

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
ІНСТИТУТ ШТУЧНОГО ІНТЕЛЕКТУ ТА РОБОТОТЕХНІКИ
КАФЕДРА ПРОГРАМНИХ І КОМП'ЮТЕРНО-ІНТЕГРОВАНІХ ТЕХНОЛОГІЙ

Методичні вказівки з дисципліни
ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ
(Лабораторний практикум. Частина друга)
Для студентів інституту штучного інтелекту та робототехніки

Перший (бакалаврський) рівень вищої освіти

Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології

Освітньо-професійна програма: Комп'ютерні технології автоматизації;

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
ІНСТИТУТ ШТУЧНОГО ІНТЕЛЕКТУ ТА РОБОТОТЕХНІКИ
КАФЕДРА ПРОГРАМНИХ І КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ

Методичні вказівки з дисципліни
ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ
(Лабораторний практикум. Частина друга)
Для студентів інституту штучного інтелекту та робототехніки

Перший (бакалаврський) рівень вищої освіти

Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології

Освітньо-професійна програма: Комп'ютерні технології автоматизації;

Схвалено на засіданні кафедри ПКІТ протокол
№7 від 26.01.2022 р.

Методичні вказівки з дисципліни «Обчислювальна техніка та комп'ютерні технології» (Лабораторний практикум. Частина друга) для студ. спец. 151 – «Автоматизація та комп'ютерно-інтегровані технології» ден. та заоч. форм навч./уклад.: В.О. Давидов - Одеса: ОП, 2022. - 13 с.

Укладачі: **В.О. Давидів**, Канд. техн. наук

ЗМІСТ

ЗМІСТ.....	4
Загальні відомості	5
Лабораторна робота №1	8
Інформаційна модель.....	8
Модель станів	8
Алгоритми дій	8
Лабораторна робота 2	10
Програмна реалізація.....	10
Тестування	13
Лабораторна робота 3	14

ЗАГАЛЬНІ ВІДОМОСТІ

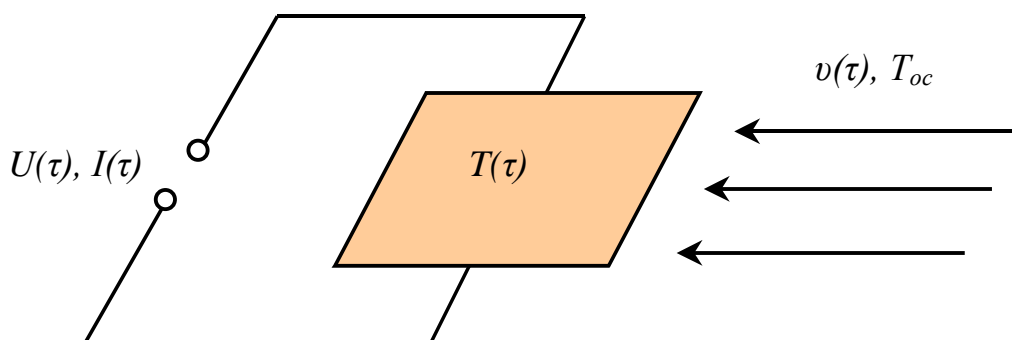
Основні завдання даного циклу лабораторних робіт:

- Створити програмну модель навчального стенду;
- удосконалити у процесі роботи над моделлю стенду свої навички програмування;
- Набути за допомогою моделі стенду навички дослідницької роботи.

Ціль робіт: створення моделі автоматизованої системи управління температурою металеві пластинки (АСУ ТМП).

2. Характеристика об'єкта автоматизації

Об'єктом управління (ОУ) є мідна пластинка, що нагрівається з одного боку напругою або струмом і охолоджується з іншого боку потоком повітря (рис. 1).



Мал. 1. Об'єкт управління.

Величини струму $I(\tau)$ і напруги $U(\tau)$ визначають кількість енергії, що підводиться до пластинки. Швидкість потоку повітря і температура навколишнього середовища T_{oc} визначають величину втрат.

Мідна пластинка характеризується такими атрибутами як: довжина (a), ширина (b), висота (h), густина (ρ), теплоємність (Cm), коефіцієнт теплопередачі (α).

