

Стандартні позначення параметрів технологічних потоків ПТП приведені в таблиці 1.1.

Таблиця 1.1 – Стандартні позначення параметрів технологічних потоків ПТП

Тип позначення	Приклад застосування
Номер потоку	
Температура	
Тиск	
Значення параметра з інтелектуального датчика	
Витрата рідини	
Витрата газу	
Витрата маси	
Вхідний чи вихідний технологічний потік (незафарбована частина вказує на напрям потоку)	

1.4 Приклад аналізу БСА ПТП

Розглянемо БСА ПТП, що зображена на рисунку 1.2. Мета процесу – розділити суміш метанолу з водою, що поступає до колони, на метанол та воду.

Технологічний процес має три технологічних агрегата (колона для розділення, охолоджувач та ребойлер), три вхідних потоки (суміш, пара та холодна вода), чотири вихідних потоки (два потоки теплої води, потік чистого метанолу, вода з домішками).

Автоматичне регулювання проводиться по двом КЗ: концентрації метанолу в верхньому і нижньому продукті колони Т-101. Нормальний хід процесу залежить від роботи охолоджувача Е-101 та ребойлера Е-102. КВ процесу є витрата пари та витрата частини охолодженого метанолу, що повертається до колони Т-101 (т.з. флегми).

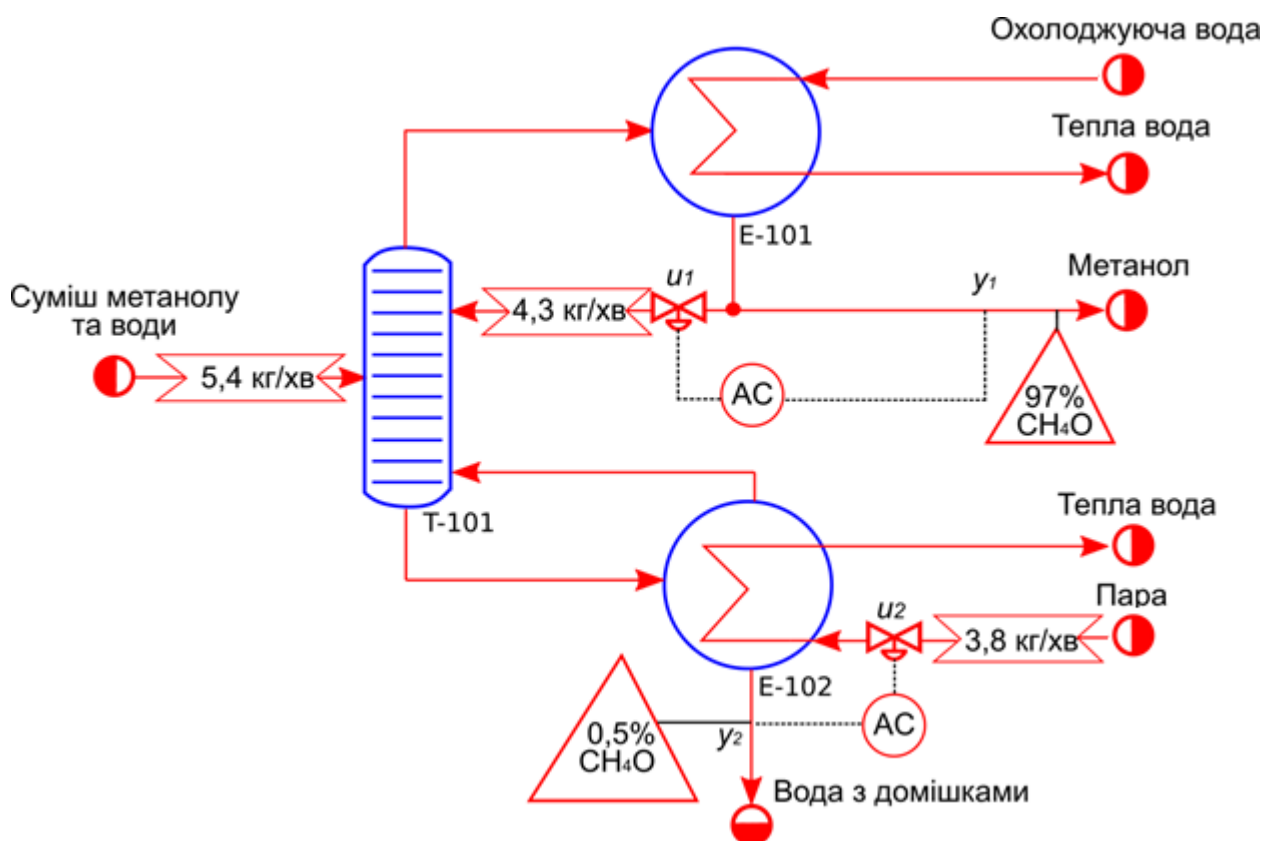


Рисунок 1.2 – БСА ПТП розділення суміші метанолу та води

2. ПРОЦЕДУРА РОЗРОБКИ СИСТЕМИ АВТОМАТИЧНОГО КЕРУВАННЯ (САК) ПТП

Процедура розробки САК ПТП має базуватись на досягненнях теорії автоматичного керування і використовувати засоби математичного моделювання. Для виконання завдання вона виконується в три кроки:

- 1) Математичне моделювання динаміки ПТП;
- 2) Розрахунок параметрів ПІ-регуляторів з використанням математичної моделі;
- 3) Моделювання поведінки(динаміки) САК при впливі збурень з метою визначення працездатності САК й можливості дотримання технологічного регламенту при заданих збуреннях.

Для розробки САК створить Matlab-скрипт з заданим ім'ям за допомогою команди edit. Наприклад введіть в командному вікні Matlab

➤ edit control_system_design

Буде визвано редактор скриптів, команди в якому можливо виконувати за допомогою натискання клавіші F5. Докладніше про роботу з Matlab можливо прочитати в відповідній літературі та документації пакета.

2.1 Математичне моделювання ПТП

Типовим представленням математичної моделі динаміки ПТП є матриця передатних функцій. Стовпці матриці означають входи(u) об'єкту, строки – виходи (y).

В якості об'єкта керування (ОК) для прикладу розглянемо ПТП розділення суміші метанолу та води, що розглянуто в розділі 1.4.

Експериментальна модель ПТП розділення суміші метанолу та води задана наступним чином (час моделі в *хвилинах*)

$$W(s) = \begin{matrix} & \begin{matrix} u_1 & u_2 \end{matrix} \\ \begin{matrix} y_1 \\ y_2 \end{matrix} & \begin{bmatrix} \frac{5,8182}{16,7s+1}e^{-1s} & \frac{-8,5909}{21s+1}e^{-3s} \\ \frac{3}{10,9s+1}e^{-7s} & \frac{-8,8182}{14,4s+1}e^{-3s} \end{bmatrix} \end{matrix}$$

де u_1 – витрата флегми – частини охолодженого метанолу, що повертається до колони, виходячи з БСА номінальне значення витрати 4,3 кг/хв).

u_2 – витрата пари в ребойлер (виходячи з БСА номінальне значення витрати 3,8 кг/хв).

y_1 – концентрація метанолу в верхньому продукті (виходячи з БСА номінальне значення концентрації 97 %).

y_2 – концентрація метанолу в нижньому продукті (виходячи з БСА номінальне значення концентрації 0,5 %).

В скрипті розглянута математична модель представляється наступним чином.

```
% процент. метанола/ кг/хв  
k=[5.8182    -8.5909 ;3.0000    -8.8182];  
T=[16.7 21; 10.9 14.4]; %час в хв.  
tau=[1 3; 7 3]; %час в хв.
```

- ❖ Зверніть увагу, що час в даному випадку *залишимо* в хвилинах, оскільки перехідні процеси будуть занадто тривалі. В більшості випадків доцільно проводити при виконанні завдання перерахунок постійних часу та запізнень в секунди з хвилин. Таким чином, масштаб часу $1с=1хв$ реального процесу.

Введемо позначення і номінальні значення КЗ і КВ розглянутого ОК, а також вкажемо одиниці вимірювання часу.

```
model_time_units = 'час в хв';  
u1_title='Витрата флегми, кг/хв';  
u2_title='Витрата пари, кг/хв';  
y1_title='Конц. метанолу зверху';  
y2_title='Конц. метанолу знизу';  
  
ynom=[96.25;0.5];  
unom=[4.29;3.85];
```

Матриця передатних функцій моделі ОК в програмі сформується наступним чином.

```
W11=tf(k(1,1), [T(1,1) 1], 'ioDelay', tau(1,1));  
W12=tf(k(1,2), [T(1,2) 1], 'ioDelay', tau(1,2));  
W21=tf(k(2,1), [T(2,1) 1], 'ioDelay', tau(2,1));  
W22=tf(k(2,2), [T(2,2) 1], 'ioDelay', tau(2,2));  
W=[W11 W12;W21 W22];  
W.InputName = {'u1', 'u2'}; W.OutputName = 'y';
```

Якщо виконати скрипт, то розгінні характеристики можливо перевірити за допомогою команди

➤ step W

Приклад виконання команди показаний на рисунку 2.1

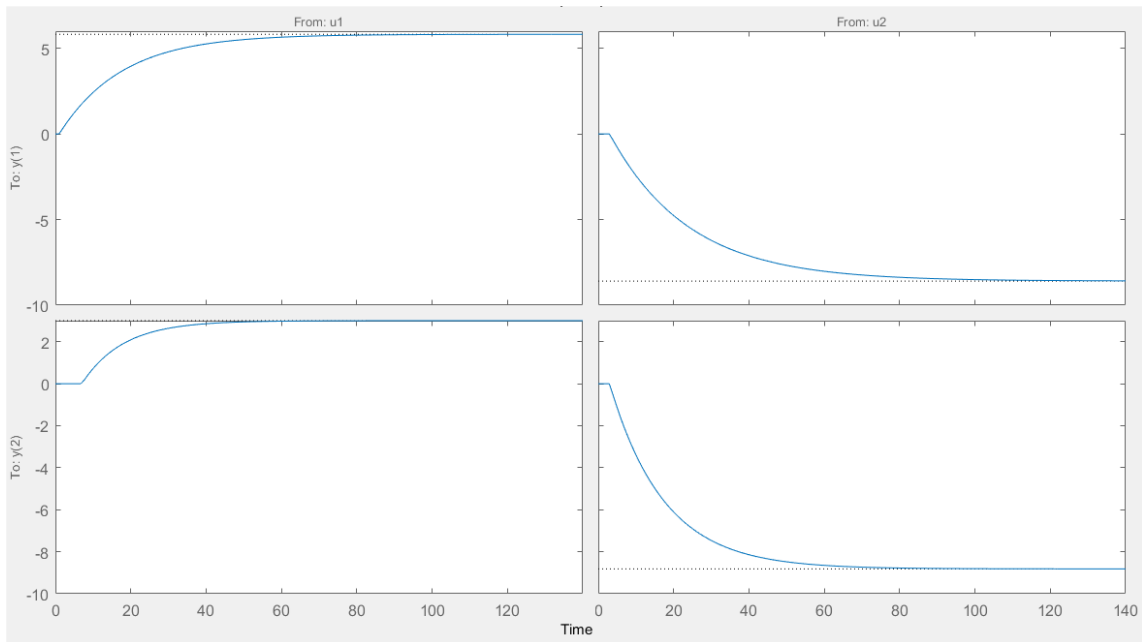


Рисунок 2.1 – Розгінні характеристики ОК

Аналіз розгінних характеристик дозволяє перевірити коректність завдання моделі: коефіцієнтів передачі, постійних часу, часів запізнення усіх передатних функцій.

2.2 Розрахунок параметрів ПІ-регуляторів

Найпростіша структура САК для розглянутих в завданні ОК – структура зі зворотнім зв'язком, що включає два ПІ-регулятора.

Налаштування ПІ-регуляторів в такій структурі можливо визначити великою кількістю шляхів, однак більшість з них не буде враховувати наявність перехресних зв'язків між каналами ОК. Тому результативні налаштування в ряді випадків будуть не оптимальними чи взагалі непрацездатними, оскільки ПІ-регулятори будуть «заважати» один одному.

Розглянемо метод, що використовує функцію автоматичного визначення налаштувань Matlab за непрямим частотним критерієм – частотою зрізу. Частота зрізу – частота, при якій амплітудно-частотна характеристика системи $A_z(j, \omega)$ приймає значення $A_z(0)$ чи найменша частота, в якій передатна функція розімкненої САК має одиничне підсилення. Ця частота непрямо характеризує швидкість перехідних процесів. Для об'єктів першого порядку можливо приблизно встановити, що

час регулювання замкненої САК за завданням буде визначатись за частотою зрізу розімкненої САК наступним чином

$$T_s \approx (1..2) \frac{2\pi}{w_c}$$

Розглянемо приклад аналізу часу регулювання для найпростішого об'єкту

$$W = \frac{1}{5s + 1}$$

з ПІ-регулятором виду

$$Kp + \frac{Ki}{s}$$

Визначення налаштувань проводиться наступним чином

➤ `Obj=tf(1,[5 1])`

ПІ-регулятор з запасом по фазі в 60° (значення в межах $45..60^\circ$ гарантують працездатність САК при порівняно значному відхиленні поведінки ОК від його поведінки по моделі, що використовувалась при розрахунку параметрів регулятора) отримаємо за допомогою функції `pidtune`

➤ `C=pidtune(Obj,'pi')`

Отримаємо регулятор з $Kp=1.04, Ki=0.626$

Частота зрізу отримається з допомогою команди

➤ `sys=C*Obj;wc = getGainCrossover(sys,1)`

і дорівнює 0.3562. Тобто час регулювання має складати приблизно 18 с

Перехідний процес

➤ `step(C*Obj/(1+C*Obj))`

буде мати наступний вигляд

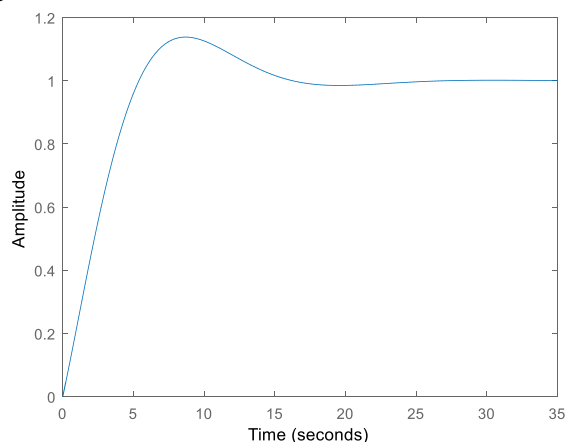


Рисунок 2.2 – Перехідний процес по завданню для найпростішого об'єкта
Таким чином, час регулювання дійсно близький до 18 с.

Для багатовимірного об'єкту, тобто для ОК що має число входів та виходів ≥ 2 , для визначення параметрів декількох пов'язаних ПІ регуляторів з гарантованою робастністю можливо використовувати функцію Matlab looptune. Ця функція в якості аргументу приймає орієнтовну частоту зрізу для всієї системи, яку потім уточнює шляхом вирішення оптимізаційної задачі. Простим варіантом вибору такої частоти w_c буде вибір частоти, що розраховується з визначення що T_s становить двократне середнє арифметичне суми постійних часу та запізнень передатних функцій ОК.

Наприклад для розглянутого ОК

$$W(s) = \begin{matrix} & u_1 & u_2 \\ y_1 & \frac{5,8182}{16,7s+1}e^{-1s} & \frac{-8,5909}{21s+1}e^{-3s} \\ y_2 & \frac{3}{10,9s+1}e^{-7s} & \frac{-8,8182}{14,4s+1}e^{-3s} \end{matrix}$$

таке середнє арифметичне становить $(15.75+3.5)*2 \approx 38$ хв, тоді w_c становитиме 0.1653 рад/хв.

Програмний код розрахунку параметрів ПІ-регуляторів для розглянутого ОК матиме наступний вид

```
D = tunableGain('Decoupler',eye(2));D.Gain.Free = [0
0;0 0];
D.InputName = 'e';D.OutputName = {'uu1','uu2'};
PI_AC1 = tunablePID('PI_AC1','pi');
PI_AC1.InputName = 'uu1';PI_AC1.OutputName = 'u1';
PI_AC2 = tunablePID('PI_AC2','pi');
PI_AC2.InputName = 'uu2';PI_AC2.OutputName = 'u2';
sum1 = sumblk('e = r - y',2);
C0 =
connect(PI_AC1,PI_AC2,D,sum1,{'r','y'},{'u1','u2'});
wc=1/(2*(fix(mean(mean(T))+mean(mean(tau)))/(2*pi)));
[W,C,gam,Info] = looptune(pade(W,1),C0,wc);
```

В результаті виконання коду будуть отримані 2 ПІ-регулятори з налаштуваннями

$$W_{PI-AC1} = 0.749 + \frac{0.0589}{s} = 0.749 \cdot \left(1 + \frac{1}{12.7s}\right)$$

$$W_{PI-AC2} = -0.0578 - \frac{0.0245}{s} = -0.0578 \cdot \left(1 + \frac{1}{2.36s}\right)$$

2.3 Моделювання поведінки(динаміки) САК

Для моделювання та представлення багатовимірних систем зручним є використання форми простору станів, що представляє систему диференціальних рівнянь першого порядку, представлених в матричному вигляді.

Перетворення в таку систему проводиться за допомогою функції `ss`. Апроксимація запізнення в системі проводиться за допомогою функції `pade`. Перетворення в дискретний час з кроком дискретності $1s$ проводиться за допомогою функції `c2d`.

```
Wss=c2d(ss(pade(W,1)),1);Css=c2d(ss(C),1);
```

Проведемо моделювання системи керування при збільшенні завдання на 1% по концентрації метанолу в верхньому і нижньому продуктах.

```
r=[1;1];  
  
y=[0;0];uu1=[];uu2=[];yy1=[];yy2=[];  
tt=[];f=0;t_max=max(max(T))+max(max(tau));u=[0;0];  
  
for i=0:3.5*t_max  
    e=[r(1);r(2);y(1);y(2)];  
    uu1=[uu1;u(1)];uu2=[uu2;u(2)];  
    yy1=[yy1;y(1)];yy2=[yy2;y(2)];  
    tt=[tt; i];  
    if abs(r(1)-y(1))<0.01 && abs(r(2)-y(2))<0.01  
        f=f+1;  
        if f>t_max*0.2, break; end  
    else  
        f=0;  
    end  
    xr=Css.a*xr+Css.b*e;    u=Css.c*xr+Css.d*e;  
    x=Wss.a*x+Wss.b*u;      y=Wss.c*x+Wss.d*u;  
end  
figure(1);  
clf  
subplot(2,2,1);plot(tt,yy1+ynom(1));ylabel(y1_title);  
subplot(2,2,2);plot(tt,uu1+unom(1));ylabel(u1_title);  
subplot(2,2,3);plot(tt,yy2+ynom(2));ylabel(y2_title);  
xlabel(model_time_units);subplot(2,2,4);  
plot(tt,uu2+unom(2));xlabel(model_time_units);ylabel(u2_title);
```

Перехідні процеси отриманої системи показані на рис. 2.3.

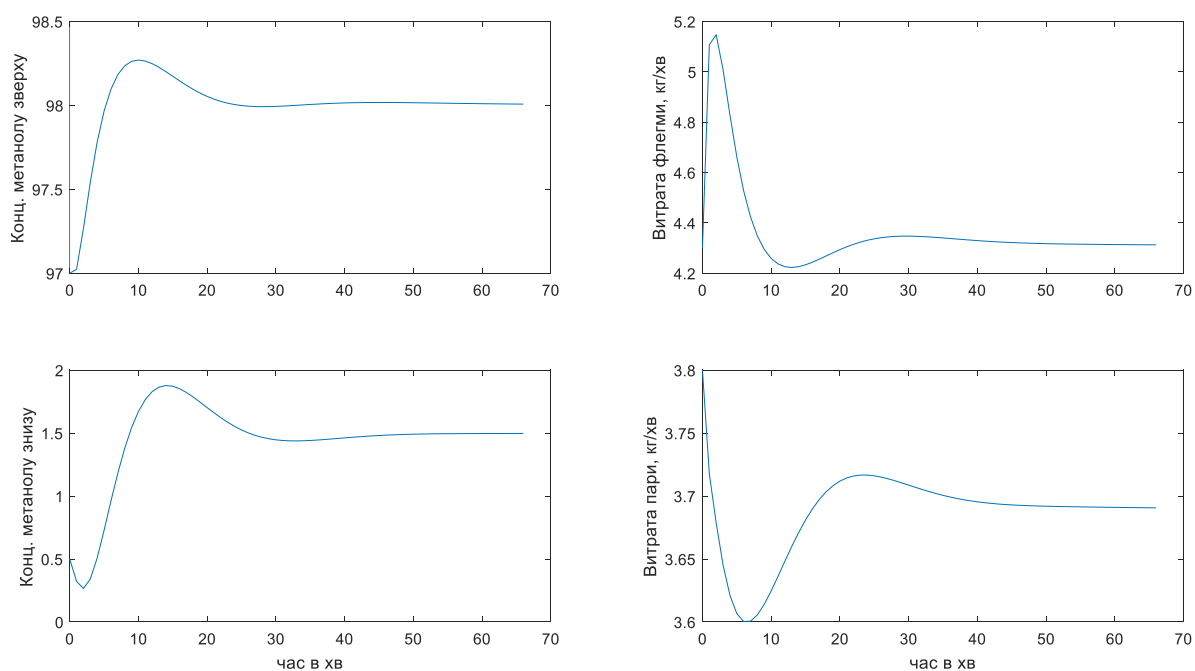


Рисунок 2.3 – Перехідні процеси при збільшенні завдань по концентрації двох продуктів

Аналіз перехідних процесів, що показані на рис. 2.3., демонструє що система керування збільшила величину концентрації верха та низа на 1% при реалістичному відхиленню КВ (витрати флегми й пари).

