|  |  |
| --- | --- |
| Міністерство освіти та науки Українинаціональний університет «Одеська політехніка»інститут штучного інтелекту та робототехніки **Кафедра програмних та комп'ютерно-інтегрованих технологій**  Методичні вказівки з дисципліні сучасні системи управління (Теоретична частина)  Для студентів інституту штучного інтелекту та робототехніки  Інший (магістерський) рівень вищої освіти  Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології  Освітньо-професійна програма: Комп'ютерні технології автоматизації;  Кількість рік/кредитів ЄКТС за навчальним планом: 135/4,5 | |
|  | Схвалено на засіданні кафедри ПКІТ протокол №7 від 26.01.2022р. |
| **Одеса 2022** | |

Методичні вказівки з дисципліни «Сучасні системи управління» (Теоретична частина) для студ. спец. 151 – «Автоматизація та комп'ютерно-інтегровані технології» ден. та заоч. форм навч./уклад.: В.О. Давидов - Одеса: НУ "Одеська політехніка", 2022. - 54 с.

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

Зміст

[1 Автоматизація у пакетах word та excel 4](#_Toc102839652)

[1.1 Автоматизація у діловодстві 4](#_Toc102839653)

[1.2 Злиття документів Word та Excel 10](#_Toc102839654)

[2 Принципи побудови ПІД регуляторів 16](#_Toc102839655)

[2.1 Вступ 16](#_Toc102839656)

[2.2 Класичний ПІД-регулятор 16](#_Toc102839657)

[2.3 Модифікації ПІД-регуляторів 21](#_Toc102839658)

[2.4 Коректна реалізація різницевої схеми ПІД регулятора 29](#_Toc102839659)

[3 Адаптивні системи управління 32](#_Toc102839660)

[3.1 Системи зі змінною структурою 32](#_Toc102839661)

[3.2 Екстремальні системи керування 33](#_Toc102839662)

[4 Нечіткі регулятори 39](#_Toc102839663)

[4.1 Постановка задачі 39](#_Toc102839664)

[4.2 Динамічна модель об'єкту 40](#_Toc102839665)

[4.3 Налаштування регулятора 40](#_Toc102839666)

[4.4 Регулятор на основі нечіткої логіки 41](#_Toc102839667)

[4.5 Трохи теорії 42](#_Toc102839668)

[4.6 Синтез регулятора на основі нечіткої логіки 45](#_Toc102839669)

[4.7 Порівняння перехідних процесів 45](#_Toc102839670)

[4.8 Налаштування регулятора на основі нечіткої логіки шляхом оптимізації 47](#_Toc102839671)

[4.9 Створення власного регулятора на основі нечіткої логіки 48](#_Toc102839672)

[Висновки 52](#_Toc102839673)

[5 Історія штучного інтелекту 53](#_Toc102839674)

[5.1 Умовні позначення 57](#_Toc102839675)

[5.2 Список невикористовуваної літератури 57](#_Toc102839676)

[5.3 Основні положення 58](#_Toc102839677)

[5.4 Як це буде виглядати 59](#_Toc102839678)

[5.5 Епілог 61](#_Toc102839679)

# 1 Автоматизація у пакетах word та excel

## 1.1 Автоматизація у діловодстві

Microsoft Excel - один із найпопулярніших інструментів аналізу даних у світі. Все більше фірм використовують це програмне забезпечення і багато людей щодня користуються Excel. Однак, лише деякі з них повною мірою використовують можливості, які може запропонувати ця програма. Нижче наведено список із 20 найкорисніших формул Excel, які кожен має використовувати для підвищення ефективності. Функції Excel також дуже рекомендуються для обчислень, оскільки можуть значно зменшити кількість помилок. І останнє, але не менш важливе: вивчення цих функцій – чудова можливість покращити Ваші знання Excel та стати впевненим користувачем MS Excel.

1) Sum

"Sum" - це, напевно, найпростіша, але й найважливіша функція Excel. Ви можете використовувати цю формулу для обчислення суми діапазону осередків.

= SUM(A1:A10) або = SUM(A1,A10)

У цьому прикладі перша формула підсумовує осередки від A1 до A10. Друга підсумовує тільки осередки A1 та A10. Ви можете спробувати з ним і спробувати використовувати різні діапазони, щоб звикнути до цієї функції. Зверніть увагу, що ви можете використовувати числа всередині цієї функції замість посилань.

2) Average

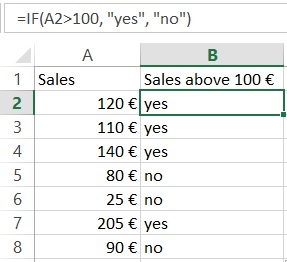
Ще одна дуже важлива формула. Як випливає з назви, ця формула оцінює середнє число діапазону осередків. Припустимо, що маємо діапазон чисел у стовпці A.

= AVERAGE(A1:A10)

Це поверне середнє значення осередків від A1 до A10.