Завдання управління полягає у підтримці заданої температури пластинки $T(\tau) = T_{зад}(\tau)$ шляхом зміни напруги, або струму, або швидкості потоку повітря. Структурна схема системи керування представлена на рис. 2.

3. Опис вимог до системи

3.1. Модель АСУ ТМП має являти собою один програмний модуль (exe-файл).

3.2. У ПЗ моделі АСУ ТМП мають бути реалізовані модель технологічного процесу зображеного на рис. 1 та модель АСУ представлена на рис. 2.

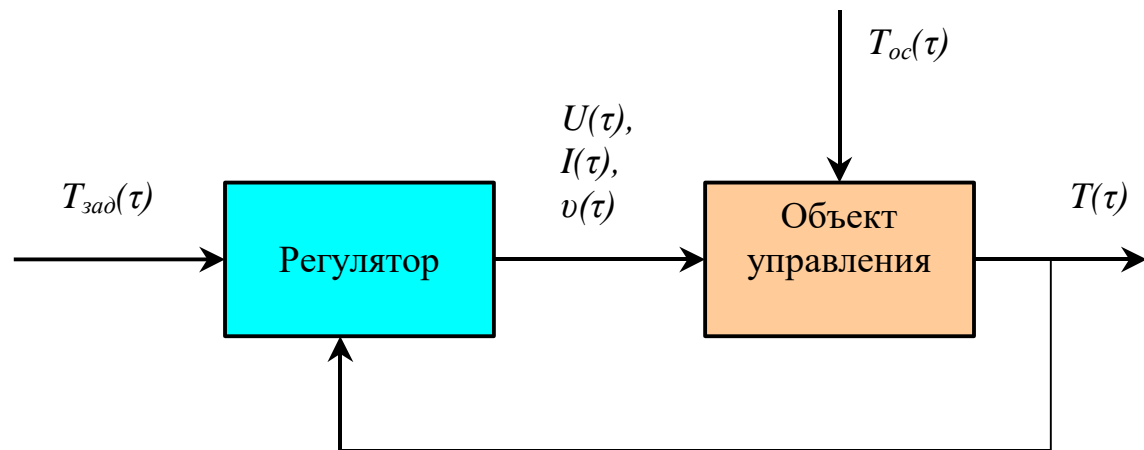
3.3. Модель АСУ ТМП повинна мати інтерфейс користувача.

3.4. Приклад інтерфейсу моделі АСУ ТМП представлений на рис. 3.

Інтерфейс повинен дозволяти оператору:

- задавати значення основних атрибутів ОУ (геометричні розміри та фізичні властивості);
- задавати поточне значення температури навколишнього середовища T_{oc} ;
- перемикає автоматичний та ручний режими керування;
- Змінювати в ручному режимі управління значення величин $U(\tau)$, $I(\tau)$, $v(\tau)$, з кроком не більше 0.01 від максимального значення;
- задавати закон управління (П, І, Д) для автоматичного режиму (за погодженням з викладачем, закон управління може бути один);
- Індикувати в режимі реального часу поточну температуру платівки;
- Задавати значення заданої температури $T_{зад}(\tau)$ з точністю ± 0.2 °С.

Дозволяється використовувати цифрову індикацію поточної температури замість графічного відображення.



Мал. 2. Структурна схема системи керування.

3.5. Модель АСУ ТМП має функціонувати як реального часу.

3.6. Повинен вестись протокол роботи ОУ та СУ.

3.7. Документація на модель АСУ ТМП має включати:

- Специфікацію вимог;
- Об'єктно-орієнтовану модель АСУ ТМП;
- методику тестування АСУ ТМП

4. Обмеження допустимих витрат на розробку

4.1. Прийняти коефіцієнт теплопередачі $\alpha = 50 + 30 \cdot v_2 + 30 \cdot v_3$.