3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МОДЕЛЮВАННЯ ПТП В РЕАЛЬНОМУ ЧАСІ

Для моделювання ПТП необхідно розробити Matlab-скрипт, що виконується в реальному часі. Засобом обміну даними з моделлю має бути протокол OPC. В якості емулятора OPC-сервера доцільно використовувати сервер Greybox.

3.1. Аналіз БСА ПТП та визначення тегів для OPC-сервера

В сервері Greybox доступно 20 тегів для збереження дійсних чисел та 20 тегів для логічних значень. Це цілком достатньо для виконання завдання. Ознайомлення з сервером проводиться в лабораторній роботі №1 дисципліни.

За результатами аналізу прикладу БСА (рис. 1.4) сформуємо таблицю з закріпленням певних тегів за параметрами.

Таблиця 3.1 – Перелік змінних OPC-сервера

Ім'я тега	Параметр	Значення за замовчуванням
storage.numeric.reg01	Витрата флегми (u1)	4.3
storage.numeric.reg02	Витрата пари (u2)	3.8
storage.numeric.reg03	Концентрація метанола в верхньому продукті (y1)	97
storage.numeric.reg04	Концентрація метанола в нижньому продукті (y2)	0.5
storage.numeric.reg05	Витрата живлення в колону	5.4
storage.numeric.reg06	Збурення за витратою флегми	0
storage.numeric.reg07	Збурення за витратою пари	0
storage.bool.reg01	Система запущена	true
storage.bool.reg02	Відмова двигуна клапана витрати флегми	false
storage.bool.reg03	Відмова двигуна клапана витрати пари	false

3.2 Розробка скрипта для моделювання ОК

Основні вимоги до скрипта, що моделює ОК наступні:

- 1) Скрипт зчитує та записує необхідні параметри як змінні на OPC-сервері
- 2) Скрипт моделює реакцію об'єкта в реальному часі з інтервалом циклу в 0.5с

Для відлагодження роботи скрипта будемо використовувати Matrikon OPC Explorer.

Програмний код реалізуємо в вигляді функції oku_coursework. Вихідними даними функції буде модель, що відлагоджена в розділі 2.1 та дані таблиці 3.1.

```
function oku_cousework
    listOfTimers = timerfindall;
    if ~isempty(listOfTimers)
        delete(listOfTimers(:));
    end

    da = opcda('localhost','Graybox.Simulator.1');
    connect(da);
    cleanupObj = onCleanup(@finishScript);
    %крок дискретності моделювання об'єкта
    Ts=0.5;
    grp = addgroup(da, 'MColumn');
    f_reflux = additem(grp, 'storage.numeric.reg01');
    write(f_reflux, 4.3); %Витрата флегми (4.3 кг/хв)
    f_steam = additem(grp, 'storage.numeric.reg02');
    write(f_steam, 3.8); %Витрата пари (3.8 кг/хв)
    c_top = additem(grp, 'storage.numeric.reg03');
    write(c_top, 97); %Концентрація верха (97 %)
    c_btm = additem(grp, 'storage.numeric.reg04');
    write(c_btm, 0.5); %Концентрація низу (0.5 %)
    f_feed = additem(grp, 'storage.numeric.reg05');
    write(f_feed, 5.4); %Витрата живлення (5.4)
    d_reflux = additem(grp, 'storage.numeric.reg06');
    write(d_reflux, 0); %Збурення 1
    d_steam = additem(grp, 'storage.numeric.reg07');
    write(d_steam, 0); %Збурення 2

    running = additem(grp, 'storage.bool.reg01');
    write(running, 1); %Система запущена (1/0)
    f_reflux_alarm = additem(grp, 'storage.bool.reg02');
    write(f_reflux_alarm, 0); %Двигун для зміни витрати флегми
    відмовив (1/0)
    f_steam_alarm = additem(grp, 'storage.bool.reg03');
    write(f_steam_alarm, 0); %Двигун для зміни витрати пари
    відмовив (1/0)
```

```

pause(1)

vars={ 'f_reflux', 'f_steam', 'c_top', 'd_reflux', ...
       'd_steam', 'c_btm', 'f_feed', 'f_reflux_alarm',
...
       'f_steam_alarm', 'running'};

ynom=[97;0.5];
unom=[4.3;3.8];

k=[5.8182 -8.5909 ;3.0000 -8.8182];%процент. метанола/ кг/хв
T=[16.7 21; 10.9 14.4]; %час в хв
tau=[1 3; 7 3]; %час в хв

W11=tf(k(1,1),[T(1,1) 1],'ioDelay', tau(1,1));
W12=tf(k(1,2),[T(1,2) 1],'ioDelay', tau(1,2));
W21=tf(k(2,1),[T(2,1) 1],'ioDelay', tau(2,1));
W22=tf(k(2,2),[T(2,2) 1],'ioDelay', tau(2,2));
W=[W11 W12;W21 W22];

obj=c2d(ss(pade(W,1)),Ts);
x=zeros(length(obj.a),1);

t = timer;
t = timer('StartDelay',0, 'TimerFcn', ...
          @(h,~)onTimerCallback(h), 'Period',Ts);
set(t, 'ExecutionMode', 'FixedRate');
start(t)

while 1
    disp(sprintf('%s %s %s', datestr(now, 'HH:MM:SS'), ...
        ['u=',num2str(u')],...
        ['y=',num2str(y')]))
    pause(1)
end

function onTimerCallback(h)
    u=[f_reflux.Value-unom(1)+d_reflux.Value;...
       f_steam.Value-unom(2)+d_steam.Value];
    if u(1)<-unom(1), u(1)=-unom(1); end;
    if u(2)<-unom(2), u(2)=-unom(2); end;

    if u(1)>unom(1), u(1)=unom(1); end;
    if u(2)>unom(2), u(2)=unom(2); end;

    x=obj.a*x+obj.b*u;
    y=obj.c*x+obj.d*u;
    write(c_top,y(1) + ynom(1));
    write(c_btm,y(2) + ynom(2));
end

function finishScript()
    listOfTimers = timerfindall;

```

```

    if ~isempty(listOfTimers)
        delete(listOfTimers(:));
    end
    write(running,0);%Система запущена (1/0)
end

```

end

Запустимо скрипт в командному вікні Matlab

В командному вікні почнеться вивід поточних значень відхилень КВ та КЗ.

```
18:27:04 u=0  0 y=0  0
```

Запустимо OPC Explorer і додамо в ньому необхідні змінні. Зменшимо значення витрати пари на 0.2 кг/хв.

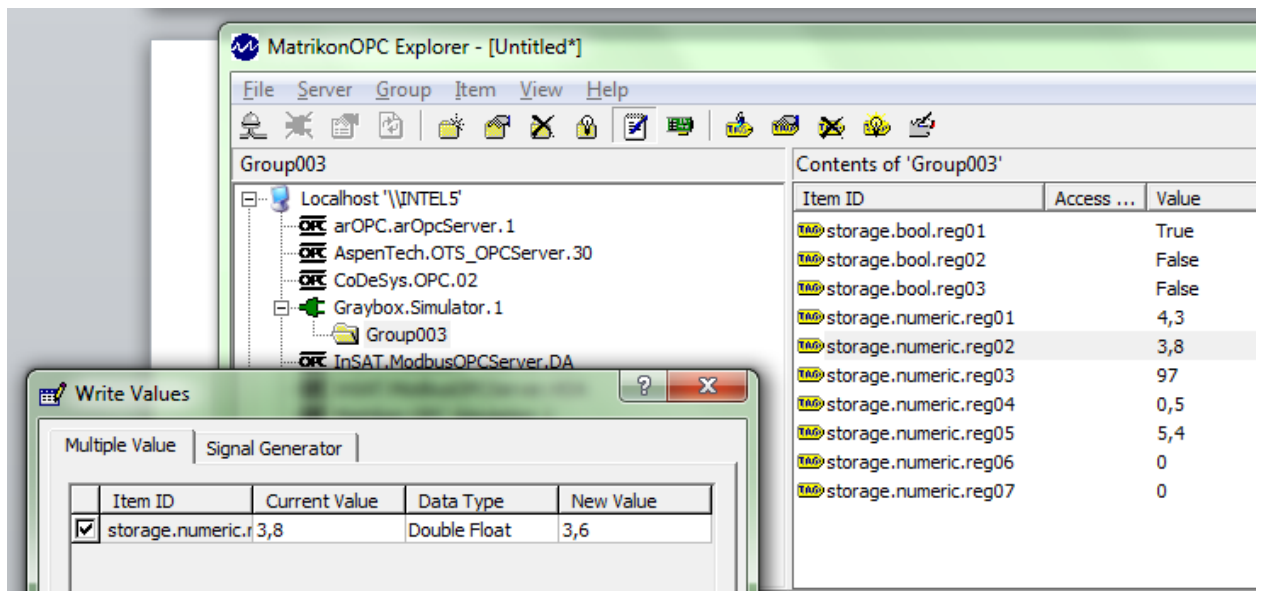


Рисунок 3.1 – Зміна відповідного значення в OPC Explorer

В командному вікні почнеться зміна значень КЗ

```

18:27:44 u=0      -0.2 y=-0.036428  -0.053736
18:27:45 u=0      -0.2 y=-0.014559  -0.020148
18:27:46 u=0      -0.2 y=0.035355   0.054169
18:27:47 u=0      -0.2 y=0.097868   0.14557
18:27:48 u=0      -0.2 y=0.16513    0.24217
18:27:49 u=0      -0.2 y=0.23321    0.3381

```

...

18:28:12	u=0	-0.2	y=1.2199	1.4735
18:28:13	u=0	-0.2	y=1.2431	1.493
18:28:14	u=0	-0.2	y=1.2652	1.5111

...

18:30:58	u=0	-0.2	y=1.718	1.7636
----------	-----	------	---------	--------

Таким чином, встановлені коефіцієнти передачі вірні, оскільки

$$-0.2 * -8.5909 = 1.71818$$

$$-0.2 * -8.8182 = 1.76364$$

Перехідні процеси тривали більше 2хв масштабованого часу, що відповідає рис. 2.1.

4. РОЗРОБКА МАКЕТА ЛЮДИНО-МАШИННОГО ІНТЕРФЕЙСУ (ЛМІ) СИСТЕМИ АВТОМАТИЗАЦІЇ

4.1 Визначення основних вимог до функціональності ЛМІ

Визначення принципів розробки ЛМІ оператора-технолога СА проведемо виходячи з аналізу основних проблем проектування інтерфейсів за Анохіним.

Проблема ступеню автоматизації. Оператор має виконувати наступні можливості втручання в хід процесу: зміна технологічного режиму роботи (завдань) і ручне управління технологічним процесом (відключення регуляторів). Задачі автоматичного пуску, зупинки, переналаштування процесу не розглядаємо. ЛМІ в SCADA-системі має виконувати задачу представлення інтерфейсу оператора та автоматичного регулювання параметрів за допомогою ПІ-регуляторів.

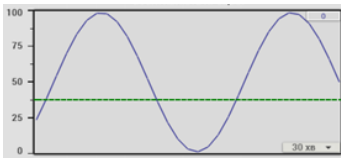
Проблема підтримки когнітивної діяльності. Функціональність елементів інтерфейсу для оператора повинна бути легко зрозумілою. Для цього використовуються такі засоби як колір, форма і положення на екрані. Найбільш швидким для розрізнення фактором інтерфейсу є колір. ISA 101 рекомендує використовувати в якості фону інтерфейсу світло-сірий колір. Цей колір мінімізує відблиски і віддзеркалення на екрані і оптимальний для порівняння з червоним, за яким закріплюються аварійні ситуації. За жовтим кольором закріплюються передаварійні ситуації. Рекомендовані значення кольорів відображені в таблиці 4.1.

Табл. 4.1 – Кольорова карта ЛМІ оператора-технолога

Колір	Значення	Колір	Значення
Темно-сірий	Підпис, рамка для зображення, елементи технологічної схеми	Аквамариновий	Припустима зона зміни параметра
Світло-сірий	Фон, неактивний елемент	Білий	Активний елемент (кнопка, текстове поле)
Синій	Поточне значення параметра	Зелений	Норматив параметра
Світло-синій	Бажане значення параметра		
Жовтий	Попереджувальна сигналізація	Червоний	Аварійна сигналізація

Рекомендована елемента база ЛМІ приведена в таблиці 4.2.

Табл. 4.2 – Елементи ЛМІ оператора-технолога

Елемент	Значення змін елемента
	Сигналізатор параметра (не зафарбований – нормальний стан, жовтий – передаварійний, червоний – аварійний)
	Сигналізатор помилки. Жовтий – передаварійна, червоний – аварійна.
	Індикатор показника виробництва. Синім текстом пишеться поточне значення параметра. Синя стрілка визначає поточне значення на шкалі. Зеленою лінією визначається норматив параметра.
	Кнопка переміщення по екранам, підтвердження вводу
	Поле вводу, з бажаним значенням параметра
	Тренд параметра в часі. Синій – зміна параметра в заданій шкалі. Шкала відображається якщо відображується важливий параметр, на який проводиться на екрані вплив. За допомогою поля вибору обирається час тренду. Зеленою пунктирною лінією позначаються припустимі значення.

Положення на екрані елементів інтерфейсів визначається по зонам для спрощення локалізації. Екрани розбиваються на зони за допомогою товстих сірих пунктирних ліній. Для спрощення визначення зв'язків між елементами контуру регулювання використовується тонка крапкова лінія.

Проблема великих обсягів даних. Відомо, що можливості пам'яті людини обмежені, тому кількість змінних, що необхідні для прийняття рішення оператором в ефективному ЛМІ не має перевищувати число Міллера (7 ± 2). Для досягнення цієї мети використовується три підходи: розбивання інформації на екрани, функціональна ієрархічна організація екранів, динамічна перестройка екранів.

Розбиття інформації на екрани має ґрунтуватися на проблемно-орієнтованому принципі. Тобто кожен екран повинен бути призначений для вирішення однієї або більше завдань. Інформації на екрані має бути достатньо для вирішення поставленого завдання, але сам екран не повинен по можливості бути перевантаженим інформацією і надлишково складним. Для зниження складності використовується поділ екрану на логічні зони та подання інформації за допомогою найбільш підходящих елементів.

Для виконання завдання необхідно провести розробку екранів трьох рівнів: I – оглядового, II – первинного і III – налагодження.

4.2 Розробка макета екрана I рівня

Для розв'язку задачі розробки макета екрана I рівня необхідно дати відповідь на наступні запитання:

- 1) Які основні показники виробництва ПТП, що доцільно фіксувати за час та зміну?
- 2) Які параметри ПТП необхідно сигналізувати?
- 3) Які регулятори в САК ПТП присутні?
- 4) Який узагальнений критерій ефективності ведення ПТП можливо обрати.

Дамо відповідь на ці питання на прикладі ПТП розділення суміші метанол-вода.

1) В ПТП основними показниками виробництва доцільно обрати витрату розділеної суміші, витрату пари, витрату флегми.

2) Необхідно сигналізувати значні відхилення витрат суміші, пари, флегми, концентрацій верху та низу.

3) Присутні два регулятори – концентрації верхнього та нижнього продуктів (АС101, АС102).

4) Критерій ефективності має давати узагальнене представлення об енергоефективності процесу, тобто скільки витрачається енергоресурсів на одиницю якісної продукції. Зрозуміло, що чим менше – тим краще. Бажано, щоб критерій був нормований, наприклад до 1. 1 – найвища енергоефективність, 0 – найнижча.

Визначимо, що критерій ефективності буде визначатись за відношенням кількості затраченої флегми і пари на одиницю потоку живлення й нормуватись до одиниці – найвищої енергоефективності.

Припустимо, що витрата флегми й пари однаково важлива, тоді вклад кожного компоненту складає від 0 до 0.5.

Задамо, що припустиме відхилення параметрів складає $\pm 30\%$ від номіналу.

Тоді відхилення витрати живлення знаходиться в межах 3.78 .. 7.02,

відхилення витрати пари знаходиться в межах 2.66 .. 4.94,

відхилення витрати флегми знаходиться в межах 3.01 .. 5.59.

Виведемо рівняння енергоефективності відношення витрати пари до витрати живлення.

Максимальним значенням відношення буде $\frac{4.94}{3.78} = 1.31$.

Мінімальним значенням відношення буде $\frac{2.66}{7.02} = 0.38$

Використовуючи лінійне рівняння

$$y = ax + b$$

сформуємо систему

$$\begin{cases} 0.5 = a \cdot 0.38 + b \\ 0 = a \cdot 1.31 + b \end{cases}$$

Розв'язком системи буде $b = 169/240$, $a = -819/1520$

Проведемо аналогічний розрахунок за витратою флегми до витрати живлення

Максимальним значенням відношення буде $\frac{5.59}{3.78} = 1.48$.

Мінімальним значенням відношення буде $\frac{3.01}{7.02} = 0.43$

Розв'язком системи буде $b = 169/240$, $a = -819/1720$

Таким чином, рівняння енергоефективності приймемо як

$$E = -0.4762 \cdot \frac{F_R}{F_F} + 0.7042 - 0.5388 \cdot \frac{F_S}{F_F} + 0.7042$$

Де F_R – витрата флегми, F_S – витрата пари, F_F – витрата живлення.

Використавши редактор векторної графіки, наприклад Inkscape, Visio, Corel Draw! можливо сформулювати макет. Приклад макета для екрана I рівня наведено на рисунку 4.1 Макет для вашого варіанту має виконуватись аналогічно.

Зверніть увагу, що надписи на екрані I рівня пишуться переважно великими літерами, щоб не відволікати на них увагу оператора. Він має запам'ятати місця відображення параметрів і зчитувати їх на рівні звички, тому підписи мають не відволікати (оскільки текст великими літерами читається складніше). Екрани II та III рівнів мають мати підписи, що легко читаються, тому вони виконуються переважно літерами малого регістру для того, щоб в момент стресу оператор не сплутав покази.

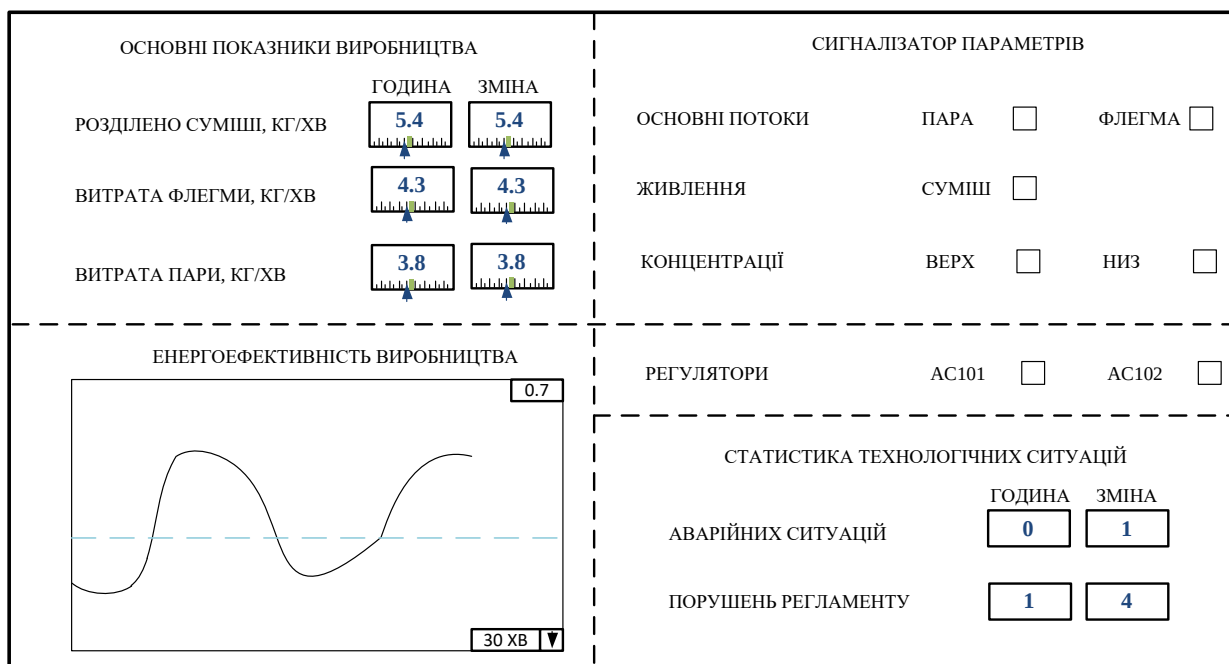


Рисунок 4.1 – Приклад екрана макета I рівня

4.3 Розробка макета екрана II рівня

Екран II рівня призначено для постійного перегляду оператора, але не для втручання в хід ПТП.

Для розв'язку задачі розробки макета екрана II рівня необхідно дати відповідь на наступні запитання:

- 1) Чи потрібно організовувати інтерфейс на базі БСА ПТП?
- 2) Які параметри необхідно відображати чисельно і які серед них мають мати шкалу?
- 3) Які параметри необхідно відображати за допомогою трендів. Який часовий діапазон цих трендів?

Дано відповідь на ці питання на прикладі ПТП розділення суміші метанол-вода.

- 1) Так, потрібно. Оскільки схема цілком зручна для розуміння технологічного процесу та локалізації місця помилки.
- 2) Все відображення має відображатись тільки на трендах з зображенням шкали та поточного значення параметра.
- 3) Усі параметри мають відображатись за допомогою трендів (діапазон 5 годин, тобто масштабовані 5 хв).

Приклад макета для екрана II рівня наведено на рисунку 4.2.