3) If

Інша дуже потрібна формула. Вона відносно проста у використанні, але й дуже потужна. Ви можете використовувати цю функцію, щоб перевірити, чи даний оператор є істинним або помилковим, і щоб повернути вказане число.

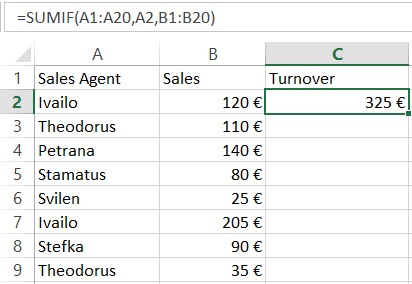


= IF(A2>100,"yes","no")

Тут формула перевірить, якщо продажі в осередку «A2» більше 100 €. Якщо ця заява правильна, формула повертає «так». Якщо продаж нижче 100 євро, формула повертає «ні».

4) Sumif

Sumif – ще одна дуже важлива формула Excel. Формула сумуватиме числа, якщо виконується певна умова.

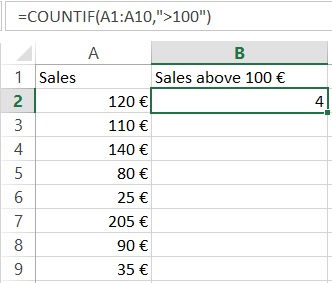


= SUMIF(A1:A20,A2,B1:B20)

Давайте розглянемо наведений вище приклад. У нас є торгові агенти, які реєструють певну кількість продажів. Ми хочемо дізнатися про загальну кількість продажів для кожного агента. Функція = SUMIF (A1: A20, A2, B1: B20) підсумовує значення осередків B1: B20, які відповідають тексту в осередку A2 (у випадку "Івайло"). Хлопець зареєстрував загальний обсяг продажів 325€. Ми можемо використати формулу для розрахунку обігу інших агентів.

5) Countif

Countif – дуже корисна функція, яка працює як sumif. Він підсумовується за певної умови. Різниця між sum і count полягає в тому, що функція count підсумовує кількість осередків, де певна умова є істинною. Навпаки, функція, що підсумовує, підсумовує значення всередині осередків.



= COUNTIF(A1:A10,">100")

Формула у наведеному вище прикладі перевіряє, скільки продажів були вищими ніж 100 євро і повертає кількість продажів. Ви також можете включити більше одного критерію, використовуючи countif для цієї мети.

PS: Якщо ви не впевнені в синтаксисі функції, ви завжди можете використовувати функцію insert function в Excel (просто натисніть знак функції в лівій частині функціональної панелі). У цьому прикладі ми продемонструємо, як це працює.

6) Counta

Ця формула підраховує кількість осередків певного діапазону, які є порожніми. Це дуже корисно, коли у вас є тисячі рядків і ви хочете знати, де все це закінчується.

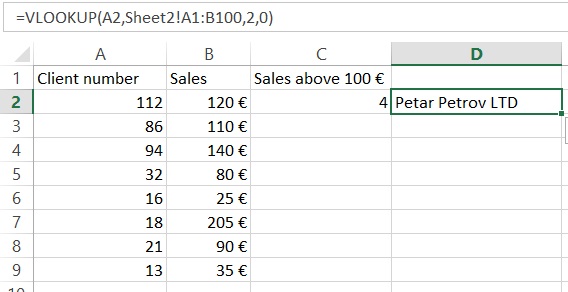
= COUNTA(A:A)

Це дасть вам кількість рядків, які містять значення, текст чи будь-який інший символ.

7) Vlookup

Ймовірно, найпотужніша функція Excel. Дуже корисно, коли вам доводиться мати справу з більш ніж однією таблицею, різними аркушами або іншого файлу. Vlookup дозволяє вам отримувати дані з однієї таблиці до іншої.

Продовжимо цей приклад, додавши ще один стовпець "А" з номерами клієнтів.



Зверніть увагу, що є інформація про продаж та номери клієнтів, але імена клієнтів відсутні. Однак у нас є ще один лист Excel (Sheet2) з іменами клієнтів та номерами клієнтів.



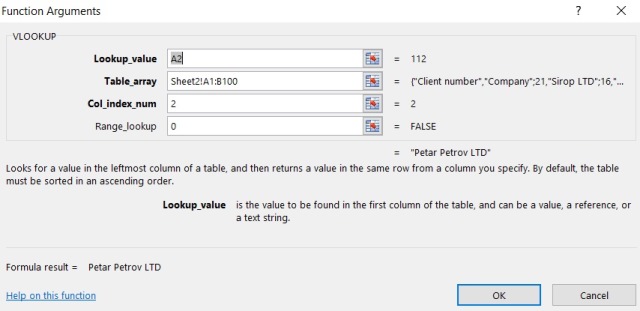
Замість того, щоб витрачати свій час, щоб подивитися з Sheet1 на Sheet2 і шукати збіг вручну для кожного окремого рядка, ми можемо використати таку формулу:

= VLOOKUP(A2,Sheet2!$A$1:$B$100,2,0)

Ця формула шукає значення в осередку A2 (112 у нашому випадку) у листі2, стовпці A & B (знаки «$» означають, що ми завжди хочемо мати постійний діапазон пошуку). Потім функція повертає значення другого стовпця. Остання частина формули ми пишемо 0. Таким чином, ми робимо Excel не шукати збігу в порядку чисел.