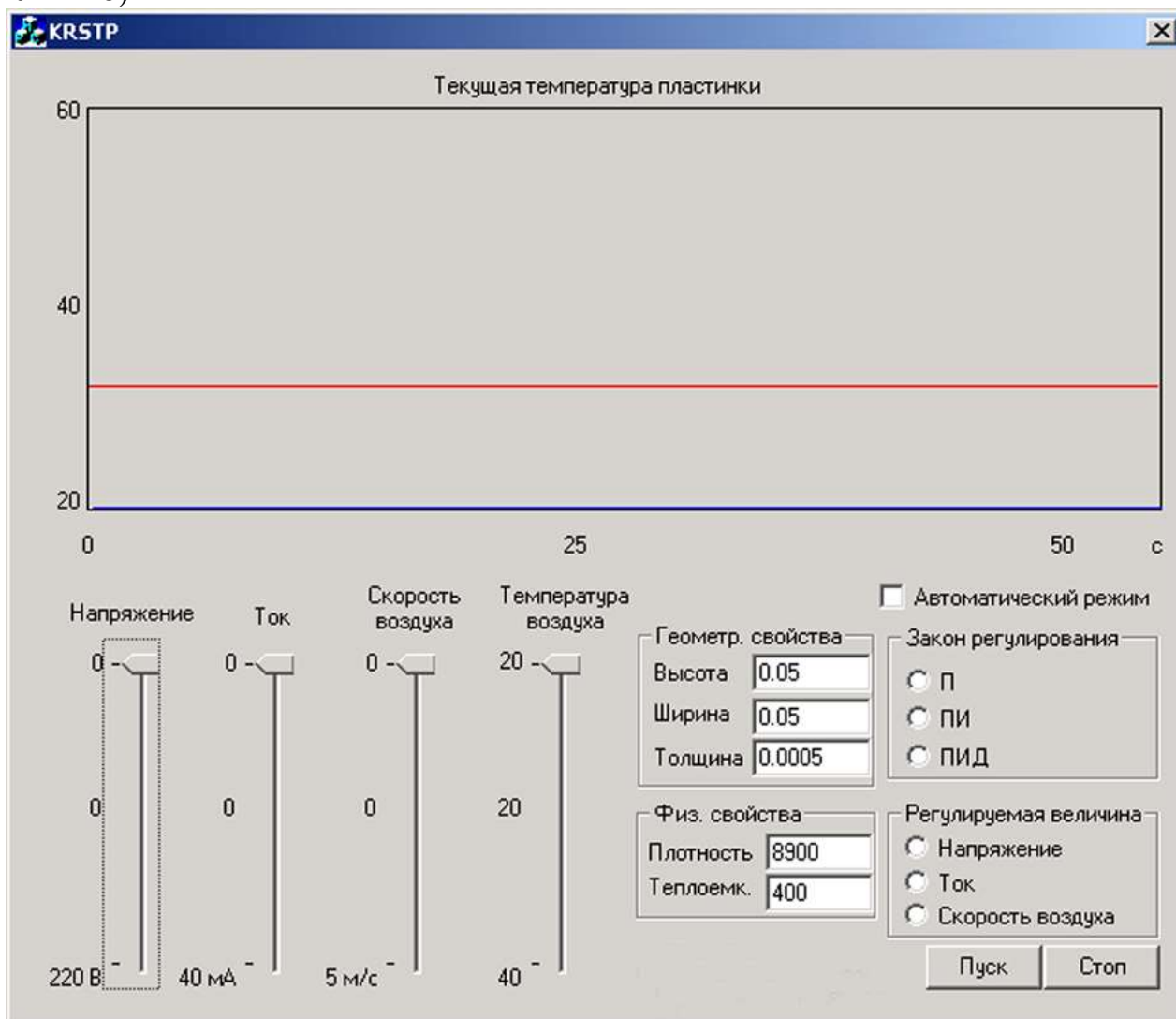
4.2. Прийняти налаштування регулятора:

П-складова = 0.1 ÷ 0.3 від максимуму регульованої величини,

І-складова = 2 * П-складова,

Д-складова = П-складова ÷ І-складова.

4.3. На виконання лабораторних робіт студентам приділяється 9 тижнів (з 10 по 18).



Мал. 3. Приклад інтерфейсу моделі АСУ ТМП.

Отже, технологічний процес, який підлягає програмній реалізації, нам відомий. Наразі можна перейти до проектування моделі стенду.