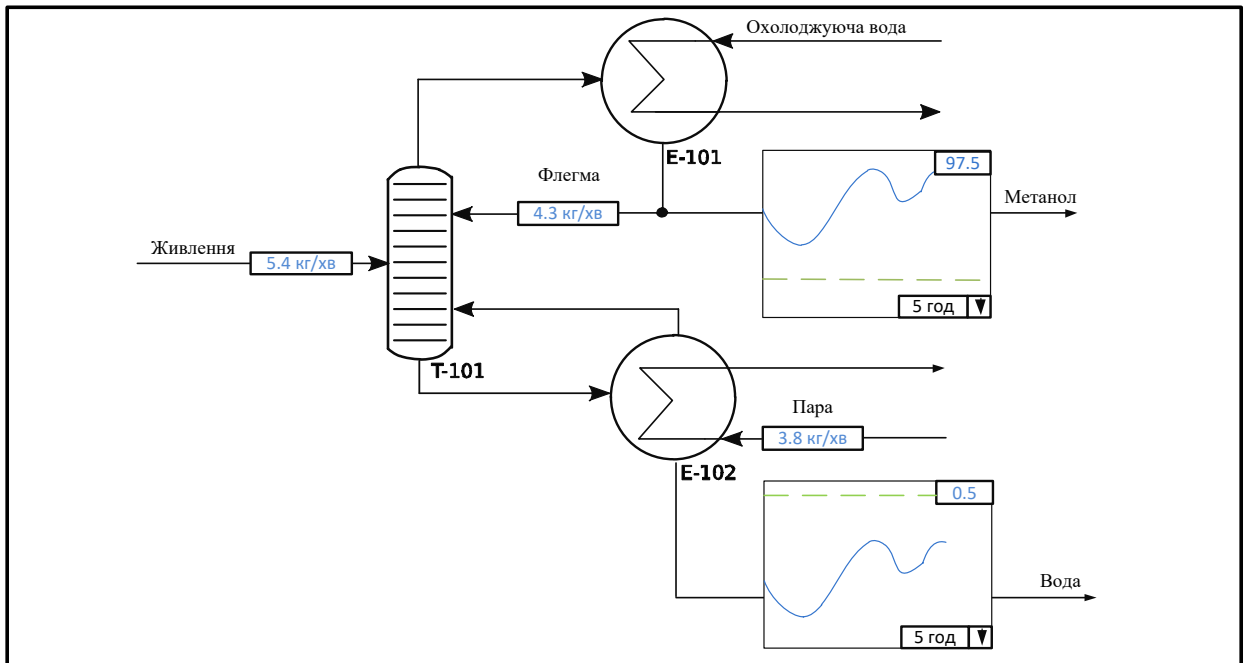


Рисунок 4.2 – Приклад макета екрана II рівня

4.4 Розробка макета екрана III рівня

Екран III рівня призначено для втручання оператора в хід ПТП.

Для розв'язку задачі розробки макета екрана III рівня необхідно дати відповідь на наступні запитання:

- 1) Чи потрібно організовувати інтерфейс на базі БСА ПТП?
- 2) Які параметри необхідно відображати чисельно і які серед них мають мати шкалу.
- 3) Які параметри необхідно відображати за допомогою трендів. Який часовий діапазон цих трендів.
- 4) Яким чином реалізувати зміну параметрів регуляторів та їх відключення?

Дамо відповідь на ці питання на прикладі ПТП розділення суміші метанол-вода.

- 1) Так, потрібно. Оскільки схема цілком зручна для розуміння технологічного процесу та локалізації місця помилки. Зображення техн. схеми на екрані цього рівня краще незначно змінити.
- 2) Оскільки кількість параметрів незначна усі з них можливо зображувати трендом з надписом й дублювати значення в панелях управління регуляторів.
- 3) Всі основні параметри з часових діапазон 5 год (в масштабованому часі - 5 хв.)

- 4) За допомогою таблиць, що пов'язують КЗ (PV) з КВ (OP). Оператор може змінювати завдання (SP) та вимикати регулятор.

Приклад макета екрана III рівня наведено на рисунку 4.3.

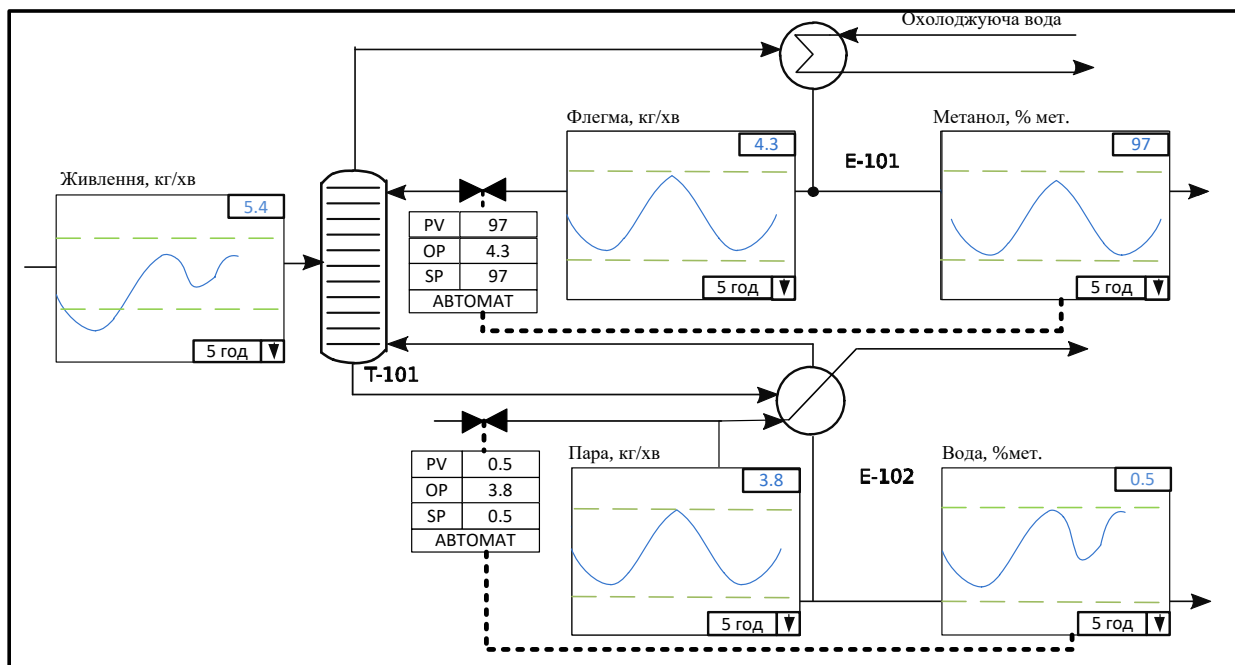


Рисунок 4.3 – Приклад екрана макета III рівня

5. РЕАЛІЗАЦІЯ ЛЮДИНО-МАШИННОГО ІНТЕРФЕЙСУ (ЛМІ) В SCADA-СИСТЕМІ SIMPLE SCADA

Реалізація ЛМІ проводиться в редакторі (editor) Simple SCADA, робота з яким розглянута в курсі практичних робіт.

Зверніть увагу, що оскільки в проекті застосовуються скрипти, то проект після збереження може автоматично не оновлюватись на сервері, оскільки старі періодичні скрипти продовжують працювати. Тому необхідно після збереження проекту в редакторі перед відкриттям в клієнті вручну зупиняти його виконання на сервері (програма server).

Процедура розробки проекту докладна розглянута в методичних рекомендаціях щодо виконання лабораторних робіт.

5.1 Налаштування переліку змінних проекту

Змінні проекту розділяються в Simple SCADA на два типи: зовнішні (пов'язані з OPC-сервером) та внутрішні (що використовуються тільки в межах проекту). Кожна змінна має мати відповідну шкалу, в якій вказується діапазон вимірювання, значення аварійні та передаварійні. Значення шкали мають братись з технологічного регламенту ведення процесу.

Для реалізації проекту ЛМІ СА розділення суміші метанола-вода необхідні шкали та змінні наведені в табл.5.1 й 5.2. Змінні для вашого варіанта реалізуються схожим чином.

Таблиця 5.1 – Перелік шкал Simple SCADA для ЛМІ СА розділення суміші метанол-вода

Ім'я шкали	Од.вим	Мін	Макс
reflux_scale	кг/хв	2,1	6,5
steam_scale	кг/хв	1,9	5,7
ctop_scale	%	90	100
cbtm_scale	%	0	0.9
feed_scale	кг/хв	2,7	8,1
kp_scale	%/кг/хв	-10	10
ki_scale	1/с	-1	1
ctop_scale_normed	%	0	2
cbtm_scale_normed	%	0	1

Таблица 5.2 – Перелік змінних Simple SCADA для ЛМІ СА розділення суміші метанол-вода

		Шкала							Границы			
Имя переменной	Тип данных	Имя шкалы	Ед. измерения	Минимум	Максимум	Описание	Тип тега	Адрес	ВА	ВП	НП	НА
var_u1	Double	reflux_scale	кг/хв	2.1	6.5	Storage register	внешний	storage.numeric.reg01	6.5	5.6	3	2.1
var_u2	Double	steam_scale				Storage register	внешний	storage.numeric.reg02	5.7	4.8	2.66	1.9
var_y1	Double	ctop_scale	%	90	100	Storage register	внешний	storage.numeric.reg03	98	97.5	96.5	96
var_y2	Double	cbtm_scale	%	0	0.9	Storage register	внешний	storage.numeric.reg04	0.95	0.75	0.25	0.05
var_feed	Double	feed_scale	кг/хв	2.7	8.1	Storage register	внешний	storage.numeric.reg05	8.1	7	3.8	2.7
var_running	Boolean					Storage register	внешний	storage.bool.reg01				
var_u1_fail	Boolean					Storage register	внешний	storage.bool.reg02				
var_u2_fail	Boolean					Storage register	внешний	storage.bool.reg03				
PID1_Kp	Double	kp_scale	%	-10	10		внутренний					
PID1_Ki	Double	ki_scale	1/s	-1	1		внутренний					
PID1_PrevInput	Double						внутренний					
PID1_MaxOutput	Double						внутренний					
PID1_MinOutput	Double						внутренний					
PID1_First_Time	Boolean						внутренний					
PID1_Int_Improvement	Boolean						внутренний					
PID1_Diff_Improvement	Boolean						внутренний					
PID1_Prev_Abs_Error	Double						внутренний					
PID1_Integrated	Double						внутренний					
PID1_OutputNominal	Double						внутренний					

PID2_Kp	Double	kp_scale	%	-10	10		внутренний					
PID2_Ki	Double	ki_scale	1/s	-1	1		внутренний					
PID2_PrevInput	Double						внутренний					
PID2_MaxOutput	Double						внутренний					
PID2_MinOutput	Double						внутренний					
PID2_First_Time	Boolean						внутренний					
PID2_Int_Improvement	Boolean						внутренний					
PID2_Diff_Improvement	Boolean						внутренний					
PID2_Prev_Abs_Error	Double						внутренний					
PID2_Integrated	Double						внутренний					
PID2_OutputNominal	Double						внутренний					
var_regon1	Boolean						внутренний					
var_regon2	Boolean						внутренний					
efficiency	Single						внутренний					
counter_alarms	Int64						внутренний					
counter_warnings	Int64						внутренний					
state_steam	Integer						внутренний					
state_reflux	Integer						внутренний					
state_feed	Integer						внутренний					
state_ctop	Integer						внутренний					
state_cbtm	Integer						внутренний					
state_reg1	Integer						внутренний					
state_reg2	Integer						внутренний					
var_sp1	Double						внутренний					
var_sp2	Double						внутренний					
var_y1_normed	Double	ctop_scale_normed	%	0	2		внутренний					
var_y2_normed	Double	cbtm_scale_normed	%	0	1		внутренний					

Для спрощення відтворення прикладу можливо використовувати CSV файл з додатка А.

5.2 Розробка програми для ПД-регулювання

Мовою програмування Simple SCADA є мова Object Pascal. В проект додаються скрипти цією мовою, що мають певний фіксований інтервал виконання. Кожен скрипт компілюється окремо.

В цьому розділі буде описано як реалізувати САК з ПД-регуляторами в Simple SCADA. Для відлагодження програми створимо простий екран відлагодження, що не входить до переліку необхідних в ЛМІ.

Перейменуємо сторінку, що додана в новий проект автоматично, в «Регулювання».

Розташуємо в ній елементи аналогічно до рис. 5.1.

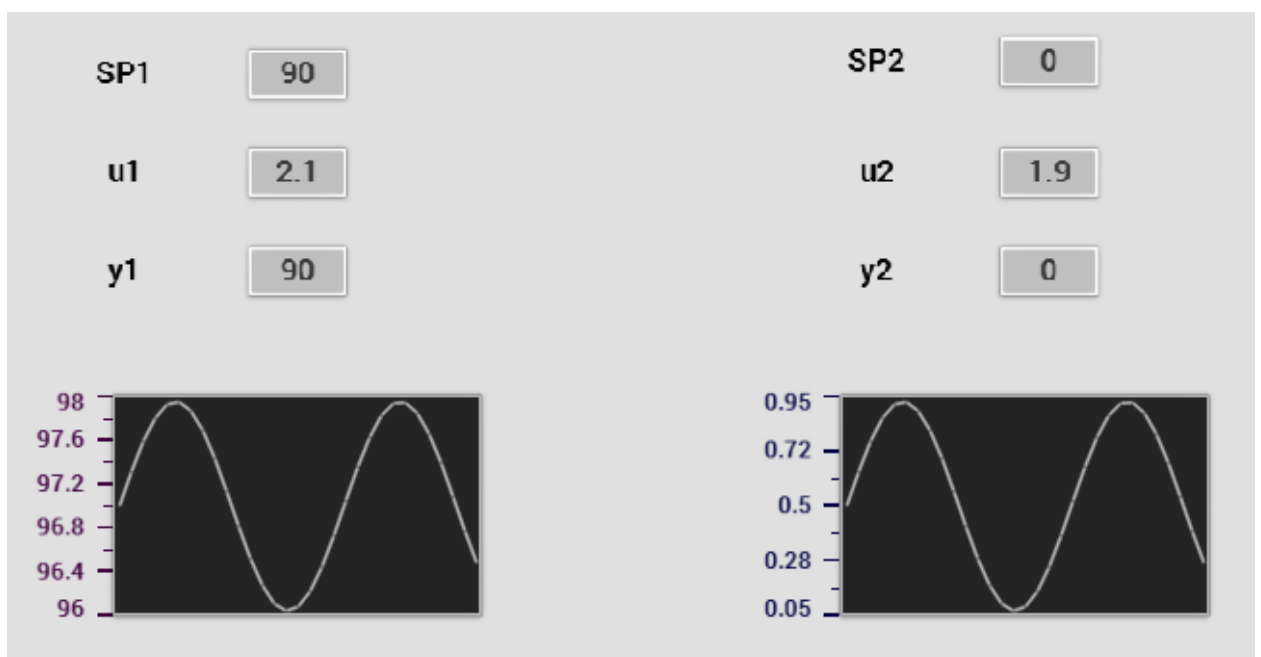


Рисунок 5.1 – Екранна форма відлагодження роботи регуляторів в редакторі Simple SCADA

Розглянемо ліву частину. Поле завдання на 1-у КЗ – концентрацію в верхньому продукті (SP1) зв'язується з полем var_sp1, поле КВ (u1) зв'язується з var_u1, поле КЗ і лівий тренд зв'язується з var_y1. Права частина виконується аналогічно відносно var_sp2, var_u2 та var_y2.

Додамо скрипт initPID, що виконується при запуску проекту. Цей скрипт ініціалізує параметри регуляторів.

Код налаштування параметрів регулятора концентрації верхнього продукту, параметри якого отримані в розділі 2. 2, має наступний вигляд

```
procedure initPID (Sender: TM_Control);  
begin
```

```

var_sp1.Value := 97;
PID1_Kp.Value := 0.749;
PID1_Ki.Value := 0.0589;
PID1_Kd.Value := 0;

PID1_Output_Nominal.Value := 4.3;
PID1_MaxOutput.Value := 4.3*2;
PID1_MinOutput.Value := 0;
PID1_First_Time.Value := True;
PID1_Int_Improvement.Value := False;
PID1_Diff_Improvement.Value := False;
Pid1_Prev_Abs_Error.Value := 0;
PID1_Integrated.Value :=0;

state_reg1.Value := True;
end.

```

Код налаштування параметрів регулятора концентрації нижнього продукту виконується аналогічно

Додамо скрипт PIDCount, що виконується кожну секунду виконання проекту . Код скрипта наведено в додатку Б. Код скрипта в вашому проекті може бути застосований без змін.

Збережемо проект та відкриємо його в клієнті. Результат виконання має бути аналогічним, показаному на рисунку 5.2.

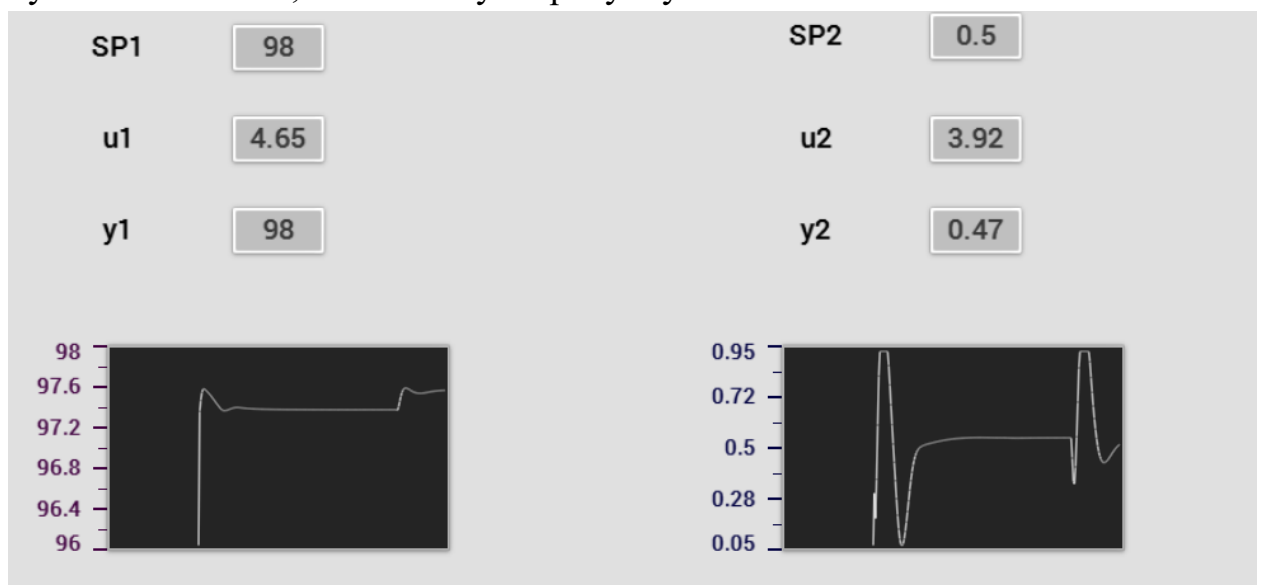


Рисунок 5.2 – Екранна форма виконання в клієнті після зменшення завдання першого регулятора і повертання його на попереднє значення

Відмітимо, що запропонований підхід є гарним лише для учбової задачі. В реальній задачі з метою збільшення надійності СА задачу ПД-регулювання необхідно реалізовувати за допомогою ПЛК чи окремої програми, що працює в фоновому режимі на серверному комп'ютері.

5.3 Підготовка зображень

Для реалізації ЛМІ необхідно підготувати наступні зображення:

- 1) Зображення стрілки для індикатора ПС
- 2) Зображення сигналізатора помилки
- 3) Зображення технологічної схеми для екрана II рівня
- 4) Зображення схеми з вказування контурів керування для екрана III рівня

Зображення виконуються в графічному редакторі і мають бути збережені в форматі PNG і покладені в папку Pictures SCADA-системи.

Приклад зображення стрілки показано на рис. 5.3.



Рисунок 5.3 – Приклад зображення стрілки в індикаторі ПС
(називається arrow2.png)

Приклад зображення сигналізатора, що складається з двох кадрів, показано на рис. 5.4.



Рисунок 5.4 – Приклад зображення сигналізатора
(називається indicator4.png)

За допомогою програми images.exe зображення сигналізатора розбивається на кадри.

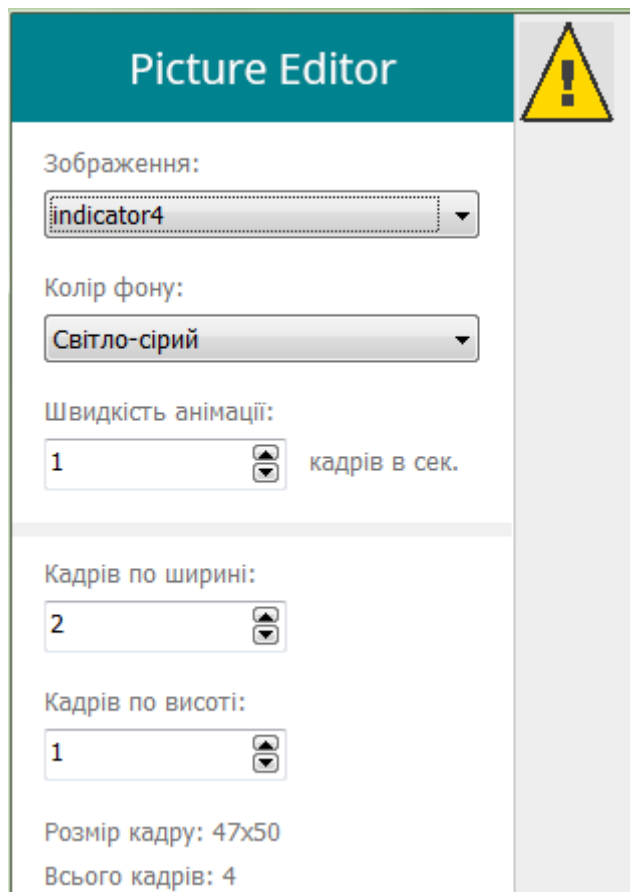


Рисунок 5.5 – Розбивка зображення сигналізатора на кадри

Зображення технологічних схем виконується на прозорому фоні за допомогою темно-сірих ліній. Орієнтовне розширення зображення – 786x455. Зображення не має мати тексту, трендів, кнопок та ін., тобто елементів SCADA-системи. В якості кольору ліній використовується колір з RGB(79, 79, 79).