Спробуйте створити власну формулу vlookup, використовуючи опцію insert function.

У нашому випадку це виглядає так:



8) Left, Right, Mid

Left, right та mid функції - дуже важливі функції, які дозволяють витягувати певну кількість символів з рядка

= Left(A1,1) or = MID(A1,4,6) or = RIGHT(A1,5)

У прикладі з формулою left ми беремо перший символ з лівого боку. З формулою right ми можемо зробити те саме, але з символами справа наліво. Майте на увазі, що порожній простір також є символом.

9) Trim

Ще одна виняткова функція MS Excel. Функція trim видаляє порожні місця після слів, коли їх більше одного. Це може статися досить часто в Excel. Можливо, що інформація витягується з різних баз даних та заповнюється непотрібними порожнім простором між словами. Це може бути величезною проблемою, тому що ваші формули Excel можуть не працювати. Щоб позбавитися дратівливих порожніх просторів, ми використовуємо формулу trim. = TRIM(A1)

У цьому прикладі формула видалить зайві прогалини, якщо вона знайде кілька.

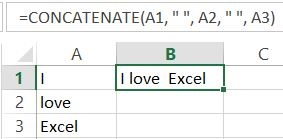
10) Concatenate

Це зручна функція Excel, яка допомагає, коли хочемо об'єднати осередки. Тут ми хочемо об'єднати осередки A1, A2 та A3. Це можна зробити за допомогою наступної функції:

= CONCATENATE(A1,A2,A3)

Таким чином, у нас не буде прогалин між словами. Щоб додати пробіли між "I" та "love", нам потрібно додати пробіли. Це можна вирішити, додавши лапки всередині функції:

= CONCATENATE(A1," ",A2," ",A3)



PS: Ви також можете об'єднати осередки, просто додавши:

= A1&" "&A2&" "&A3

Спробуйте самі!

11) Len

Ще одна прекрасна формула, яка може застосовуватись у багатьох ситуаціях. Вона широко використовується для побудови складніших формул.

= LEN(A1)

Ця функція підраховуватиме кількість символів у комірці.

12) Max

Функція max повертає найбільше значення діапазону.

= MAX(A2:A8)

Ця формула буде виглядати в осередках A2: A8, і тоді вона матиме найвище значення з цього діапазону.

13) Min

Функція min функціонує аналогічно до функції max, проте вона дає протилежний результат. Він отримує найнижче значення з діапазону даних.

= MIN(A2:A8)

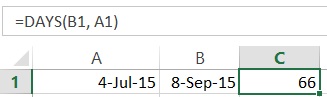
Формула з наведеного вище прикладу поверне найменшу кількість вибраних осередків у стовпці «A».

14) Days

Це функція, коли потрібно обчислити кількість днів між двома днями в електронній таблиці.

= DAYS(B1,A1)

Формула поверне кількість минулих днів.



15) Networkdays

Іноді ви хочете працювати лише з робочими днями. Ця функція працює дуже схоже на наведений вище приклад. Різниця в тому, що він повертає робочі дні, що минули, за винятком вихідних.

16) SQRT

Це досить просто і досить корисно водночас. Як випливає з назви, формула дає квадратний корінь із числа

= SQRT(1444)

Це поверне квадратний корінь із 1444 (38). Ви також можете використовувати посилання на комірку тут, як і в інших формулах Excel.

17. Now

Іноді потрібно знати поточну дату, коли ви відкриваєте електронну таблицю Excel.

= NOW()

Це поверне поточну дату. Не забудьте встановити формат на сьогоднішній день. Ви можете спробувати з ним стільки, скільки хочете. Ви можете, наприклад, додавати або віднімати дні. = Now() -14 поверне дату до двох тижнів.

18) Round

Функція Round дуже корисна, коли вам доводиться працювати із заокругленими номерами. Excel може відображати числа для певного символу, але зберігає вихідний номер і використовує для обчислень. Однак у деяких випадках це може бути проблемою, і нам потрібно працювати тільки із заокругленими номерами. У таких випадках функція ROUND звертається до нашої допомоги. = ROUND(B1,2)

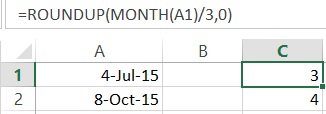
У цьому прикладі число заокруглюється до другого десяткового знака.