ЛАБОРАТОРНА РОБОТА №1

Мета роботи – удосконалити навички проектування об'єктно-орієнтованого ПЗ.

Завдання:

- Розробити об'єктно-орієнтовану модель стенду;

Інформаційна модель

В основу інформаційної моделі АСУ ТМП рекомендується покласти модель навчального стенду для вирішення завдання управління з відхилення, розроблену у лабораторній роботі №3 3-го семестру з дисципліни КТіВР.

З попереднього досвіду, атрибутами ОУ будуть: температура пластинки; напруга, струм, швидкість потоку повітря; Температура навколишнього середовища; висота, ширина, довжина платівки; щільність, теплоємність пластинки; час, що минув від початку моделювання.

Відповідно для СУ атрибутами будуть: режим роботи (ручний/автоматичний); коефіцієнти П, І, Д; зона нечутливості, зона перегулювання; задану температуру.

Модель станів

Модель станів ОУ та СУ, практично без змін, можна взяти з лабораторної роботи №3 3-го семестру з дисципліни КТіВР. Покладені в її основу принципи дозволяють використовувати цю модель для кола завдань.

Алгоритми дій

Як і раніше алгоритми потребують дії в станах **StCalc** для ОУ та СУ.

В основу алгоритму ОУ покладемо баланс енергій.

За рахунок використання електричної енергії у платівці виділяється деяка кількість тепла $Q_{\text{эл}}$, що визначається виразом:

$$Q_{\text{эл}} = UI\tau[\text{Дж}] \quad (1)$$

Ця кількість тепла витрачається на нагрівання самої платівки:

$$Q_{\text{нагр}} = mC_p(T_{i+1} - T_i)[\text{Дж}](2)$$

де m - Маса платівки,

C_p - Теплоємність пластинки,

T_{i+1}, T_i - відповідно нова та поточна температури платівки.

А також на втрати в довкілля:

$$Q_{oc} = \alpha F(T_i - T_{oc})\tau [\text{Дж}] (3)$$

де $\alpha = 50 + 50v^2 + 50v^3$ - Коефіцієнт теплопередачі від платівки до повітря,

F - Площа пластинки,

T_{oc} - Температура навколишнього середовища.

Із закону збереження енергії випливає, що:

$$Q_{эл} = Q_{нагр} + Q_{oc} \quad (4)$$

або

$$U\tau = mC_p(T_{i+1} - T_i) + \alpha F(T_i - T_{oc})\tau \quad (5)$$

З виразу (5) легко вивести залежність для нової температури платівки T_{i+1} від часу τ .

Для СУ нам потрібний алгоритм визначення нового значення регульованої величини. У випадку для ПД закону:

$$\mu = \mu \pm (K_p \varepsilon + K_i \int \varepsilon d\varepsilon + K_d \frac{d\varepsilon}{d\tau}) \quad (6)$$

або ж:

$$\mu_i = \mu_{i-1} \pm (K_p \varepsilon_i + K_i \sum_{j=i-9}^i \varepsilon_j \tau + K_d \frac{\varepsilon_i - \varepsilon_{i-1}}{\tau}) \quad (7)$$

У протоколі лабораторної роботи навести: інформаційну модель, моделі станів та таблиці станів для ОУ та СУ, алгоритми дій.

ЛАБОРАТОРНА РОБОТА 2

Ціль роботи: удосконалити навички розробки об'єктно-орієнтованого ПЗ.

Завдання:

- Розробити програмний код моделі навчального стенду;
- Розробити методику тестування;
- Провести тестування.

Програмна реалізація

Нагадаємо, що в 3-му семестрі в рамках лабораторної роботи №3 ми розробили об'єктно-орієнтоване програмне забезпечення, що дозволяє вирішувати певне коло завдань пов'язаних з моделюванням об'єкта управління та системи управління. Тоді ми розглядали технологічний процес підтримки ємності заданого рівня рідини. Тепер ГО підхід дозволить нам вирішити зовсім інше фізичне завдання з мінімумом змін у коді.