Приклад зображення технологічної схеми для екрана II рівня показаний на рис. 5.6.

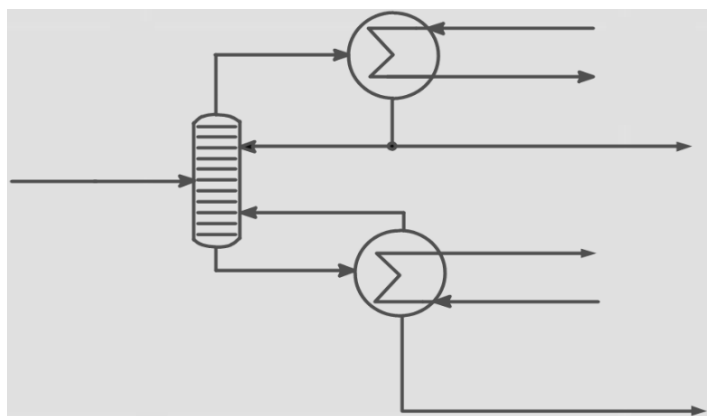


Рисунок 5.6 – Зображення технологічної схеми для екрана II рівня (файл dc_level2.png)

Приклад зображення технологічної схеми для екрана III рівня показаний на рис. 5.7.

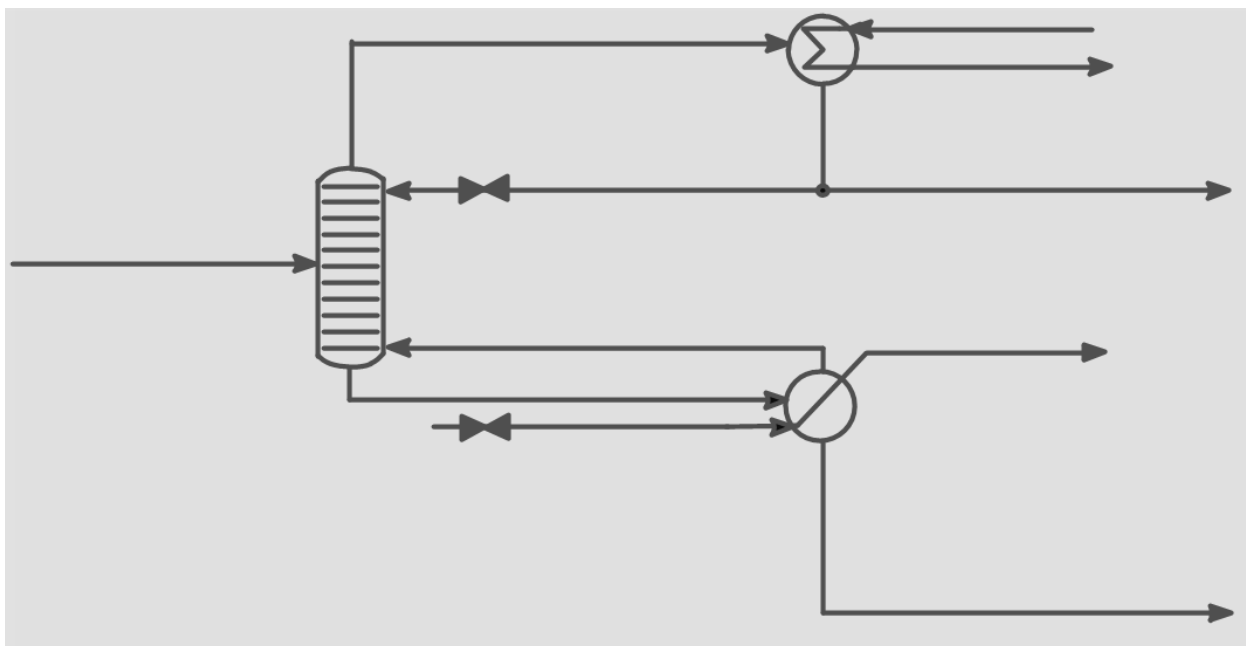


Рисунок 5.7 – Зображення технологічної схеми для екрана III рівня (файл dc_level3.png)

5.4 Розробка екрана I рівня

Встановимо схему кольорів проекта «Світло-сірий 13%» та розширення 5:3 (1280x768).

Додамо нову сторінку –«Загальний стан».

Розділимо сторінку на 5 зон за допомогою товстої пунктирної лінії (переривчата лінія з товщиною 4px).

Зона 1 – Показники виробництва. Підпишімо першу зону текстовим написом «ОСНОВНІ ПОКАЗНИКИ ВИРОБНИЦТВА», також добавимо в неї підписи «ЗМІНА», «ГОДИНА», та підписи назв ПС.

Розглянемо задачу відображення індикатора ПС на прикладі реалізації індикатора кількості розділеної суміші в ПТП розділення суміші метанол-вода. Для розуміння спростимо її – будемо відображати не середнє значення за період, а поточне значення.

Для цього необхідно додати елементи SCADA - поле, шкалу, зображення та лінію.

Додамо поле приблизно розміром 136x56. Назвемо його kpi_feed_h. Встановимо колір шрифту - синій, колір рамки – чорний. Розмір шрифту зменшимо, щоб залишилось місце для шкали, орієнтовно до 24 px.

Додамо шкалу. Назвемо її `kpi_feed_h_scale`. Зробимо положення вертикальним. Знімимо підписи. Поставимо кількість міток – 9. Поставимо шкалу внизу поля.

Додаю зображення та поставимо текстуру – `arrow2`. Назвемо зображення `kpi_feed_h_arrow`. Змасштабуємо зображення та поставимо в центр знизу поля. Для точності будемо використовувати властивості X та Y.

Додамо зелену лінію товщиною `4px kpi_feed_h_line` та розмістимо її по середині шкали над стрілкою.

Індикатор, що додано, має бути аналогічним тому, що зображено на рис. 5.8.

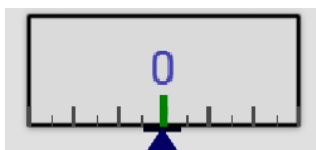


Рисунок 5.8 – Еталонне зображення індикатора ПС

Збережемо проект та відкриємо його в клієнті. Перевіримо за допомогою Matrikon OPC Explorer що стрілка та світова індикація працює коректно (див. рис. 5.9).

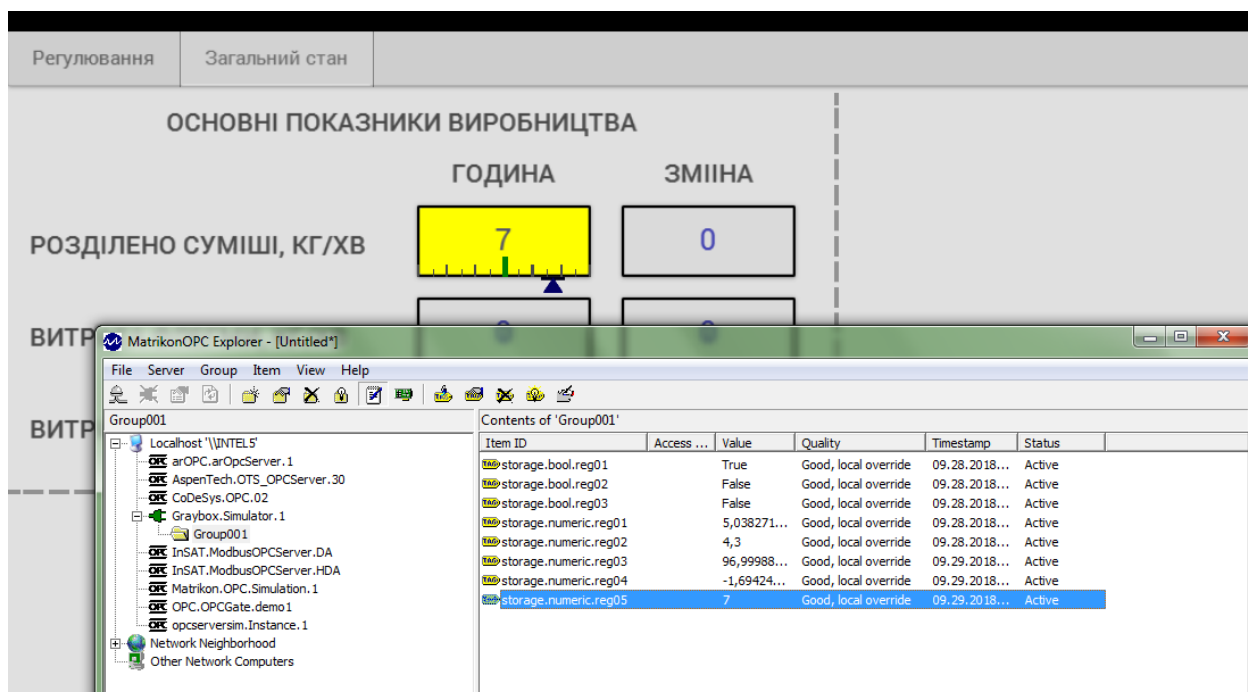


Рисунок 5.9 – Відлагодження індикатора ПС

Розробка та відображення інших індикаторів ПС виконується аналогічно.

Відмітимо, що розраховувати повноцінно середнє значення в Simple SCADA можливо, але тільки коли проект зв'язується з сервером баз даних і сервер автоматично фіксує параметри кожен заданий інтервал часу. Тоді ці значення з легкістю розраховуються за допомогою SQL-запиту.

Зона 2 – Сигналізатори параметрів. Встановимо сигналізатори пари, флегми, суміші, концентрації верха, низу за допомогою інструмента «фігура» (TM_Shape). Надамо їм наступні імена – lamp_steam, lamp_reflux, lamp_feed, lamp_stop, lamp_cbtm. Товщина межі для всіх фігур – 2, колір межі – чорний.

Встановимо сигналізатори помилки справа від кожної лінії як зображення з текстурою indicator4: ind1_main, ind1_feed, ind1_conc. Таким чином, частина форми має мати вигляд, аналогічний до рис. 5.10.



Рисунок 5.10 – Зона сигналізатора параметрів в редакторі

Додамо скрипт lampUpdate, що виконується кожну секунду виконання проекту. Код скрипта наведено в додатку Г.

Основною внутрішньою функцією скрипта є функція indicateLamp, що має наступні параметри: lamp – ідентифікатор сигналізатора (TM_SHAPE), stateVar – змінна, що зберігає стан індикації, errorValue – значення модуля невідповідності бажаного та дійсного значення змінною, alarmErrorValue – аварійне значення різниці, warningErrorValue – попереджувальне значення різниці; deadZoneValue – порог нечутливості; alarmMessage – повідомлення аварійне; warningMessage – повідомлення попереджувальне, normMessage – повідомлення повернення в нормальний стан параметру);

Приклад індикації при збуренні за витратою флегми на 1 кг/хв показано на рис. 5.11. Оператор має підтверджувати кожне повідомлення. До моменту поки повідомлення не підтверджується повторюється звукова сигналізація.

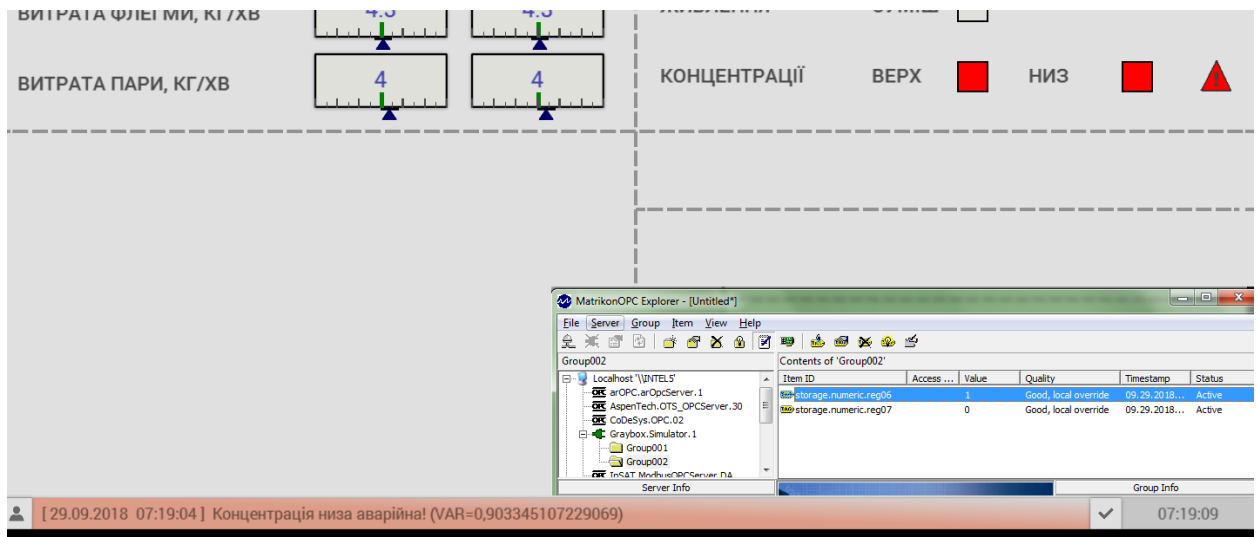


Рисунок 5.11 – Аварійна сигналізація за відхиленням концентрацій

Зона 3 – Сигналізатор стану регуляторів. Обмежимося лише сигналізацією стану роботи регуляторів. Якщо регулятор в ручному режимі, то використовується голубий колір для індикації. Додамо сигналізатори lamp_ac101 та lamp_ac102.

```
В скрипт lampUpdate додамо в кінець наступний код
if state_reg1.Value = False then begin
    lamp_ac101.Color := clSkyBlue;
end
else begin
    lamp_ac101.Color:=RGB(224,224,218);
end;
```

Код для AC102 виконується аналогічно.

Зона 4 – Статистика технологічних ситуацій. Як і для зони 1 приймемо спрощення - будемо відображати лише поточний стані змінних. Тоді необхідно додати 4 поля і підписати їх текстовими надписами. Поля зв'язуються зі змінними counter_alarms (лічильний аварійних повідомлень) та counter_warnings (лічильник передаварійних повідомлень).

Зона 5 – Зона показу тренду енергоефективності. Тренд необхідно зв'язати зі змінною efficiency. Оформлення зони має бути відповідно до рис. 5.12. Тобто необхідно задіяти три елемента – тренд, поле (зверху) і текстове поле (в якому просто пишеться час тренда). Для спрощення задачі перемикач часу реалізовувати не будемо, оскільки в Simple SCADA воно доволі складно вирішується. Крім того, значення ефективності будемо

нормувати до 100, оскільки Simple SCADA некоректно масштабує нецілі значення.



Рисунок 5.12 – Приклад оформлення нижньої частини екрана

Розрахунок ефективності реалізуємо в скрипті countEfficiency, що буде складатись з однієї строки в розглянутому випадку.

```
efficiency.Value := 100*(-0.4752 *
(var_u1.Value/var_feed.Value) + 0.7042 -
0.5388 * (var_u2.Value/var_feed.Value)+0.7042);
```

Приклад роботи екрана першого рівня показано на рисунку 5.13.

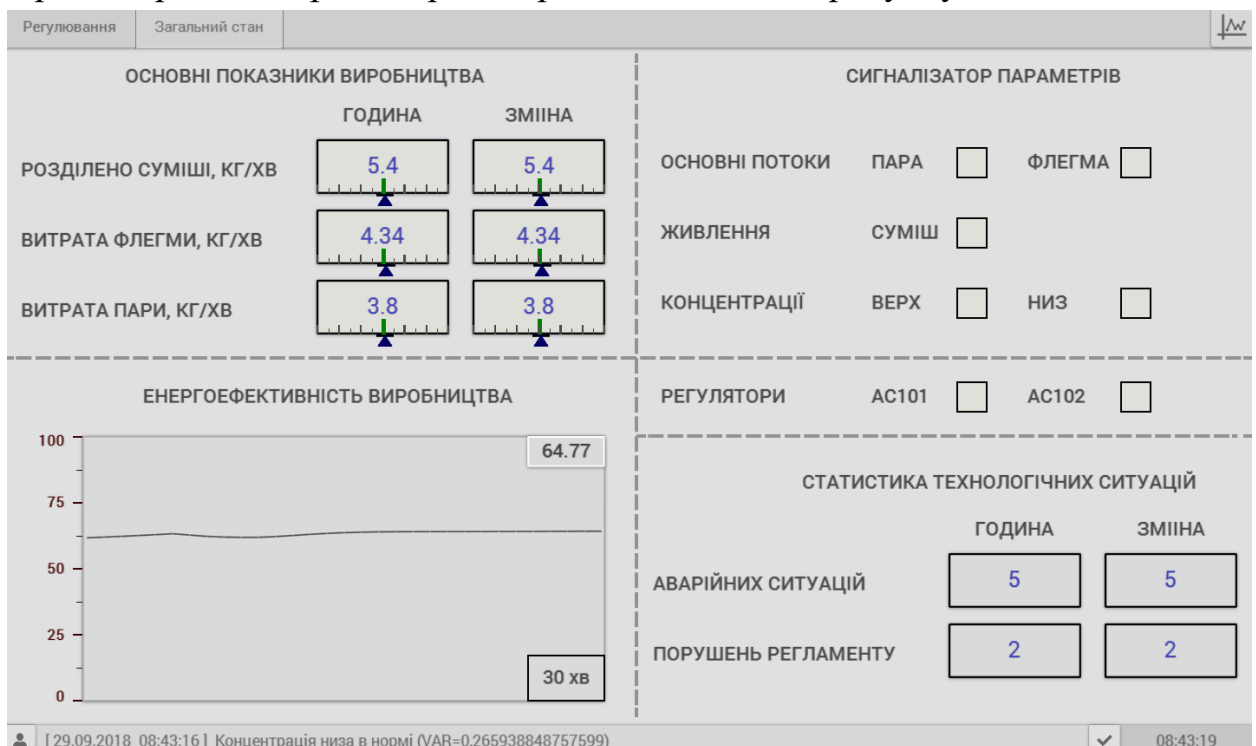


Рисунок 5.13 – Виконання екрана 1 рівня в клієнті

5.5 Розробка екрана II рівня

Екран другого рівня реалізується простіше. Додамо сторінку та назвемо її «Основний екран». Додамо зображення з текстурою `dc_level2` й зробимо необхідні підписи. Також додамо індикатори поряд з кожним параметром й назвемо їх відповідно `ind2_feed`, `ind2_steam`, `ind2_reflux`, `ind2_ctop`, `ind2_cbtm`. Тренди прив'яжимо до змінних `var_y1_normed`, `var_y2_normed` з діапазонами `ctop_scale` та `cbtm_scale`.

Таким чином, екранна форма буде мати наступний вигляд.

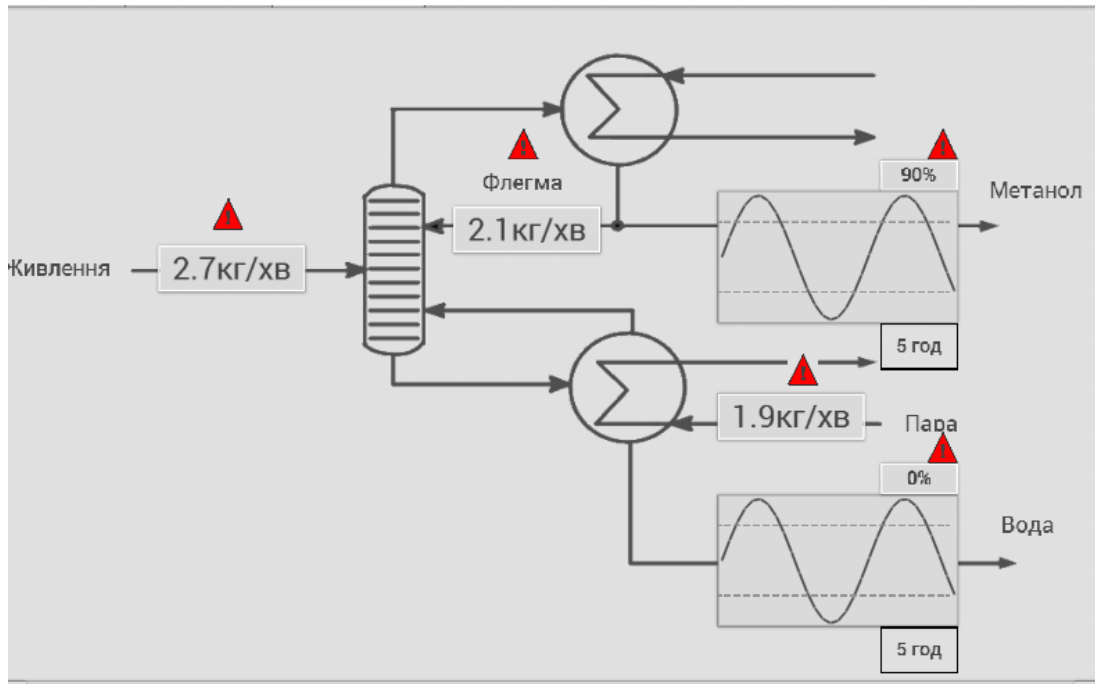


Рисунок 5.14 – Екран II рівня в редакторі

Скрипт, що виконується кожну секунду, `level2` має реалізовувати 2 задачі: нормування значень для трендів та показ індикаторів, якщо спрацьовують відповідні сигналізації на екрані I рівня.