19) Roundup

Це варіація формули round, яка може дуже допомогти у деяких випадках. Вона округляє значення до найближчого цілого числа. Це дуже корисно у поєднанні з іншими формулами Excel. Давайте подивимося на цей приклад:

= ROUNDUP(MONTH(B3)/3,0)

Тут ми хочемо знайти, у якому кварталі поточного року знаходиться поточну дату. Секрет цієї формули – проста математика. Функція приймає номер місяця, наприклад, січень дорівнює 1, лютий дорівнює 2 і т. д., а потім ділить його на 3, округляючи до найближчого більшого числа.



20. Rounddown

Rounddown працює аналогічно Roundup, проте вона округляє число до найближчого цілого числа

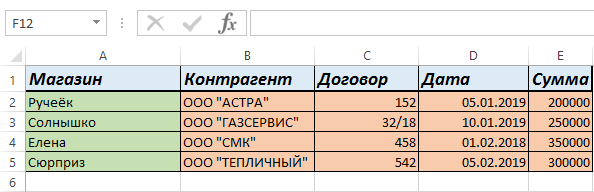
= ROUNDDOWN(20.65,0)

Це поверне двадцять.

## 1.2 Злиття документів Word та Excel

Розглянемо задачу. Використовуючи "базу" контактів магазинів з книги Excel, завантажити в договір, складений у програмі Word.

Маємо:



Також є порожній документ Word. До нього потрібно підставити дані.

Скористайтеся вбудованою функцією "Злиття", доступною в пакеті Microsoft Office.

Суть функції злиття - з файлу Excel беруться заголовки, ними розставлятимуться поля заповнення у документі Word. Дані будуть автоматично підставлятися за зміною режиму перегляду.

Зазначимо, що заголовки стовпців обов'язково мають бути унікальними! Надалі вони використовуватимуться як ідентифікатори відповідних полів.

У способу є багато мінусів (порівняно з макросами), але для повсякденного та швидкого використання цілком пригодиться.

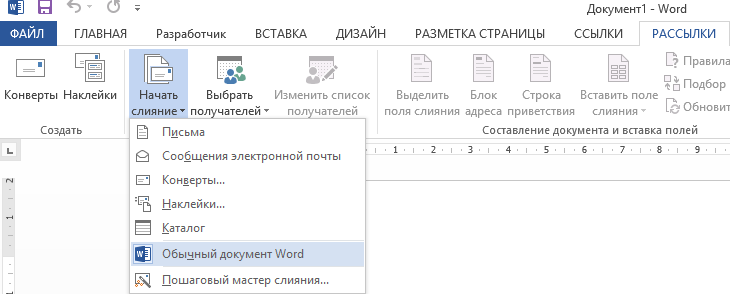
Приступимо:

- Створимо порожній документ;

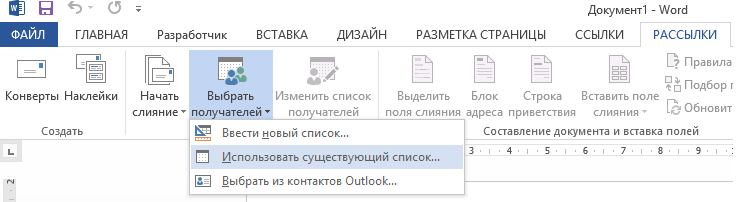
- Перейдемо на вкладку "Розсилки";

- у блоці кнопок "Початок злиття" натискаємо кнопку "Почати злиття";

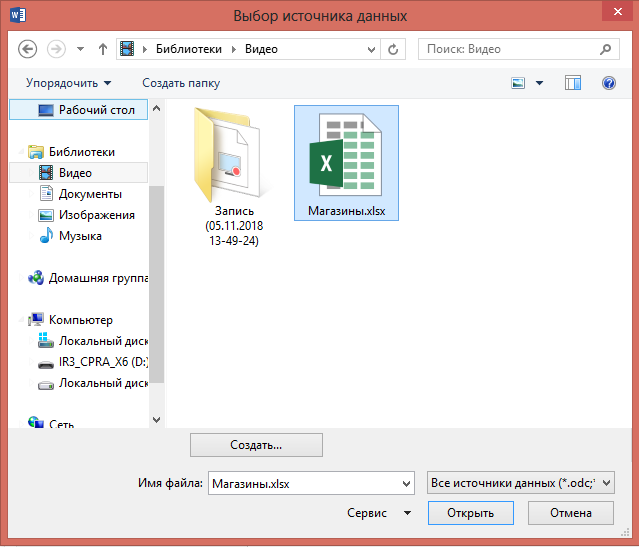
- Вибираємо пункт "Звичайний документ Word".



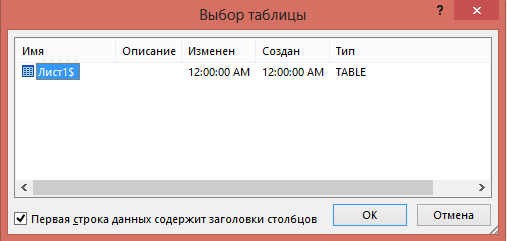
Спочатку нічого не зміниться. Тепер варто зазначити – звідки отримати інформацію. У тому ж блоці кнопок натисніть кнопку "Вибір одержувачів" та виберемо пункт "Використовувати існуючий список".