Більшість програмного коду без змін ми візьмемо зі старого проекту. Ви пам'ятаєте як ми в різних файлах реалізовували моделі ОУ та СУ, протоколювання, коди повідомлень, інтерфейс користувача і т.д. Такий підхід дозволить нам тепер просто взяти необхідні файли та підключити їх до нового проекту.

Як і раніше, рекомендується почати реалізацію ПЗ з розробки інтерфейсу користувача. Спеціальних вимог до зовнішнього вигляду інтерфейсу немає. У той же час він повинен забезпечувати виконання всіх вимог до введення та виведення інформації, наприклад, таких як забезпечення введення напруги або індикація поточної температури.

Після того як інтерфейс буде створено (створена форма вікна та заготовка всіх необхідних обробників) можна переходити до програмної реалізації об'єктно-орієнтованої моделі навчального стенду. Т.к. модель майже повністю повторює розроблену раніше в лабораторній роботі №3 3го семестру з дисципліни КТіОВР, то й програмний код багато в чому буде ідентичний розробленому раніше.

Студенти можуть взяти за основу розроблене раніше ПЗ і модифікувати його, або створити повністю нове ПЗ, користуючись методичними вказівками до лабораторної роботи №3 3го семестру.

Під час роботи в програмному коді необхідно буде внести такі зміни.

У файлі `SensorData.h` необхідно буде змінити атрибути ОУ на нові, відповідні розв'язуваної задачі. До атрибутів СУ необхідно буде додати інтегральну та диференціальну складові закону управління.

У файлі `Init.h` необхідно буде переробити функцію `LoadInitData()`. Тепер вона має завантажувати з текстового файлу `init.txt` нові атрибути відповідні задачі, що вирішується. Відповідно треба буде змінити вміст файлу `init.txt` створивши його формат даних.

У файлі `ControlObject.h` зміни також пов'язані з новими атрибутами. Змінити потрібно буде параметри конструктора `CControlObject()` та власне перелік атрибутів ОУ.

Ці зміни необхідно буде внести в реалізацію конструктора класу `CControlObject` у файлі `ControlObject.cpp`. У цьому файлі потрібно буде змінити функцію `ActionCalc()`. Нагадаємо, що в ній буде реалізована власне математична модель процесу нагрівання платівки.

Аналогічні зміни необхідно буде внести до файлів `ControlSystem.h`, `ControlSystem.cpp`. Так само зміни підлягають конструктор класу `CControlSystem` та функція `ActionCalc()`.

У файлі `BaseFunction.cpp` необхідно буде модифікувати функцію `InitModel()`. Зміни будуть пов'язані з передачею до конструкторів ОУ та СУ відповідних параметрів.

У файлі `msgcode.h` необхідно буде змінити коди команд та повідомлень так, щоб вони відповідали розв'язуваному завданню. Це призведе до необхідності скоригувати програмний код у файлах `proto.h` (протоколювання повідомлень), `MsgManager.h` і `MsgManager.cpp` (клас обміну повідомленнями у напрямку об'єкт – інтерфейс користувача).

Інші файли, такі як: `BaseFunction.h`, `BaseObject.h`, `BaseObject.cpp`, `defines.h`, `IMsgReceive.h`, залишаться без змін

Окремого пояснення вимагає алгоритм реалізації ПІД-закону регулювання:

$$\mu_i = \mu_{i-1} \pm (K_p \varepsilon_i + K_i \sum_{j=i-9}^i \varepsilon_j \tau + K_d \frac{\varepsilon_i - \varepsilon_{i-1}}{\tau}) \quad (*)$$

Для реалізації суми в інтегральній складовій пропонується використовувати кільцевий масив, а для обчислення диференціальної складової доведеться зберігати попереднє значення ε_{i-1} . Приблизно реалізація виразу (*) виглядатиме так.

```
double dEpsilon = Tzad - T;
m_dEpsilon[m_nPointer] = dEpsilon;
if ( m_nPointer < 9 )
{
    m_nPointer++;
}
else
{
    m_nPointer = 0;
}
double dInt = 0;
for ( int i = 0; i < 10; i++ )
{
    dInt = dInt + m_dEpsilon[i] * m_dTime;
}
double dP = dEpsilon * K;
double dI = dInt * I;
double dD = (dEpsilon - m_dEpsilon_prev) / m_dTime;
```

```
m_dParam = m_dParam + (dP + dI + dD);
if (m_dParam < min) m_dParam = min;
if ( m_dParam > max ) m_dParam = max;
```

Пояснимо наведений код.