Масштабування виконаємо наступним чином

```
var_y1_normed.Value := var_y1.Value-97+1;  
var_y2_normed.Value := var_y2.Value-0.5+0.5;
```

Задачу індикації розглянемо на прикладі індикатора флегми.

```
if state_reflux.Value = 1 then begin  
    ind2_reflux.Visible:=True; ind2_reflux.Frame:=2;  
end;  
if state_reflux.Value = 2 then begin  
    ind2_reflux.Visible:=True; ind2_reflux.Frame:=1;  
end;  
if state_reflux.Value = 0 then begin  
    ind2_reflux.Visible:=False;  
end;
```

Інші індикатори реалізуються аналогічно. Працюючий інтерфейс при дії збурення, що викликало сигналізацію, показаний на рисунку 5.15.

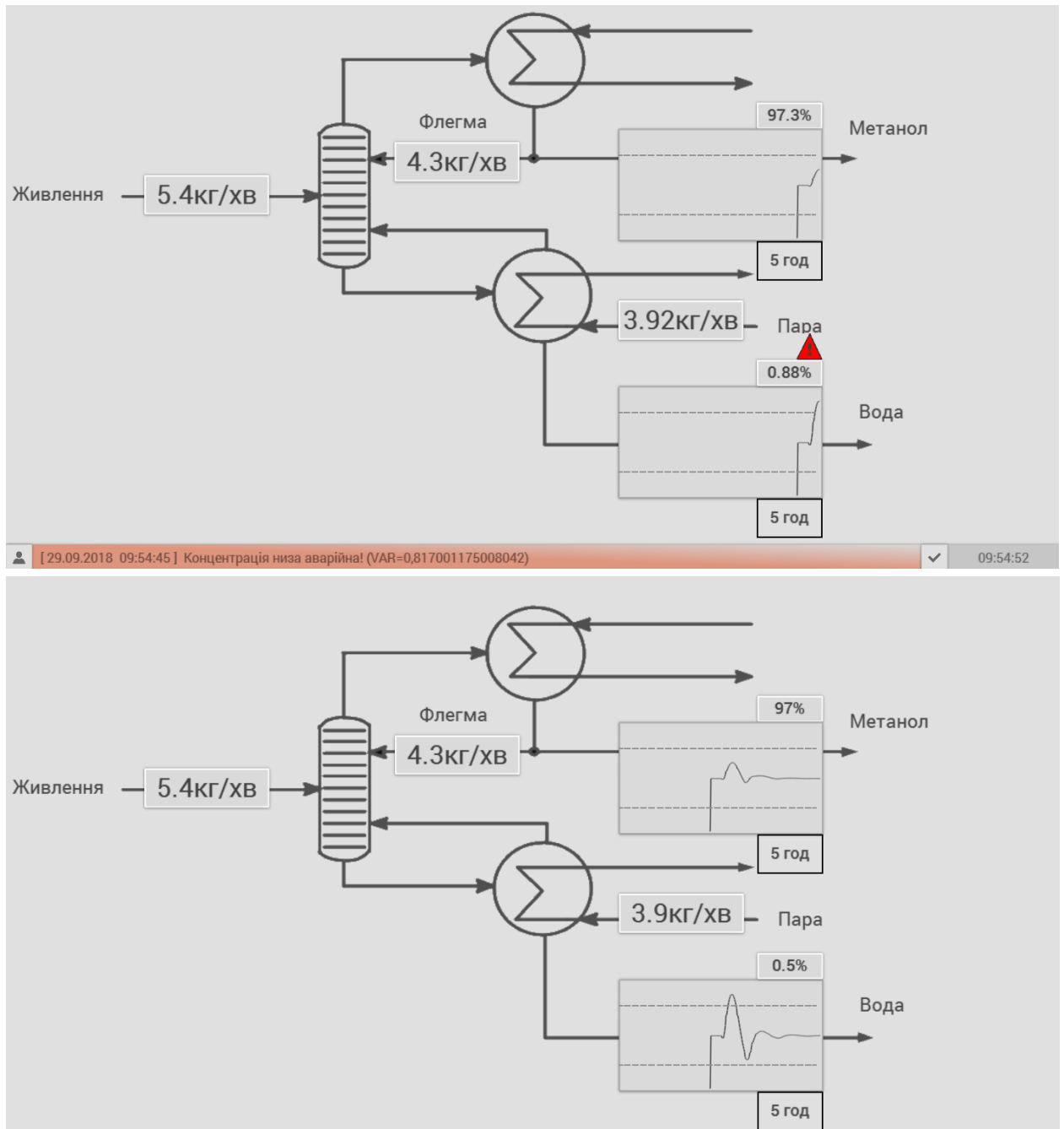


Рисунок 5.15 – Працюючий екран II рівня при дії збурення та після його подолання

5.6 Розробка екрана III рівня

Виконується аналогічним чином, як й екран II рівня. З метою зменшення часу розробки й складності програми в завданні достатня проробка тільки макета екрана III рівня й зображення технологічної схеми для екрана. Програмна реалізація не обов'язкова.

ДОДАТКИ

Додаток А. CSV файл з переліком тегів

```
;;;Архив;;;Шкала;;;;;;;;;Границы;;;Режим масштабирования;
Имя группы;Имя переменной;Тип данных;Тип архивации;Тип тренда;Зона нечувствительности;Интервал
архивации;Имя шкалы;Ед. измерения;Минимум;Максимум;Описание;Тип тега;Адрес;Имя ПК;ОПС-
сервер;Частота опроса тегов;Формат;Сдвиг;ВА;ВП;НП;НА;Виз. минимум;Виз. максимум
..\;;;;;;;;;
;var_u1;Double;не архивировать;обычный;0;как в настройках;reflux_scale;кг/хв;2.1;6.5;Storage
register;внешний;storage.numeric.reg01;localhost;Graybox.Simulator.1;как в настройках;0.##;0;6.5;5.6;3;2.1;;
;var_u2;Double;не архивировать;обычный;0;как в настройках;;;Storage
register;внешний;storage.numeric.reg02;localhost;Graybox.Simulator.1;как в настройках;0.##;0;5.7;4.8;2.66;1.9;;
;var_y1;Double;не архивировать;обычный;0;как в настройках;ctop_scale;%;90;100;Storage
register;внешний;storage.numeric.reg03;localhost;Graybox.Simulator.1;как в настройках;0.##;0;98;97.5;96.5;96;;
;var_y2;Double;не архивировать;обычный;0;как в настройках;cbtm_scale;%;0;0.9;Storage
register;внешний;storage.numeric.reg04;localhost;Graybox.Simulator.1;как в
настройках;0.##;0;0.95;0.75;0.25;0.05;;
;var_feed;Double;не архивировать;обычный;0;как в настройках;feed_scale;кг/хв;2.7;8.1;Storage
register;внешний;storage.numeric.reg05;localhost;Graybox.Simulator.1;как в настройках;0.##;0;8.1;7;3.8;2.7;;
;var_running;Boolean;не архивировать;обычный;0;как в настройках;;;Storage
register;внешний;storage.bool.reg01;localhost;Graybox.Simulator.1;как в настройках;0.##;0;,,,,;
;var_u1_fail;Boolean;не архивировать;обычный;0;как в настройках;;;Storage
register;внешний;storage.bool.reg02;localhost;Graybox.Simulator.1;как в настройках;0.##;0;,,,,;
;var_u2_fail;Boolean;не архивировать;обычный;0;как в настройках;;;Storage
register;внешний;storage.bool.reg03;localhost;Graybox.Simulator.1;как в настройках;0.##;0;,,,,;
;PID1_Kp;Double;не архивировать;обычный;0;как в настройках;kp_scale;%;-10;10;;внутренний;;;как в
настройках;0.##;0;,,,,;
;PID1_Ki;Double;не архивировать;обычный;0;как в настройках;ki_scale;1/s;-1;1;;внутренний;;;как в
настройках;0.##;0;,,,,;
;PID1_PrevInput;Double;не архивировать;обычный;0;как в настройках;;;внутренний;;;как в
настройках;0.##;0;,,,,;
;PID1_MaxOutput;Double;не архивировать;обычный;0;как в настройках;;;внутренний;;;как в
настройках;0.##;0;,,,,;
;PID1_MinOutput;Double;не архивировать;обычный;0;как в настройках;;;внутренний;;;как в
настройках;0.##;0;,,,,;
;PID1_First_Time;Boolean;не архивировать;обычный;0;как в настройках;;;внутренний;;;как в
настройках;0.##;0;,,,,;
;PID1_Int_Improvement;Boolean;не архивировать;обычный;0;как в настройках;;;внутренний;;;как в
настройках;0.##;0;,,,,;
;PID1_Diff_Improvement;Boolean;не архивировать;обычный;0;как в настройках;;;внутренний;;;как в
настройках;0.##;0;,,,,;
;PID1_Prev_Abs_Error;Double;не архивировать;обычный;0;как в настройках;;;внутренний;;;как в
настройках;0.##;0;,,,,;
;PID1_Integrated;Double;не архивировать;обычный;0;как в настройках;;;внутренний;;;как в
настройках;0.##;0;,,,,;
;PID1_OutputNominal;Double;не архивировать;обычный;0;как в настройках;;;внутренний;;;как в
настройках;0.##;0;,,,,;
;PID2_Kp;Double;не архивировать;обычный;0;как в настройках;kp_scale;%;-10;10;;внутренний;;;как в
настройках;0.##;0;,,,,;
;PID2_Ki;Double;не архивировать;обычный;0;как в настройках;ki_scale;1/s;-1;1;;внутренний;;;как в
настройках;0.##;0;,,,,;
```

```

;PID2_PrevInput;Double;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;PID2_MaxOutput;Double;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;PID2_MinOutput;Double;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;PID2_First_Time;Boolean;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;PID2_Int_Improvement;Boolean;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;PID2_Diff_Improvement;Boolean;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;PID2_Prev_Abs_Error;Double;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;PID2_Integrated;Double;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;PID2_OutputNominal;Double;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;var_regon1;Boolean;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;var_regon2;Boolean;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;efficiency;Single;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в настройках;0.##;0;;;;;
;counter_alarms;Int64;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;counter_warnings;Int64;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;state_steam;Integer;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;state_reflux;Integer;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;state_feed;Integer;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в настройках;0.##;0;;;;;
;state_ctop;Integer;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в настройках;0.##;0;;;;;
;state_cbtm;Integer;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в
настройках;0.##;0;;;;;
;state_reg1;Integer;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в настройках;0.##;0;;;;;
;state_reg2;Integer;не архивировать;обычный;0;как в настройках;;;;;внутренний;;;как в настройках;0.##;0;;;;;
;var_sp1;Double;не архивировать;обычный;0;как в настройках;ctop_scale;%;90;100;;внутренний;;;как в
настройках;0.##;0;;;;;
;var_sp2;Double;не архивировать;обычный;0;как в настройках;cbtm_scale;%;0;0.9;;внутренний;;;как в
настройках;0.##;0;;;;;
;var_y1_normed;Double;не архивировать;обычный;0;как в
настройках;ctop_scale_normed;%;0;2;;внутренний;;;как в настройках;0.##;0;;;;;
;var_y2_normed;Double;не архивировать;обычный;0;как в
настройках;cbtm_scale_normed;%;0;1;;внутренний;;;как в настройках;0.##;0;;;;;

```

Додаток Б. Код скрипта ПІД-регулятора

```
procedure PID_Count (Sender: TM_Control);
    function AbsReal(Vall:Real):Real;
    var
        Res:Real;
    begin
        Res := Vall;
        if Res < 0 then Res := -Res;
        AbsReal := Res;
    end;

    function Sign (Value:Real):Byte;
    var
        Res:Byte;
    begin
        Res := 0;
        if (Value >=0) then Res := 1;
        Sign := Res;
    end;

    procedure      nextControl(SP:TM_VARIABLE;      y:TM_VARIABLE;
u:TM_VARIABLE;
        PID_Kp:TM_VARIABLE; PID_Ki:TM_VARIABLE; PID_Kd:TM_VARIABLE;
        PID_Int_Improvement:TM_VARIABLE;PID_Integrated:TM_VARIABLE;
        PID_Diff_Improvement: TM_VARIABLE;
        PID_MinOutput:TM_VARIABLE;      PID_MaxOutput:TM_VARIABLE;
        PID_Output_Nominal:TM_VARIABLE;
        PID_First_Time:TM_VARIABLE;      PID_PrevInput:TM_VARIABLE;
        PID_Prev_Abs_Error:TM_VARIABLE);
    var
        Err, ErrValue, DiffValue, ErrAbs, Res: Real;
    begin
        Err := SP.Value - y.Value;
        if PID_Diff_Improvement.Value = True then
            begin
                ErrAbs := AbsReal(Err);
            end;
        ErrValue := Err * PID_Kp.Value;

        if PID_Int_Improvement.Value = True then
            begin
                if Sign(Err) <> Sign(PID_Integrated.Value) then
                    begin
                        PID_Integrated.Value := 0.0;
                    end;
            end;
        end;
```



```

end;

PID_Integrated.Value := PID_Integrated.Value + (Err *
PID_Ki.Value);
if PID_Integrated.Value < PID_MinOutput.Value then
begin
    PID_Integrated.Value := PID_MinOutput.Value;
end;

if PID_Integrated.Value > PID_MaxOutput.Value then
begin
    PID_Integrated.Value := PID_MaxOutput.Value;
end;

if PID_First_Time.Value = True then
begin
    PID_First_Time.Value      := False;
    PID_PrevInput.Value       := y.Value;
    PID_Prev_Abs_Error.Value  := 0;
end;

DiffValue := (y.Value - PID_PrevInput.Value) *
PID_Kd.Value;
PID_PrevInput.Value := y.Value;
if PID_Diff_Improvement.Value = True then
begin
    if ErrAbs < PID_Prev_Abs_Error.Value then DiffValue
:= 0;
    begin
        PID_Prev_Abs_Error.Value := ErrAbs;
    end;
end;

//    u.Value := ErrValue + PID_Output_Nominal.Value;

Res := ErrValue + PID_Integrated.Value - DiffValue;

if Res < PID_MinOutput.Value then
begin
    Res := PID_MinOutput.Value;
end;
if Res > PID_MaxOutput.Value then
begin
    Res := PID_MaxOutput.Value;
end;

```

```

        u.Value := Res + PID_Output_Nominal.Value;
//      end;
end;

begin
if var_y1.IsGoodQuality=True then begin

    if state_reg1.Value = True then begin
        nextControl(var_sp1, var_y1, var_u1,
            PID1_Kp, PID1_Ki, PID1_Kd,
            PID1_Int_Improvement, PID1_Integrated,
            PID1_Diff_Improvement,
            PID1_MinOutput, PID1_MaxOutput, PID1_Output_Nominal,
            PID1_First_Time, PID1_PrevInput, PID1_Prev_Abs_Error);
    end;
//аналогічно код для другого регулятора
end;
end.

```

Додаток В. Код скрипта для відображення ПС

```
procedure KPIUpdate (Sender: TM_Control);
procedure ChangeKPI(reg:TM_VARIABLE; nominal: Real;
kpi_scale:TM_SCALE; kpi_field:TM_FIELD; kpi_line:TM_LINE;
kpi_arrow:TM_IMAGE);
var
    valueInScale:Real;
    nominalInScale:Real;
    diapason:Real;
    barWidth:Real;
begin
    kpi_field.Variable := reg;
    valueInScale:=reg.Value;
    if reg.HighAlarm < reg.Value then
        valueInScale:= reg.HighAlarm;
    if reg.LowAlarm > reg.Value then
        valueInScale:=reg.LowAlarm;

    kpi_field.Color := RGB(224,224,218);

    if reg.Value >= reg.HighAlarm then
        begin
            kpi_field.Color := clRed;
        end
    else
        begin
            if reg.Value >= reg.HighWarning then
                begin
                    kpi_field.Color := clYellow;
                end
            end;
        if reg.Value <= reg.LowAlarm then
            begin
                kpi_field.Color := clRed;
            end
        else
            begin
                if reg.Value <= reg.LowWarning then
                    begin
                        kpi_field.Color := clYellow;
                    end
                end;
            end;

        diapason := reg.HighAlarm-reg.LowAlarm;
```

```

ValueInScale := (ValueInScale - abs(reg.LowAlarm))/diapason;
NominalInScale := (nominal - abs(reg.LowAlarm))/diapason;

barWidth := (kpi_scale.W-14);

kpi_arrow.X := kpi_scale.X + barWidth*(valueInScale);
kpi_line.X := kpi_scale.X + barWidth*(nominalInScale);
end;
begin

ChangeKPI(var_feed,5.4,kpi_feed_h_scale,kpi_feed_h,
kpi_feed_h_line,kpi_feed_h_arrow);
// відображення інших індикаторів виконується аналогічно
end.

```

Додаток Г. Код скрипта для сигналізації параметрів

```
procedure lampUpdate(Sender:TM_Control);
procedure indicateLamp (lamp:TM_SHAPE; stateVar:TM_VARIABLE;
curVariable:TM_VARIABLE; errorValue:Real; alarmErrorValue:Real;
warningErrorValue:Real; deadZoneValue:Real; alarmMessage:String;
warningMessage:String;
normMessage:String);
var
    state_before:Integer;
    state_after:Integer;
    greyColor: Cardinal;
begin
    greyColor := RGB(224,224,218);
    if curVariable.IsGoodQuality then begin
        state_before:=stateVar.Value;
        if alarmErrorValue<=errorValue then begin
            stateVar.Value:=2;
        end
        else begin
            if errorValue < (alarmErrorValue-deadZoneValue) then begin
                if errorValue>(warningErrorValue+deadZoneValue) then
begin
                    stateVar.Value:=1;
                end
                else begin
                    stateVar.Value:=0;
                end
            end;
        end;
        state_after := stateVar.Value;

        if stateVar.Value=1 then begin
            lamp.Color:=clYellow;
        end;
        if stateVar.Value=2 then begin
            lamp.Color:=clRed;
        end;
        if stateVar.Value=0 then begin
            lamp.Color:=greyColor;
        end;

        if (state_after<>state_before) then begin
            if stateVar.Value=1 then begin
```

```

        counter_warnings.Value := counter_warnings.Value +
1;
        AddMessage (Now,mkWarning,warningMessage+ '
(VAR='+curVariable.Value+') ',True,True);
    end;
    if stateVar.Value=2 then begin
        counter_alarms.Value := counter_alarms.Value + 1;
        AddMessage (Now,mkAlarm,alarmMessage+ '
(VAR='+curVariable.Value + ') ',True,True);
    end;
    if stateVar.Value=0 then begin
        AddMessage (Now,mkMessage, normMessage + '
(VAR='+curVariable.Value + ') ',
        True,True);
    end;
end;
end;
end;
begin

    indicateLamp (lamp_reflux, state_reflux, var_u1,
Abs(var_u1.Value-4.3), 4.3*0.5, 4.3*0.3, 4.3*0.05,
'Витрата флегми аварійна!','Витрата флегми вийшла за межі
регламенту!',
'Витрата флегми в нормі');
    indicateLamp (lamp_steam, state_steam, var_u2,
Abs(var_u2.Value-3.8), 3.8*0.5, 3.8*0.3, 3.8*0.05,
'Витрата пари аварійна!','Витрата пари вийшла за межі
регламенту!',
'Витрата пари в нормі');
    indicateLamp (lamp_feed, state_feed, var_feed,
Abs(var_feed.Value-5.4), 5.4*0.5, 5.4*0.3, 5.4*0.05,
'Витрата живлення аварійна!','Витрата живлення вийшла за межі
регламенту!',
'Витрата живлення в нормі');
    indicateLamp (lamp_ctop, state_ctop, var_y1, Abs(var_y1.Value-
97), 1, 0.5, 0.05,
'Концентрація верха аварійна!','Концентрація верха вийшла за
межі регламенту!',
'Концентрація верха в нормі');
    indicateLamp (lamp_cbtm, state_cbtm, var_y2, Abs(var_y2.Value-
0.5), 0.3, 0.5, 0.05,
'Концентрація низа аварійна!','Концентрація низа вийшла за межі
регламенту!',

```

```

'Концентрація низа в нормі');
if (state_reflux.Value = 1) or (state_steam.Value = 1) then
begin
    ind1_main.Visible:=True;
    ind1_main.Frame:=2;
end;
if (state_reflux.Value = 2) or (state_steam.Value = 2) then
begin
    ind1_main.Visible:=True;
    ind1_main.Frame:=1;
end;
if (state_reflux.Value = 0) and (state_steam.Value = 0) then
begin
    ind1_main.Visible:=False;
end;

if (state_feed.Value = 1) then begin
    ind1_feed.Visible:=True;
    ind1_feed.Frame:=2;
end;
if (state_feed.Value = 2) then begin
    ind1_feed.Visible:=True;
    ind1_feed.Frame:=1;
end;
if (state_feed.Value = 0) then begin
    ind1_feed.Visible:=False;
end;
if (state_ctop.Value = 1) or (state_cbtm.Value = 1) then
begin
    ind1_conc.Visible:=True;
    ind1_conc.Frame:=2;
end;
if (state_ctop.Value = 2) or (state_cbtm.Value = 2) then
begin
    ind1_conc.Visible:=True;
    ind1_conc.Frame:=1;
end;
if (state_ctop.Value = 0) and (state_cbtm.Value = 0) then
begin
    ind1_conc.Visible:=False;
end;

end.