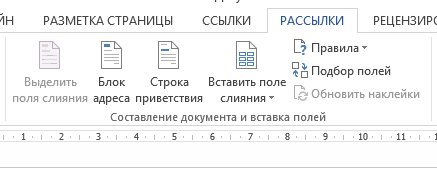
Далі відобразиться вікно вибору файлу джерела. Вкажемо шлях до файлу з даними магазинами. Натисніть на файл і натисніть "Відкрити".



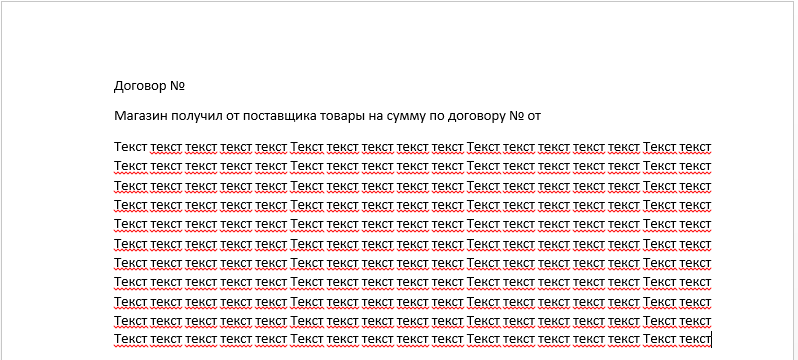
У вікні необхідно підтвердити джерело (звідки завантажуватимуться дані), на прикладі це буде Лист1. Обов'язково має стояти галка "Перший рядок містить заголовки стовпців".



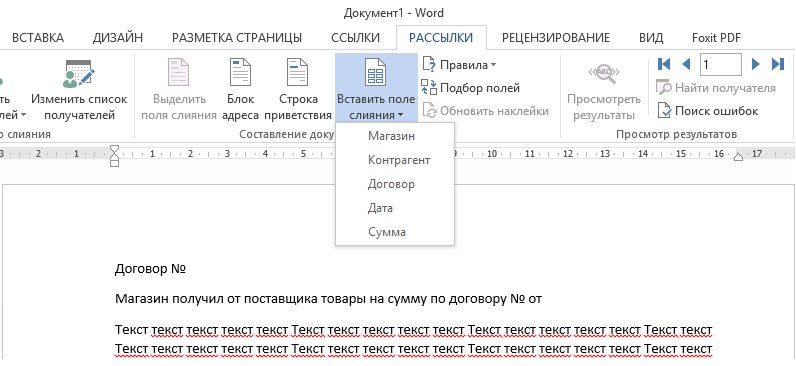
Натискаємо "ОК" і нічого не змінюється. Крім однієї деталі - стають активними дії в блоці кнопок "Складання документа та вставка полів. Він нам і потрібен.



Відкинемо перші три кнопки - у нашому випадку вони не знадобляться і надрукуємо невеликий текст, де будуть згадані заголовки даних книги Excel. Це дозволить надалі додати "Поля злиття".

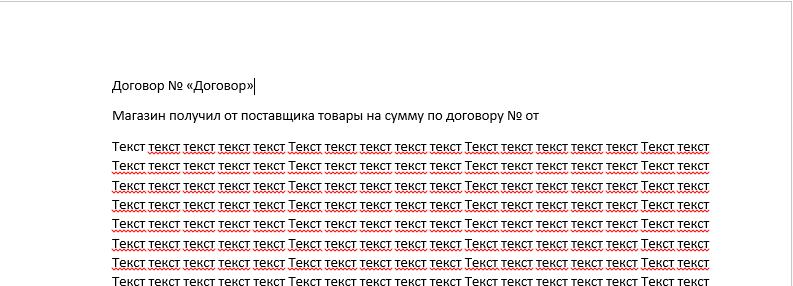


Зробимо так, щоб наші дані "підтягнулися" до документа. Натискаємо кнопку "Вставити поле злиття" і бачимо, що всі заголовки таблиці є полями злиття, іншою мовою - можуть завантажуватися в місце документа, куди були поміщені користувачем.

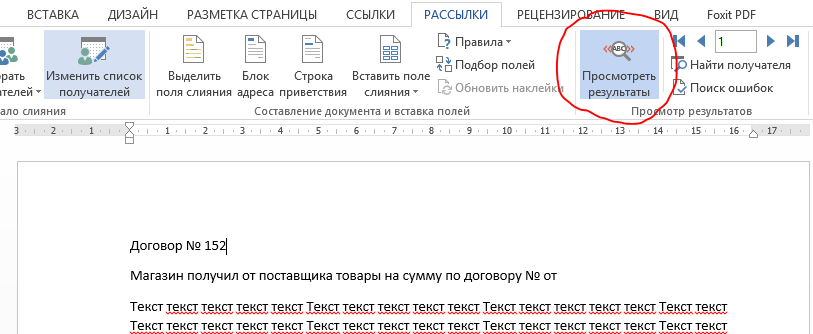


Найголовніше - дивитися куди в даний момент встановлений курсор, тому що саме в цю частину тексту додаватиметься поле злиття.