В рядку

```
double dEpsilon = Tzad - T;
```

здійснюється розрахунок поточного відхилення температури T від заданого значення T_{zad} .

`m_dEpsilon`- Речовий масив на 10 елементів. Він виконує роль кільцевого масиву. Для операцій з ним використовується цілісний покажчик на елемент масиву `m_nPointer` (Не плутайте з вказівником у мові C++).

У ділянці коду

```
m_dEpsilon[m_nPointer] = dEpsilon;
if ( m_nPointer < 9 )
{
    m_nPointer++;
}
else
{
    m_nPointer = 0;
}
```

заноситься до масиву поточне відхилення і здійснюється перерахунок покажчика. Якщо ми не досягли кінця масиву, то вказівник збільшується на 1. Якщо ми знаходимося в кінці масиву, то вказівник обнулюється. Таким чином ми як би закріплюємо масив і реалізуємо механізм при якому нові дані весь час записуються повір'я старих. В результаті ми постійно зберігаємо 10 останніх значень відхилень.

Далі методом прямокутників розраховується інтеграл

```
double dInt = 0;
for ( int i = 0; i < 10; i++ )
{
    dInt = dInt + m_dEpsilon[i] * m_dTime;
}
```

У цій ділянці коду `m_dTime` час моделювання однієї ітерації. Фактично це інтервал спрацьовування таймера, який відповідає за посилку повідомлення `CmdCalc` системі керування.

Після розрахунку інтеграла проводиться розрахунок П-, І- та Д-складових

```
double dKp = dEpsilon * K;
double dKi = dInt * I;
double dKd = (dEpsilon - m_dEpsilon_prev) / m_dTime;
```

В рядку

```
m_dParam = m_dParam + (dP + dI + dD);
```

здійснюється розрахунок нового значення регульованої величини `m_dParam`. Зверніть увагу, що перед дужкою може стояти як знак «+» так і знак «-».

Наприкінці здійснюється перевірка чи не вийшла регульована величина за допустимі межі

```
if (m_dParam < min) m_dParam = min;  
if ( m_dParam > max ) m_dParam = max;
```

якщо так, то їй надається відповідне значення.

Ще раз зазначимо, що наведений орієнтовний код. Конкретна його реалізація залежить від конкретних імен преміальних та атрибутів оголошених вами для зберігання тих чи інших величин.

Тестування

Тестування починається з розробки програми та методики випробувань. Методика розробляється відповідно до нормативної бази та має забезпечувати перевірку виконання всіх вимог виставлених до цього ПЗ.

Захист лабораторної роботи полягатиме у перевірці всіх пунктів програми та методики випробувань.

ЛАБОРАТОРНА РОБОТА 3

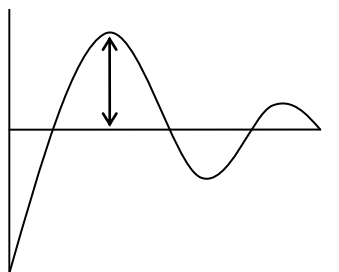
Мета роботи: здобути навички постановки наукових експериментів.

Завдання:

- Дослідити якість перехідних процесів у платівці при П, ПІ, ПІД законах управління;
- Дослідити вплив геометричних параметрів пластинки на перехідний процес при П, ПІ, ПІД законах управління;
- Дослідити вплив фізичних властивостей пластинки на перехідний процес при П, ПІ, ПІД законах управління.

Критеріями якості перехідного процесу вибрати:

- Тривалість перехідного процесу;
- максимальне абсолютне відхилення регульованої величини від заданого значення (рис. 4)



Мал. 4. Максимальне відхилення регульованої величини заданого значення.

Зміна геометричних та фізичних властивостей пластинки прийняти в діапазоні від 0.2 до 10 разів щодо початкового значення.