```

ВАРІАНТИ ЗАВДАНЬ

Варіанти завдань обираються за номером в журналі викладача. Перелік варіантів ПТП наступний:

1	Випаровування карбаміду
2	Дистиляційна (бражна) колона спиртового виробництва
3	Енергоблок 300 МВт з котлом П-50Р
4	Епюраційна колона спиртового виробництва
5	Жомосушарка цукрового виробництва
6	Колона синтезу технічного спирта
7	Млин з сепаратором для виготовлення цементу
8	Млин МПС-2650
9	Нейтралізатор олії
10	Періодичний реактор синтезу полістирола
11	Піч для спалювання органічних відходів
12	Полчата колона синтезу аміака
13	Рафінаційно-Екстракційна колона в процесі виробництва рафінованого масла
14	Ректифікаційна колона спиртового виробництва
15	Розсиропник меляси цукрового виробництва
16	Установка виготовлення синтетичних миючих засобів
17	Установка підготовки вуглекислого газу для виробництва карбаміду
18	Формпрес масложирового виробництва
19	Хлібопекарна піч ФТЛ-2

Варіант №1. Випаровування карбаміду

<https://www.omicsonline.org/open-access/realtime-control-of-industrial-urea-evaporation-process-using-modelpredictive-control-2157-7048-1000227.pdf>

Базова схема автоматизації установки показана на рис В.1.

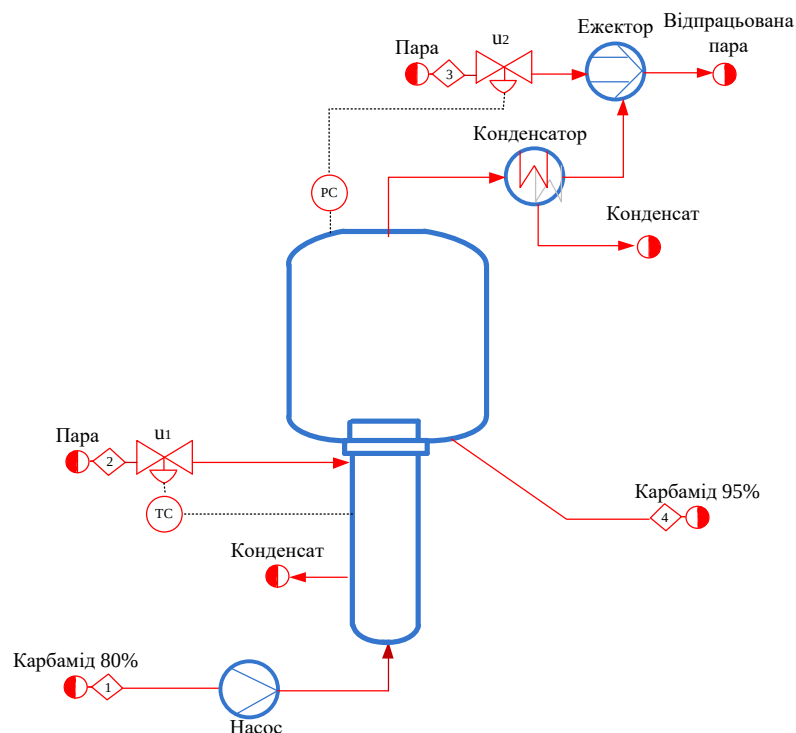


Рисунок В.1 – Базова схема автоматизації системи автоматизації
Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Номінал	Припустиме відхилення
ТС	u_1	Витрата пари в теплообмінник	75 м ³ /год	±30% номінала
	y_1	Температура розчину 80%	125 °C	±10 °C
РС	u_2	Витрата пари в ежектор	16.3 т/год	5..25 т/год
	y_2	Тиск в вакуумному сепараторі	-0.7 бар (розрідження)	± 0.1 бар

Математична модель динаміки в вигляді матриці передатних функцій
(час в секундах).

$$W(s) = \begin{matrix} u_1 & u_2 \\ y_1 & y_2 \end{matrix} \begin{bmatrix} \frac{-18.58s + 9.282}{1548s^2 + 128s + 1} & \frac{23.56s - 3.326}{2398s^2 + 157.7s + 1} \\ \frac{0.1}{30s + 1} e^{-5s} & \frac{-3.061s - 0.06131}{2.711e04s^3 + 2953s^2 + 160.2s + 1} \end{bmatrix}$$

Варіант №2. Дистиляційна (бражна) колона спиртового виробництва
Базова схема автоматизації установки показана на рис В.2.

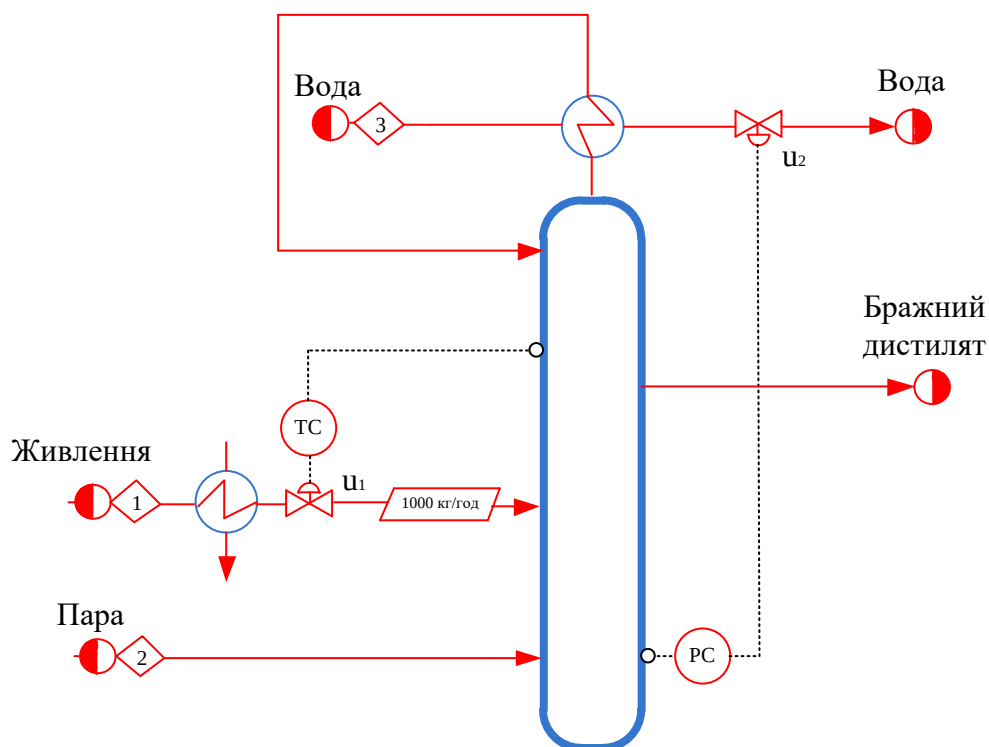


Рисунок В.2 – Базова схема автоматизації системи автоматизації
Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Номінал	Припустиме відхилення
ТС	u_1	Витрата живлення	50%	$\pm 50\%$ номінала
	y_1	Температура к.т.	93.4 °C	± 1 °C
РС	u_2	Витрата води	50%	$\pm 50\%$ номінала
	y_2	Тиск низа	15000 Па	± 5000

Математична модель динаміки в вигляді матриці передатних функцій
(час в секундах).

$$W(s) = \begin{matrix} u_1 & u_2 \\ y_1 & \begin{bmatrix} \frac{0,05}{(100s+1)^2} e^{-160s} & \frac{-0,02}{(110s+1)^2} e^{-290s} \\ \frac{-137,2}{140s+1} & \frac{-16}{(74s+1)^2} e^{-90s} \end{bmatrix} \\ y_2 \end{matrix}$$

Варіант №3. Енергоблок 300 МВт з котлом П-50Р

Базова схема автоматизації установки показана на рис В.3.

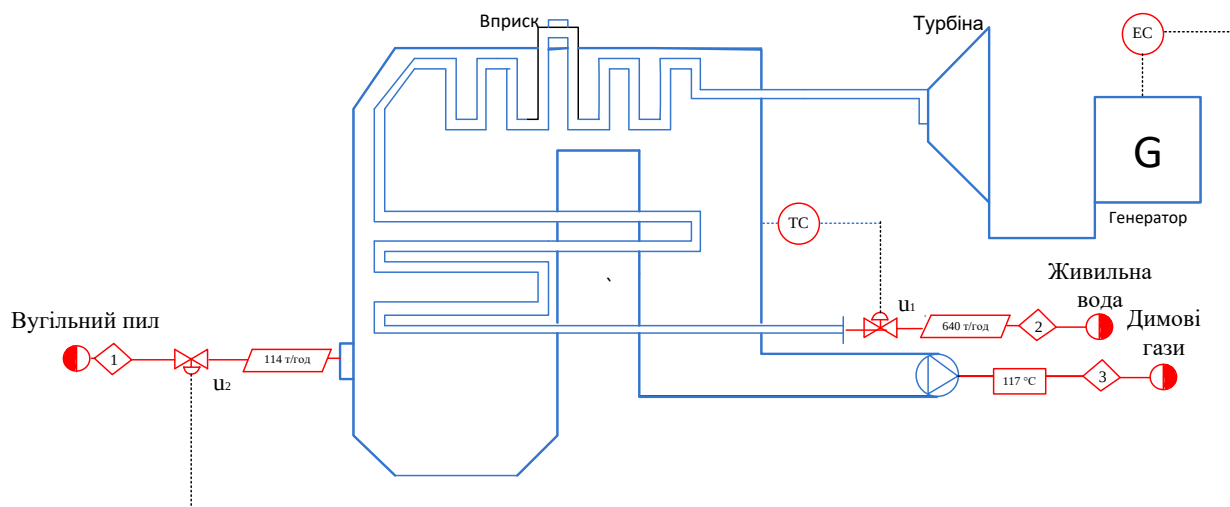


Рисунок В.3 – Базова схема автоматизації системи автоматизації
Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Номінал	Припустиме відхилення
ТС	u_1	Подача живильної води	640 т/год	$\pm 50\%$ номінала
	y_1	Температура ВРЧ	570 °C	± 15 °C
ЕС	u_2	Подача вугільного пилу	114 т/год	$\pm 40\%$ номінала
	y_2	Потужність генератора	300 МВт	± 5 МВт

Математична модель динаміки в вигляді матриці передатних функцій
(час в секундах).

$$W(s) = \begin{matrix} & u_1 & u_2 \\ \begin{matrix} y_1 \\ y_2 \end{matrix} & \begin{bmatrix} \frac{-2.97}{(1+98.5s)^2} & \frac{2.89(1+46.2s)}{(1+37.8s)^3(1+74.2s)} \\ \frac{1.52 \cdot 480s}{(1+240s)^2} & \frac{2.48(1+33.48s)}{(1+35.64s)^4} \end{bmatrix} \end{matrix}$$

Варіант №4. Епюраційна колона спиртового виробництва
Базова схема автоматизації установки показана на рис В.4.

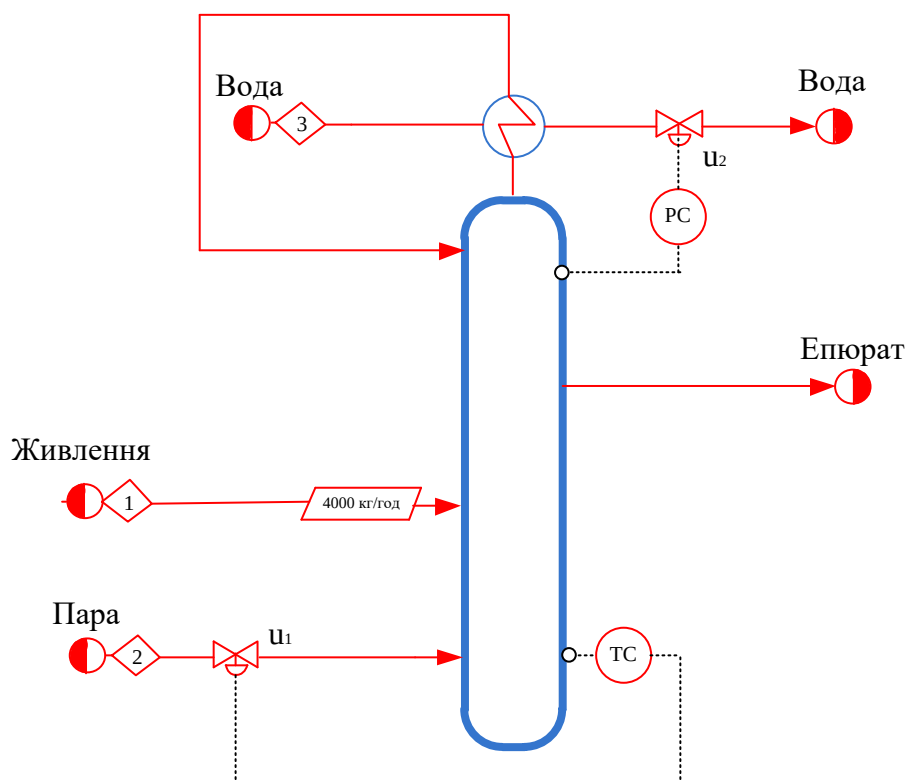


Рисунок В.4 – Базова схема автоматизації системи автоматизації
Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Номінал	Припустиме відхилення
ТС	u_1	Витрата пари	50%	$\pm 50\%$ номінала
	y_1	Температура низа	89 °C	± 1.5 °C
РС	u_2	Витрата води	50%	$\pm 50\%$ номінала
	y_2	Тиск верха	101000 Па	± 10000

Математична модель динаміки в вигляді матриці передатних функцій
(час в секундах).

$$W(s) = \begin{matrix} & u_1 & u_2 \\ \begin{matrix} y_1 \\ y_2 \end{matrix} & \begin{bmatrix} \frac{1,1}{(150s+1)(40s+1)} e^{-7s} & - \\ \frac{0,36}{195s+1} & \frac{156}{(120s+1)^2} e^{-50s} \end{bmatrix} \end{matrix}$$

Варіант №5. :Жомосушарка цукрового виробництва

Базова схема автоматизації установки показана на рис В.5.

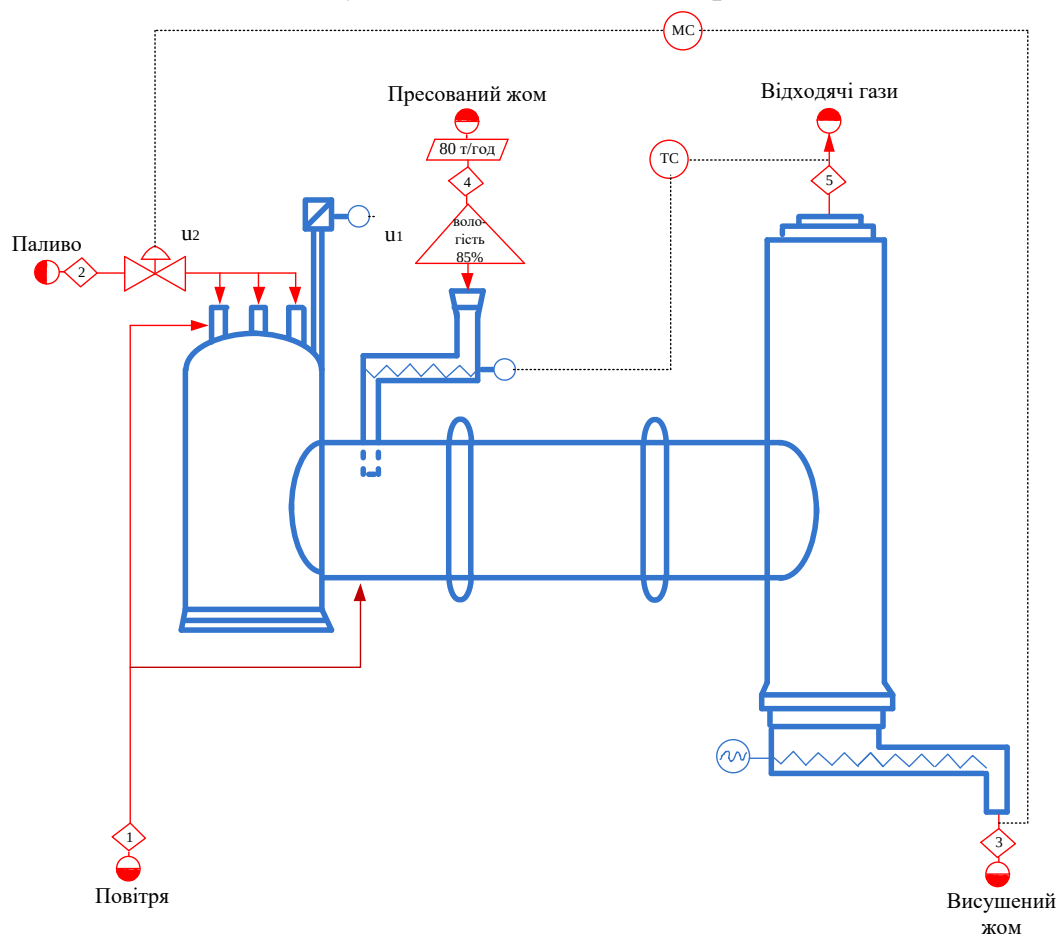


Рисунок В.5 – Базова схема автоматизації системи автоматизації
Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позна-чення	Назва параметру	Номінал	Припустиме відхилення
ТС	u_1	Подача жому	50%	$\pm 50\%$
	y_1	Температура вихідного газу	130 °C	$\pm 30^\circ\text{C}$
МС	u_2	Витрата палива	50%	$\pm 50\%$
	y_2	Вологість висушеного жому	10%	$\pm 5\%$

Математична модель динаміки в вигляді МПФ (час в секундах).

$$W(s) = \begin{matrix} u_1 & u_2 \\ y_1 & y_2 \end{matrix} \begin{bmatrix} \frac{-12}{42p+1}e^{-14p} & \frac{40}{60p+1} \\ \frac{-6}{55p+1}e^{-22p} & \frac{12}{35p+1}e^{-30p} \end{bmatrix}$$

Варіант №6. Колона синтезу технічного спирта
 Базова схема автоматизації установки показана на рис В.6.

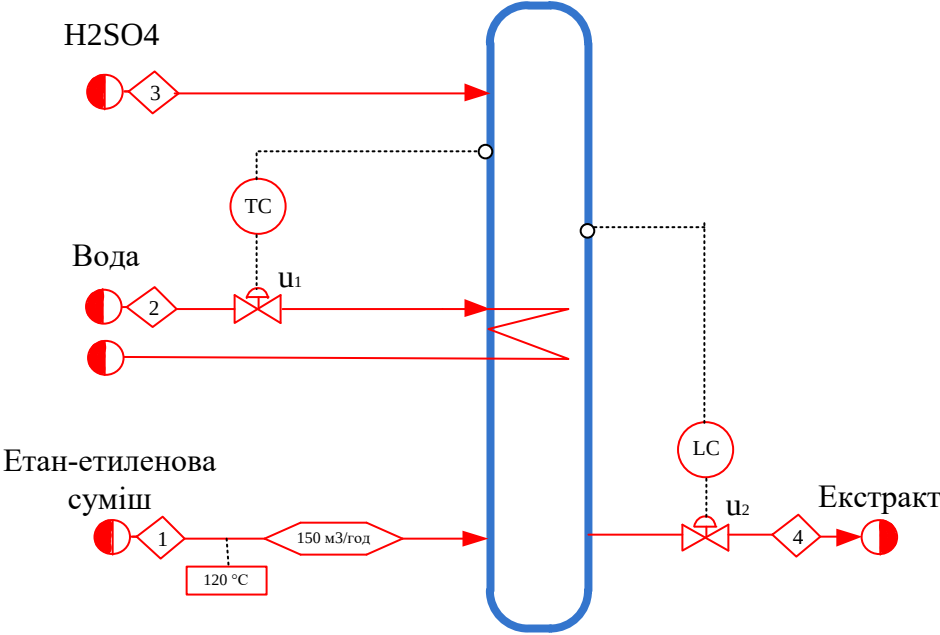


Рисунок В.6 – Базова схема автоматизації системи автоматизації
 Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Номінал	Припустиме відхилення
TC	u_1	Витрата води	50%	$\pm 50\%$ номінала
	y_1	Температура верха	80 °C	± 2 °C
LC	u_2	Витрата екстракта	50%	$\pm 50\%$ номінала
	y_2	Рівень в колоні	70%	$\pm 5\%$

Математична модель динаміки в вигляді матриці передатних функцій
 (час в секундах).

$$W(s) = \begin{matrix} u_1 & u_2 \\ y_1 & y_2 \end{matrix} \begin{bmatrix} \frac{0,13e^{-50s}}{1,79 \cdot 10^6 s^3 + 4,28 \cdot 10^4 s^2 + 342s + 1} & \frac{0,1e^{-480s}}{4,87 \cdot 10^9 s^3 + 9,6 \cdot 10^6 s^2 + 4300s + 1} \\ - & \frac{0,74}{321s + 1} e^{-65s} \end{bmatrix}$$

Варіант №7. Млин з сепаратором для виготовлення цементу
http://www2.imm.dtu.dk/~jbjo/PhD_Thesis_2011_GuruPrasath.pdf
 Базова схема автоматизації установки показана на рис В.7.