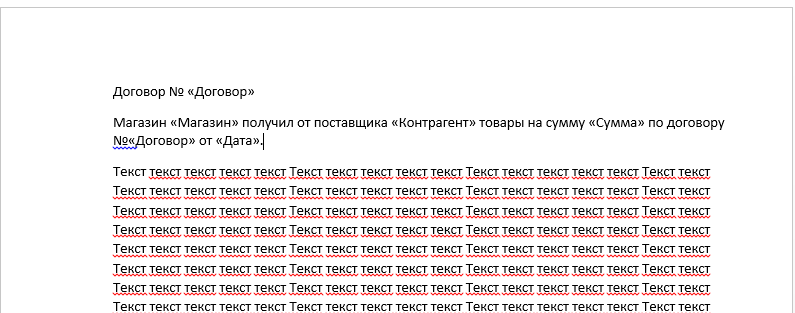
Встановимо курсор у заголовок після символу "№", потім виберемо зі списку "Вставити поле злиття" пункт "Договір".



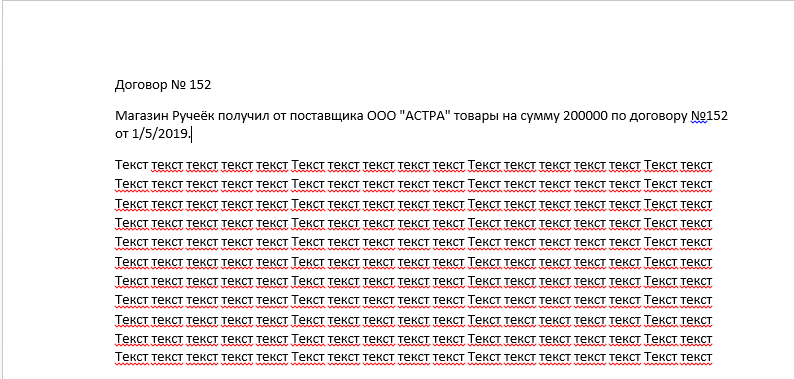
З'явиться слово "Договір" у лапках. Дивно, але ні:) Тепер потрібно натиснути кнопку "Переглянути результати". Картина буде іншою. Після номера договору підвантажився його номер по магазину "Струмок" - 152.



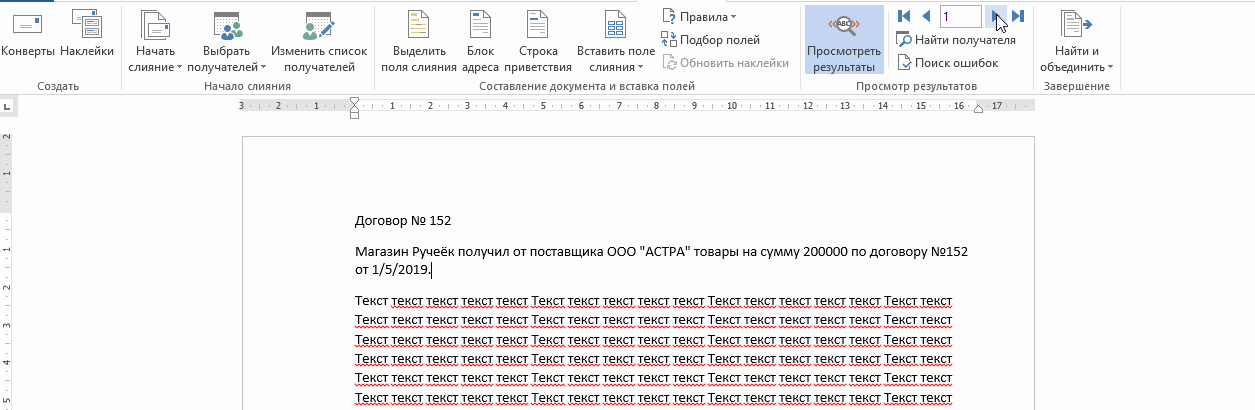
Заповнимо всі поля - клацаємо після слова "Магазин" і вибираємо пункт "Магазин", після слова "постачальника" і вибираємо поле "Контрагент", після "суму" вибираємо пункт "Сума", після "№" "Договір", після "від "Дата".



Знову натискаємо кнопку "Переглянути результати".