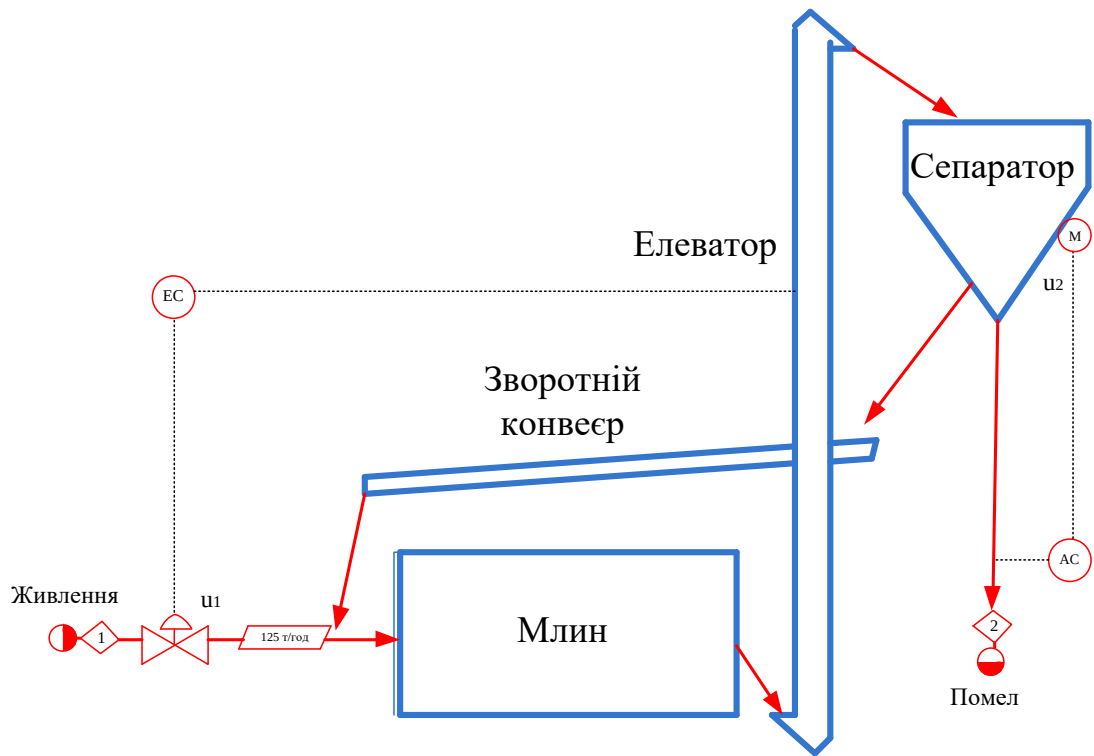


Рисунок В.7 – Базова схема автоматизації системи автоматизації
 Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позна-чення	Назва параметру	Номінал	Припустиме відхилення
ЕС	u_1	Подача цементного клинкеру	125 т/год	$\pm 30\%$ номінала
	y_1	Потужність двигуна елеватора	30КВт	$\pm 30\%$ номінала
АС	u_2	Швидкість сепаратора	70%	$\pm 30\%$
	y_2	Питома поверхня порошка	3100 $\text{см}^2/\text{г}$	$\pm 150 \text{ см}^2/\text{г}$

Математична модель динаміки в вигляді матриці передатних функцій
 (час в секундах).

$$W(s) = \begin{matrix} u_1 & u_2 \\ y_1 & y_2 \end{matrix} \begin{bmatrix} \frac{0.2}{(45s+1)(8s+1)} e^{-5s} & \frac{0.12(8s+1)}{(25s+1)(38s+1)} e^{-1.5s} \\ \frac{-8}{60s+1} e^{-5s} & \frac{2}{(14s+1)(s+1)} e^{-0.1s} \end{bmatrix}$$

Варіант №8. Валковий млин МПС-2650

Базова схема автоматизації установки показана на рис В.8.

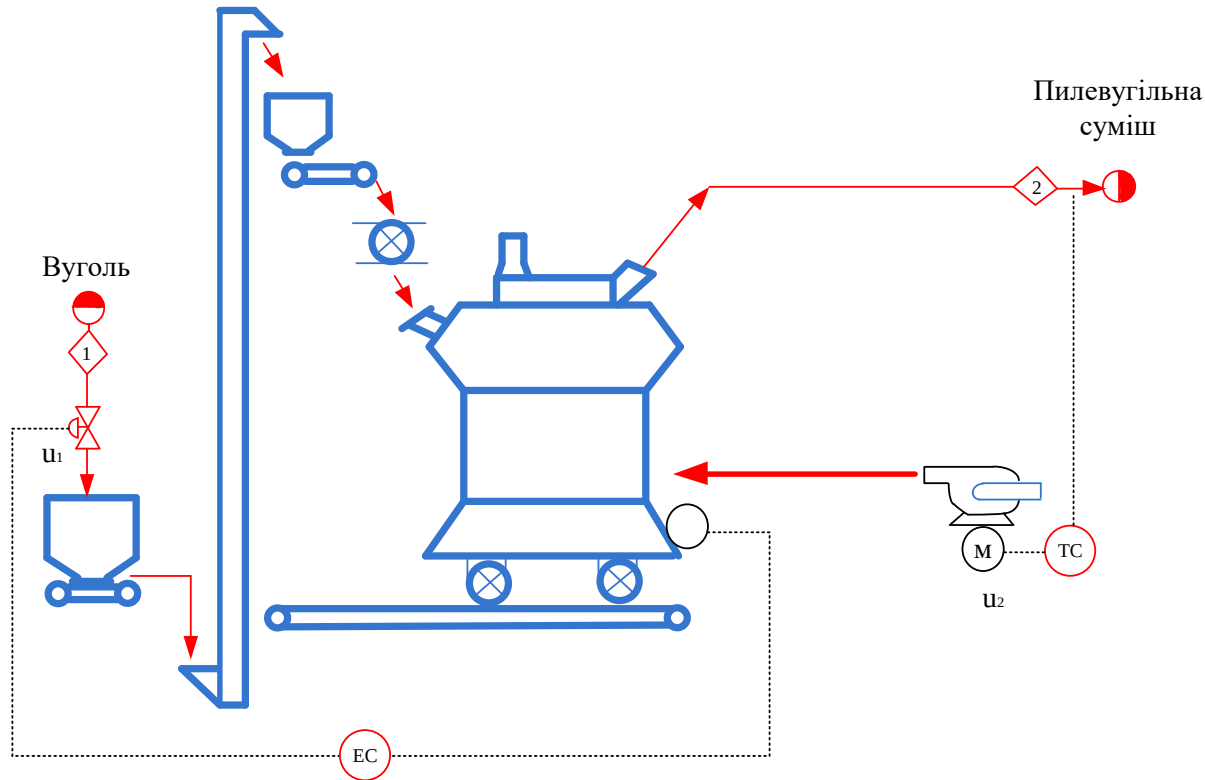


Рисунок В.8 – Базова схема автоматизації системи автоматизації

Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Номінал	Припустиме відхилення
ЕС	u_1	Витрата вугля	80 т/год	$\pm 50\%$ номінала
	y_1	Потужність двигуна млина	615 кВт	± 20 кВт
АС	u_2	Потужність насоса	60%	$\pm 40\%$ номінала
	y_2	Температура суміші	110 °С	± 5 °С

Математична модель динаміки в вигляді матриці передатних функцій (час в секундах).

$$W(s) = \begin{matrix} u_1 & u_2 \\ y_1 & y_2 \end{matrix} \begin{bmatrix} \frac{3,16}{440s+1}e^{-28s} & \frac{3,95}{100s+1} \\ \frac{0,56}{410s+1}e^{-60s} & \frac{0,88}{350p+1}e^{-50s} \end{bmatrix}$$

Варіант №9. Нейтралізатор олії

Базова схема автоматизації установки показана на рис В.9.

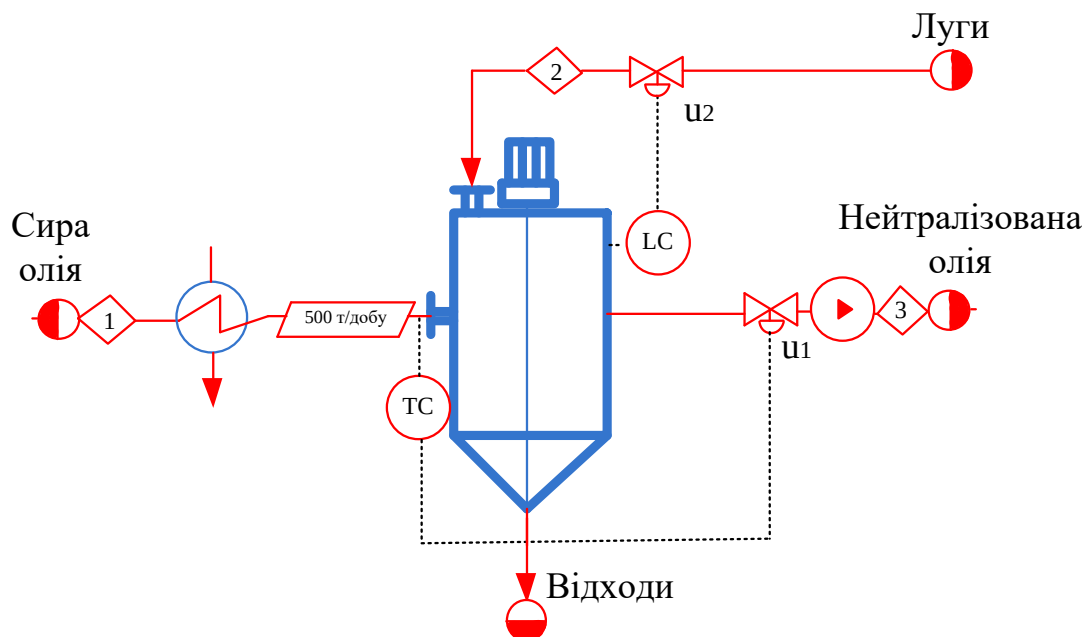


Рисунок В.9 – Базова схема автоматизації системи автоматизації

Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Номінал	Припустиме відхилення
TC	u_1	Відбір олії	50%	$\pm 50\%$ номінала
	y_1	Температура олії	72	± 2.5 °C
LC	u_2	Витрата лугів	50%	$\pm 50\%$ номінала
	y_2	Рівень в нейтралізаторі	70%	$\pm 30\%$

Математична модель динаміки в вигляді матриці передатних функцій (час в секундах).

$$W(s) = \begin{matrix} & u_1 & u_2 \\ \begin{matrix} y_1 \\ y_2 \end{matrix} & \begin{bmatrix} \frac{0,35}{87s+1} e^{-31s} & \frac{-0,5}{95s+1} e^{-25s} \\ \frac{-0,97}{540s+1} e^{-20s} & \frac{1}{12100p} e^{-8s} \end{bmatrix} \end{matrix}$$

Варіант №10. Періодичний реактор синтезу полістрирола
Базова схема автоматизації установки показана на рис В.10.

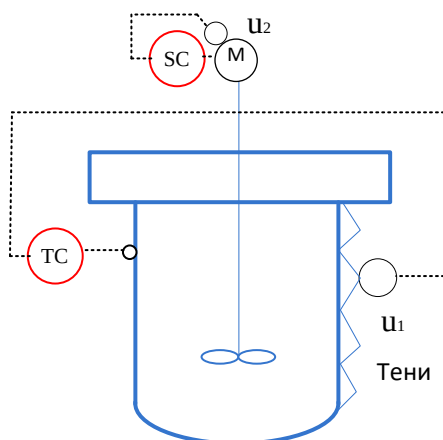


Рисунок В.10 – Базова схема автоматизації системи автоматизації
Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Номінал	Припустиме відхилення
TC	u_1	Напруга в блоці керування ТЕНами	67В	0..380 В
	y_1	Температура змішування	90°C	± 15 °C
SC	u_2	Частота обертів двигуна мішалки	840 об/хв	± 200 об/хв
	y_2	Число обертів двигуна мішалки	840 об/хв	± 200 об/хв

Математична модель динаміки в вигляді матриці передатних функцій
(час в секундах).

$$W(s) = \begin{matrix} u_1 & u_2 \\ y_1 & y_2 \end{matrix} \begin{bmatrix} W_{11} & W_{21} \\ W_{12} & W_{22} \end{bmatrix}$$

$$W_{11} = \frac{1.342 \cdot 10^{12} \cdot s^4 - 6.786 \cdot 10^9 \cdot s^3 + 3.24 \cdot 10^7 \cdot s^2 - 6.012 \cdot 10^4 \cdot s + 75.28}{2.286 \cdot 10^{13} \cdot s^5 + 9.821 \cdot 10^{10} \cdot s^4 + 4.595 \cdot 10^8 \cdot s^3 + 1.005 \cdot 10^6 \cdot s^2 + 1506 \cdot s + 1}$$

$$W_{12} = \frac{-1.53 \times 10^5 \times s^4 - 1.219 \times 10^4 \times s^3 + 129.9 \times s^2 + 9.052 \times s + 0.1}{6.376 \times 10^7 \times s^5 + 9.01 \times 10^6 \times s^4 + 4.605 \times 10^5 \times s^3 + 1.44 \times 10^4 \times s^2 + 199.2 \times s + 1}$$

$$W_{21} = \frac{-1392 \cdot s^4 - 475.4 \cdot 10^4 \cdot s^3 - 58.19 \cdot s^2 - 2.915 \cdot s - 0.05}{2.303 \cdot 10^5 \cdot s^5 + 9.941 \cdot 10^4 \cdot s^4 + 1.688 \cdot 10^4 \cdot s^3 + 1416 \cdot s^2 + 59.19 \cdot s + 1}$$

$$W_{22} = \frac{2517 \cdot s^4 - 1408 \cdot s^3 - 297.82 \cdot s^2 - 28.14 \cdot s + 1}{1.791 \cdot 10^4 \cdot s^5 + 1.253 \cdot 10^4 \cdot s^4 + 3524 \cdot s^3 + 497.3 \cdot s^2 + 35.22 \cdot s + 1}$$

Варіант №11. Піч для спалювання органічних відходів

Базова схема автоматизації установки показана на рис В.11.

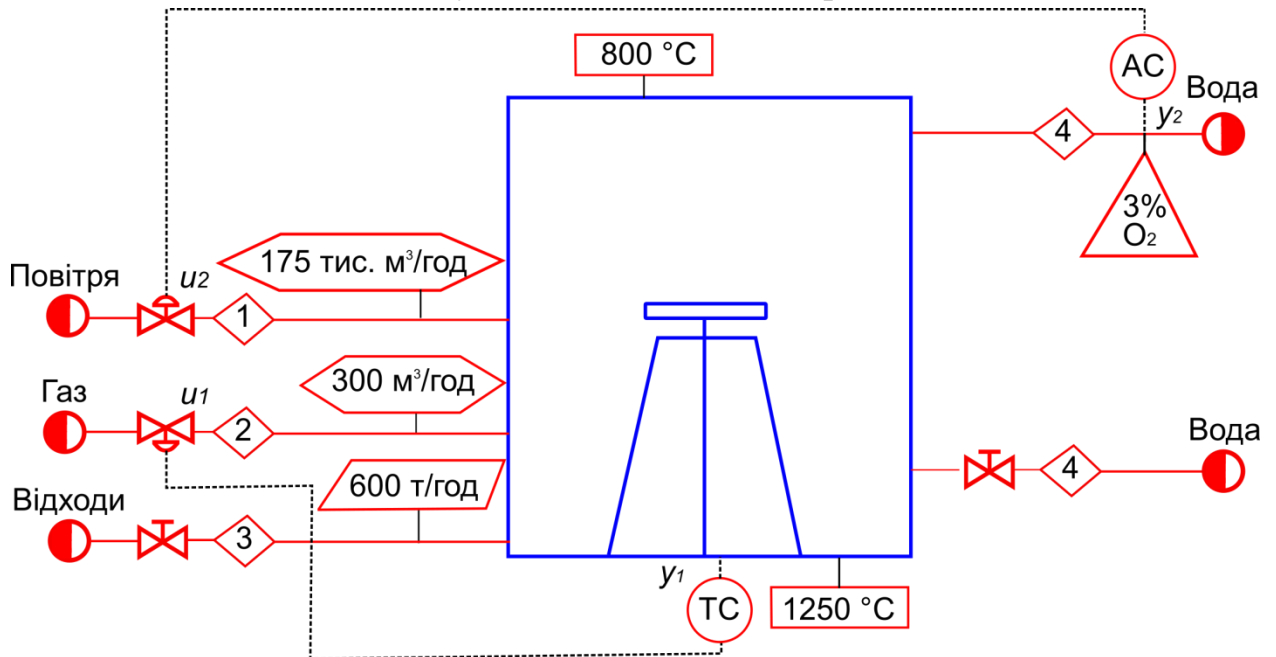


Рисунок В.11 – Базова схема автоматизації системи автоматизації

Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Припустиме відхилення
ТС	u_1	Витрата газу	$\pm 50\%$ номінала
	y_1	Температура в I зоні печі	$\pm 100\text{ }^{\circ}\text{C}$
АС	u_2	Витрата повітря	$\pm 50\%$ номінала
	y_2	Концентрація O_2 в воді	$\pm 1\%$

Математична модель динаміки в вигляді матриці передатних функцій (час в секундах).

$$W(s) = \begin{matrix} u_1 & u_2 \\ y_1 & y_2 \end{matrix} \begin{bmatrix} \frac{0,6}{360s+1} & \frac{-0,005}{360s+1} \\ \frac{-0,017}{135s+1} e^{-100s} & \frac{0,0007}{80s+1} e^{-100s} \end{bmatrix}$$

Варіант №12. Полчата колона синтезу аміака

Базова схема автоматизації установки показана на рис В.12.

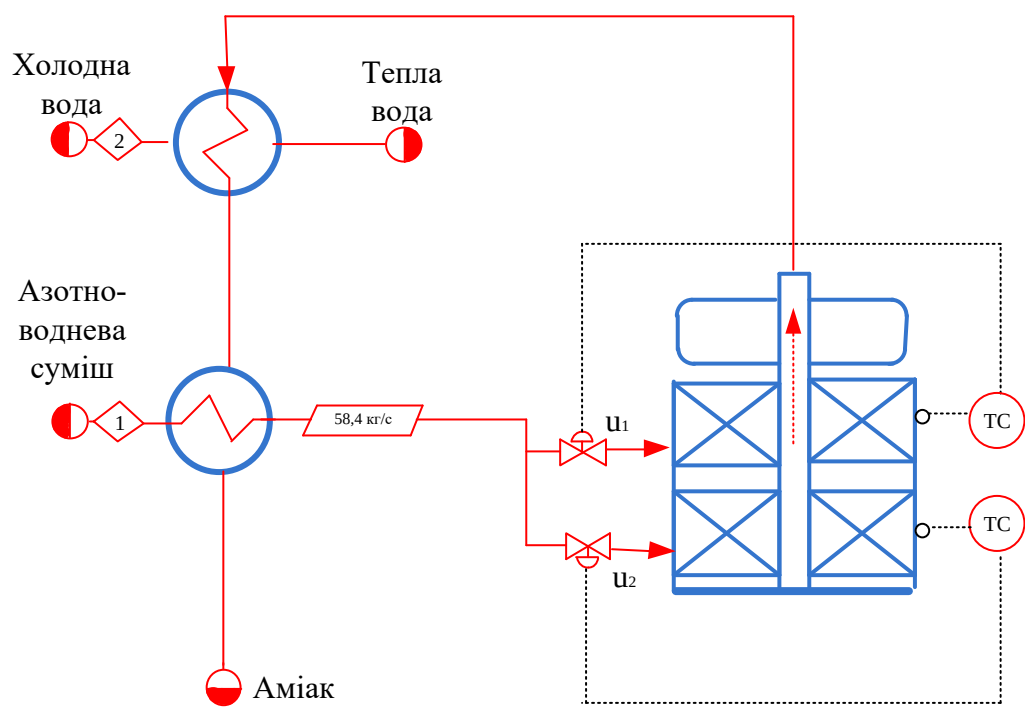


Рисунок В.12 – Базова схема автоматизації системи автоматизації

Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Номінал	Припустиме відхилення
ТС	u_1	Подача на 1 полку	50%	$\pm 50\%$ номінала
	y_1	Температура на 1 полці	400 °C	± 15 °C
ТС	u_2	Подача на 2 полку	60%	$\pm 40\%$ номінала
	y_2	Температура на 2 полці	400 °C	± 15 °C

Математична модель динаміки в вигляді матриці передатних функцій (час в секундах).

$$W(s) = \begin{matrix} & u_1 & u_2 \\ \begin{matrix} y_1 \\ y_2 \end{matrix} & \begin{bmatrix} \frac{-3.4}{102s+1}e^{-100s} & - \\ \frac{-1.6}{120s+1}e^{-160s} & \frac{-1.6}{120s+1}e^{-160s} \end{bmatrix} \end{matrix}$$

Варіант №13. Рафінаційно-Екстракційна колона в процесі виробництва рафінованого масла

Базова схема автоматизації установки показана на рис В.13.