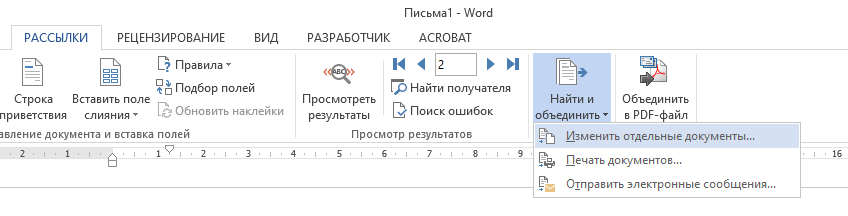


Дані підвантажилися. Як їх змінити? Поруч із кнопкою "Переглянути результати" є стрілка. Клацаючи на неї можна побачити, як змінюються дані.



Ось так можна швидко підвантажити дані з Excel у Word.

Розглянутий механізм дозволяє спростити процедуру створення безлічі документів за єдиним шаблоном, наприклад, типові листи всім клієнтам з «бази». Для цього після завершення роботи над шаблоном необхідно натиснути кнопку «Знайти та об'єднати» і вибрати пункт «Змінити окремі документи». Після цього всі документи, сформовані за вашим шаблоном на основі даних із файлу Excel, будуть об'єднані в рамках одного документа.



# 2 Принципи побудови ПІД регуляторів

## 2.1 Вступ

ПІД-регулятор був винайдений ще 1910 року [1]. Через 32 роки, в 1942 році, Зіглер і Нікольс розробили методику його налаштування [2]. Після появи мікропроцесорів у 1980-х роках розвиток ПІД-регуляторів відбувається наростаючими темпами. Загальна кількість публікацій з ПІД-регуляторів за 9 років з 1973 по 1982 рік склала 14 шт., з 1983 по 1992 рік - 111 шт., А за період з 1998 по 2002 рік (всього за 4 роки) - 225 шт. [3]. На одному семінарі IFAC (International Federation of Automatic Control) у 2000 році було представлено близько 90 доповідей, присвячених ПІД-регуляторам [4]. Кількість патентів на цю тему, що містяться в патентній базі даних http://gb.espacenet.com, у січні 2006 року становила 364 шт.

ПІД-регулятор відноситься до найпоширенішого типу регуляторів. Близько 90-95% регуляторів [1, 5], що знаходяться в даний час в експлуатації, використовують ПІД-алгоритм. Причинами такої високої популярності є простота побудови та промислового використання, ясність функціонування, придатність для вирішення більшості практичних завдань та низька вартість. Серед ПІД-регуляторів 64% посідає одноконтурні регулятори і 36% – багатоконтурні [6]. Контролери зі зворотним зв'язком охоплюють 85% всіх додатків, контролери з прямим зв'язком – 6%, і контролери, з'єднані каскадно, – 9% [6].

Після появи дешевих мікропроцесорів та аналогоцифрових перетворювачів у промислових ПІД-регуляторах використовуються автоматичне налаштування параметрів, адаптивні алгоритми, нейронні мережі, генетичні алгоритми, методи нечіткої логіки. Ускладнилася структура регуляторів: з'явилися регулятори з двома ступенями свободи, із застосуванням принципів розімкнутого управління у поєднанні зі зворотним зв'язком, із вбудованою моделлю процесу. Крім функції регулювання, ПІД-контролер були введені функції аварійної сигналізації, контролю розриву контуру регулювання, виходу за межі динамічного діапазону та ін.

Незважаючи на довгу історію розвитку та велику кількість публікацій, залишаються проблеми у питаннях усунення інтегрального насичення, регулювання об'єктів з гістерезисом та нелінійностями, автоматичного налаштування та адаптації. Практичні реалізації ПІД-контролерів не завжди містять антиаліасні фільтри, надмірний шум і зовнішні обурення ускладнюють налаштування параметрів. Проблеми ускладнюються тим, що у сучасних системах управління динаміка часто невідома, регульовані процеси не можна вважати незалежними, вимірювання сильно зашумлені, навантаження непостійне, технологічні процеси безперервні.

Далі розглянуті лише регулятори для одномірних об'єктів (з одним входом та одним виходом), для повільних (теплових) процесів, які найпоширеніші в АСУ ТП.

## 2.2 Класичний ПІД-регулятор

Найпростіша система автоматичного регулювання із зворотним зв'язком показана на рис. 1. У ній блок R називають регулятором, P – об'єктом регулювання, r – керуючим впливом, або уставкою, e – сигналом неузгодженості, або помилки, u – вихідною величиною регулятора, y – регульованою величиною.

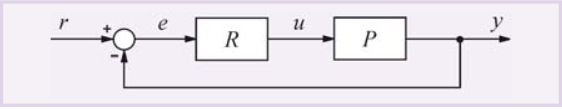


Рис. 1. ПІД-регулятор у системі зі зворотним зв'язком

Якщо вихідна змінна регулятора R u описується виразом:

 (1)

де t – час, а K, T i, T d – пропорційний коефіцієнт, постійна інтегрування та постійна диференціювання відповідно, такий регулятор називають ПІД-регулятором.