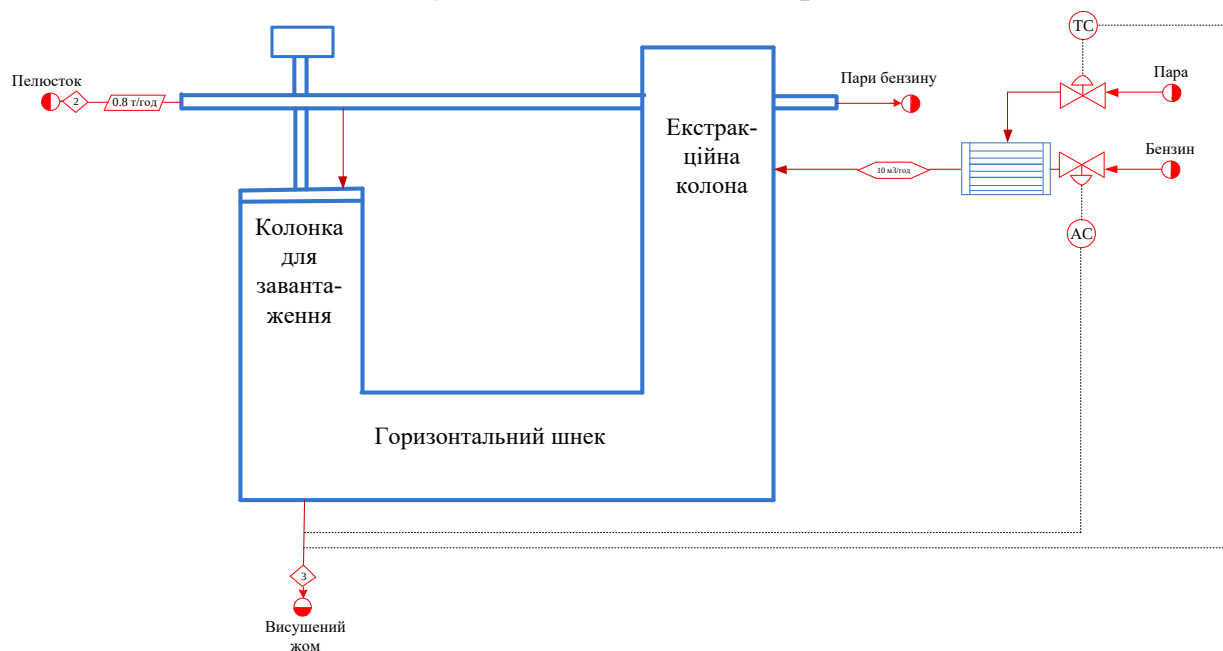


Рисунок В.13 – Базова схема автоматизації системи автоматизації
Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Номінал	Припустиме відхилення
ТС	u_1	Витрата пари	0.2 м³/год	±50% номінала
	y_1	Температура вихідного бензину	64 °C	±3°C
МС	u_2	Витрата бензину	50%	±50%
	y_2	Масличність шроту	1.7%	±0.2%

Математична модель динаміки в вигляді матриці передатних функцій (час в секундах).

$$W(s) = \begin{matrix} u_1 & u_2 \\ y_1 & \begin{bmatrix} \frac{0.106}{4.2p+1}e^{-8p} & \frac{-0.7}{3p+1}e^{-32p} \\ \frac{-0.7}{17p+1}e^{-21p} & \frac{-0.33}{22p+1}e^{-33p} \end{bmatrix} \\ y_2 \end{matrix}$$

Варіант №14. Ректифікаційна колона спиртового виробництва
Базова схема автоматизації установки показана на рис В.14.

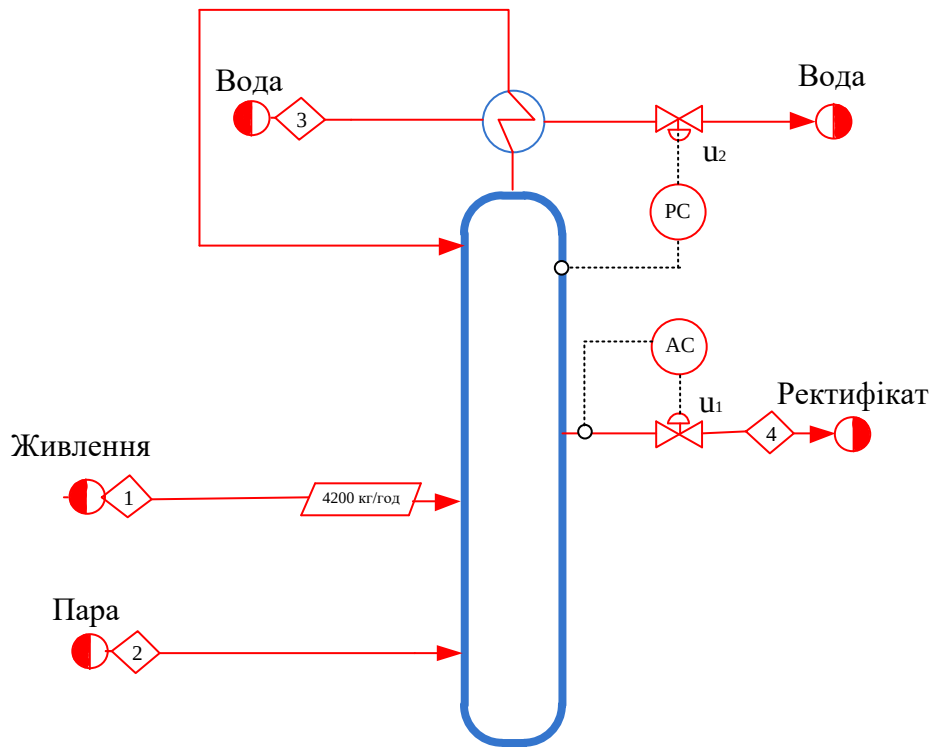


Рисунок В.5 – Базова схема автоматизації системи автоматизації
Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Номінал	Припустиме відхилення
АС	u_1	Витрата спирта	50%	$\pm 50\%$ номінала
	y_1	Концентрація спирта	96.3%	$\pm 0.1 \%$
РС	u_2	Витрата води	50%	$\pm 50\%$ номінала
	y_2	Тиск верха	125000 Па	± 10000

Математична модель динаміки в вигляді матриці передатних функцій
(час в секундах).

$$W(s) = \begin{matrix} u_1 & u_2 \\ y_1 & y_2 \end{matrix} \begin{bmatrix} \frac{1,7}{(70s+1)^2} e^{-430s} & \frac{1,4}{(27s+1)^2} e^{-640s} \\ \frac{3}{(90s+1)^2} e^{-190s} & \frac{6}{(47s+1)^2} \end{bmatrix}$$

Варіант №15. Розсиропник меляси цукрового виробництва
Базова схема автоматизації установки показана на рис В.15.

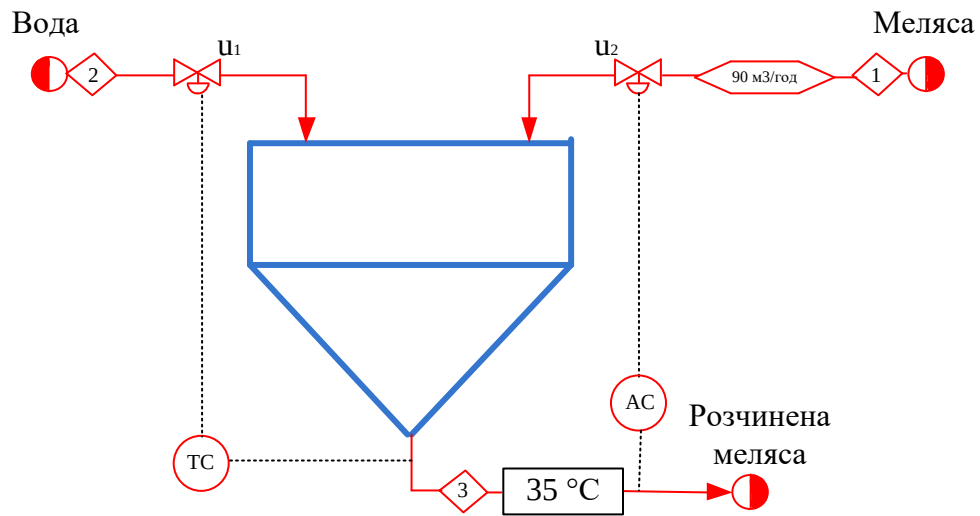


Рисунок В.15 – Базова схема автоматизації системи автоматизації
Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Номінал	Припустиме відхилення
ТС	u_1	Витрата води	50%	$\pm 50\%$ номінала
	y_1	Температура продукта	35 °C	± 2 °C
АС	u_2	Витрата меляси	50%	$\pm 50\%$ номінала
	y_2	Густина меляси	1400 кг/м3	± 50 кг/м3

Математична модель динаміки в вигляді матриці передатних функцій
(час в секундах).

$$W(s) = \begin{matrix} u_1 & u_2 \\ y_1 & \begin{bmatrix} \frac{0,6}{200s+1}e^{-12s} & \frac{-0,4}{250s+1}e^{-155s} \\ \frac{-0,1}{100s+1}e^{-500s} & \frac{0,3}{50s+1}e^{-400s} \end{bmatrix} \\ y_2 \end{matrix}$$

Варіант №16. Установа виготовлення синтетичних миючих засобів

Базова схема автоматизації установки показана на рис В.16.

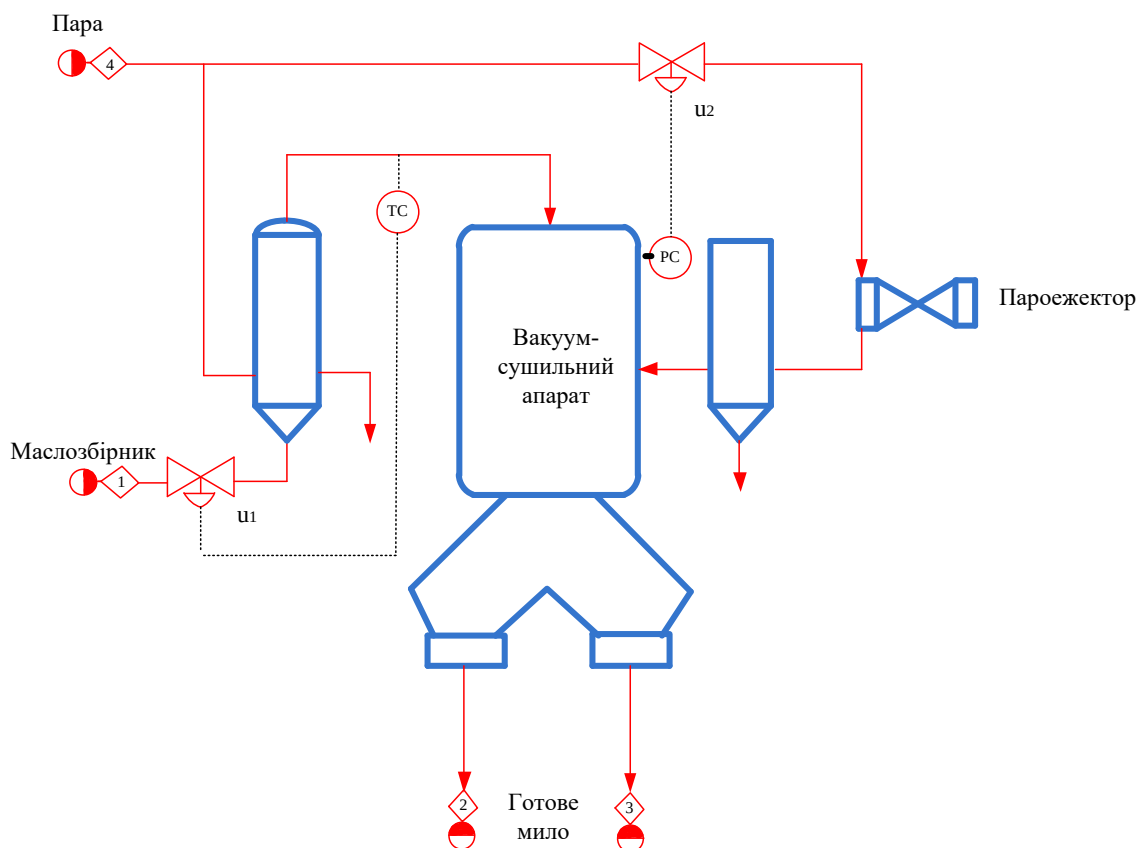


Рисунок В.16 – Базова схема автоматизації системи автоматизації

Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Номінал	Припустиме відхилення
ТС	u_1	Витрата мильної основи	35 т/год	$\pm 30\%$ номінала
	y_1	Температура мильної основи	100 °C	± 10 °C
РС	u_2	Витрата пари в паросежектор	0.5 т/год	$\pm 0.08\%$ номінала
	y_2	Розрідження в вакуумі	50 мм рт.ст.	± 35 мм.рт.ст

Математична модель динаміки в вигляді МПФ (час в секундах).

$$W(s) = \begin{matrix} u_1 & u_2 \\ y_1 & y_2 \end{matrix} \begin{bmatrix} \frac{3.8}{5s+1} e^{-2.5s} & - \\ \frac{6.5}{2.3s+1} e^{-0.6s} & \frac{-125}{1.1s+1} e^{-0.25s} \end{bmatrix}$$

Варіант №17. Установа підготовки вуглекислого газу для виробництва карбаміду

Базова схема автоматизації установки показана на рис В.17.

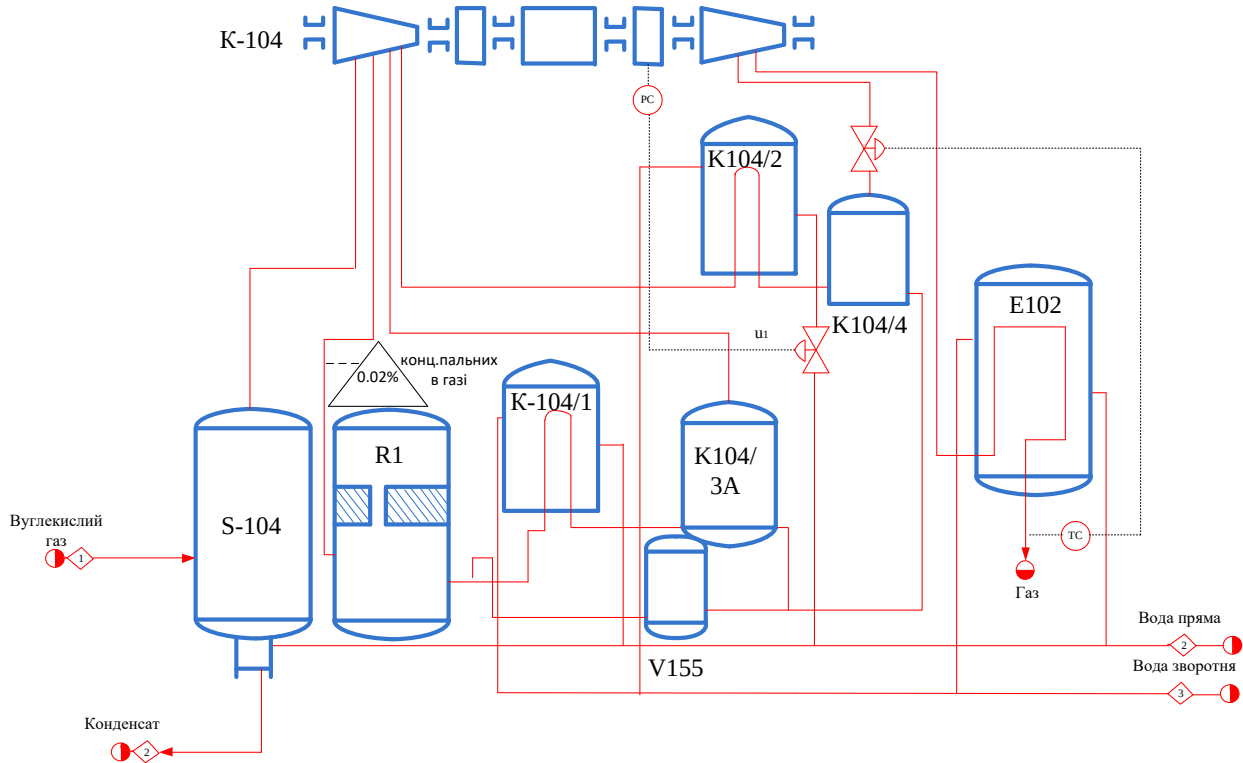


Рисунок В.17 – Базова схема автоматизації системи автоматизації

Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позна-чення	Назва параметру	Номінал	Припустиме відхилення
РС	u_1	Витрата води на охолодження холодильника K104/2	134 кг/с	$\pm 50\%$ номінала
	y_1	Тиск в третій ступені компресора K104	2.6 МПа	± 0.5 МПа
ЕС	u_2	витрата газу на шаблі компресора K104	119 кг/с	$\pm 50\%$ номінала
	y_2	Температура після холодильника E102	50 °C	± 20 °C

Математична модель динаміки в вигляді МПФ (час в секундах).

$$W(s) = \begin{matrix} u_1 & u_2 \\ y_1 & y_2 \end{matrix} \begin{bmatrix} \frac{-9.1}{100s+1} & \frac{0.086}{57.5s+1} \\ \frac{-3.79e-5}{86.55s+1} & \frac{0.1}{17.8s+1} \end{bmatrix}$$

Варіант №18. Формпрес масложирового виробництва
Базова схема автоматизації установки показана на рис В.18.

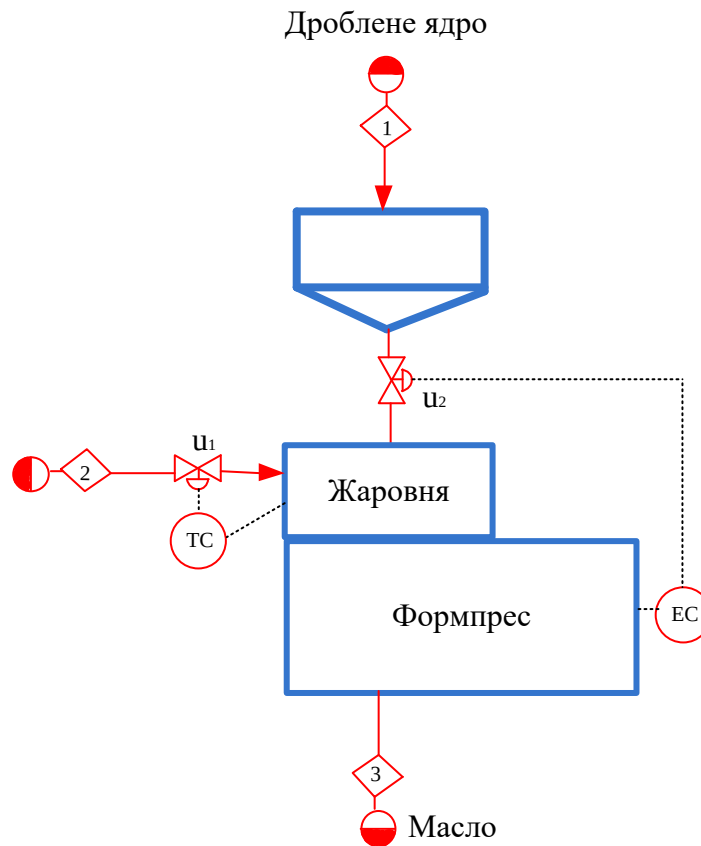


Рисунок В.18 – Базова схема автоматизації системи автоматизації
Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Номінал	Припустиме відхилення
ТС	u_1	Витрата м'ятки	50%	$\pm 50\%$ номінала
	y_1	Температура в жаровні	120 °C	± 20 °C
АС	u_2	Витрата меляси	50%	$\pm 50\%$ номінала
	y_2	Ток	80 А	± 20 А

Математична модель динаміки в вигляді матриці передатних функцій
(час в секундах).

$$W(s) = \begin{matrix} u_1 & u_2 \\ y_1 & y_2 \end{matrix} \begin{bmatrix} \frac{0,6}{53,8s+1} e^{-87,6s} & \frac{-0,8}{9,92s+1} e^{-26,5s} \\ \frac{-0,5}{13s+1} e^{-28s} & \frac{1,6}{101,4s+1} e^{-15s} \end{bmatrix}$$

Варіант №19. Хлібопекарна піч ФТЛ-2

Базова схема автоматизації установки показана на рис В.19.

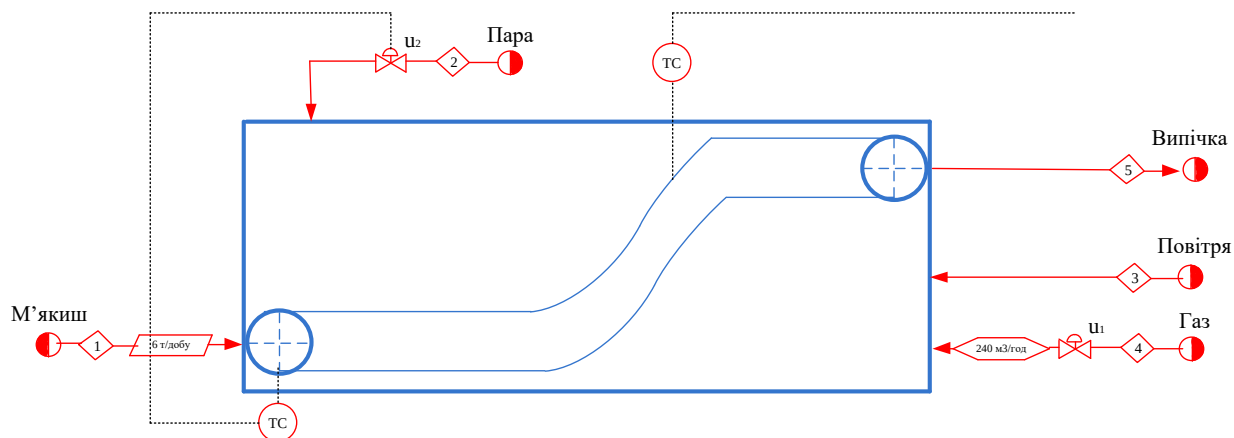


Рисунок В.19 – Базова схема автоматизації системи автоматизації
Основні вимоги до системи керування згідно технологічних вимог

Регулятор	Позначення	Назва параметру	Номінал	Припустиме відхилення
ТС	u_1	Витрата газу	50%	$\pm 50\%$ номінала
	y_1	Температура в з. 1	513 °C	± 5 °C
ТС	u_2	Витрата пара	50%	$\pm 50\%$ номінала
	y_2	Температура в з.2	265 °C	$\pm 3\%$

Математична модель динаміки в вигляді матриці передатних функцій
(час в секундах).

$$W(s) = \begin{matrix} & u_1 & u_2 \\ \begin{matrix} y_1 \\ y_2 \end{matrix} & \begin{bmatrix} \frac{1,05}{12,2s+1} e^{-1,5s} & - \\ \frac{0,17}{17s+1} e^{-4,3s} & \frac{-0,061}{15s+1} e^{-9s} \end{bmatrix} \end{matrix}$$