У окремому випадку пропорційна, інтегральна або диференціальна компоненти можуть бути відсутніми, і такі спрощені регулятори називають І-, П-, ПД- або ПІ-регуляторами. Поширені також такі модифікації виразу (1):

 (2)

 (3)

Між параметрами, що входять до виразів (1)-(3), існує простий зв'язок. Однак відсутність загальноприйнятої системи параметрів часто призводить до плутанини. Це потрібно пам'ятати при заміні одного ПІД-контролера на інший або використання програм налаштування параметрів. Ми будемо користуватися виразом (1).

Використовуючи перетворення Лапласа за нульових початкових умов, передавальну функцію ПІД-регулятора можна подати в операторній формі:

 (4)

де s - Комплексна частота. Амплітудно-частотна (АЧХ) та фазочастотна (ФЧХ) характеристики операторної передавальної функції (4) показані на рис. 2. У сфері нижніх частот АЧХ і ФЧХ визначаються інтегральним членом, у сфері середніх частот – пропорційним, у сфері високих – диференціальним.

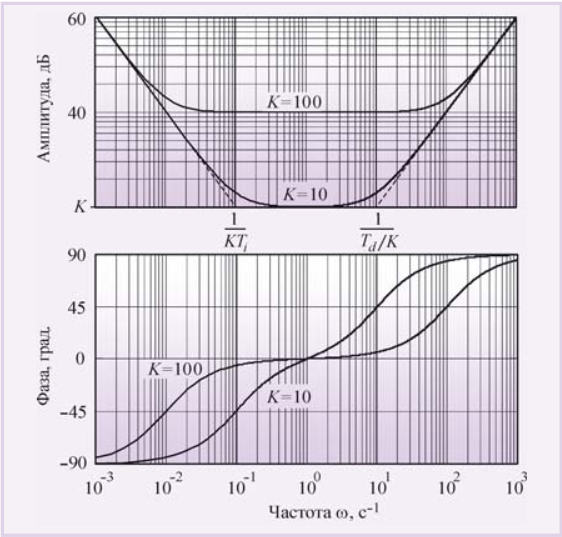


Рис. 2. АЧХ та ФЧХ ПІД-регулятора при T i = 1 с, T d = 1 с, K = 10 та K = 100

На систему автоматичного регулювання можуть впливати (рис. 3) зовнішні обурення d=d(s) та шум вимірювань n=n(s). Зовнішні обурення (вплив навантаження, зміна температури навколишнього середовища, вітер, перебіг води тощо) зазвичай просторово розподілені по об'єкту, проте для спрощення аналізу їх моделюють зосередженим джерелом d(s), що додається до входу системи. Джерело шуму n(s) моделює похибку вимірювань вихідною зміною y, похибку датчика, а також перешкоди [7, 8], що впливають на канал передачі сигналу з виходу системи на її вхід.

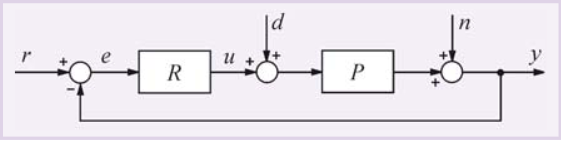


Рис. 3. ПІД-регулятор у системі з шумом n та зовнішніми обуреннями d

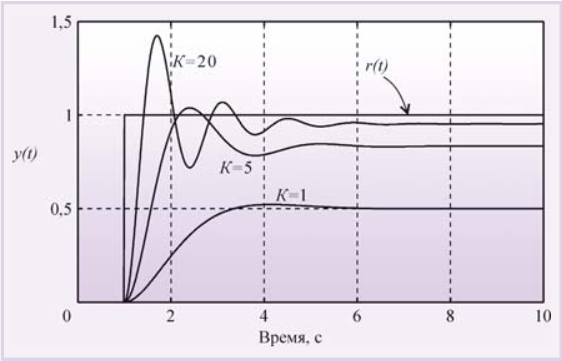


Рис. 4. Зміна змінної y у часі при подачі одиничного стрибка r(t) на вхід системи за різних K і T i = ∞∞, Td = 0 (П-регулятор)

Вид АЧХ та ФЧХ регулятора визначає його точність та запас стійкості. Зі зменшенням інтегральної складової Ti, як випливає з рис. 2, збільшується модуль коефіцієнта посилення регулятора на низьких частотах (тобто при наближенні до режиму, що встановився), і тому знижується похибка e. Зі збільшенням диференціальної складової Td зростає посилення на високих частотах, що призводить до посилення шумів вимірів та зовнішніх збурень. Тому диференціальну складову використовують лише поліпшення форми перехідного процесу у системі, та її практична реалізація зазвичай містить фільтр високих частот.

Зі зростанням пропорційного коефіцієнта K збільшуються модуль петльового посилення контуру регулювання та точність у всьому діапазоні частот, проте падає запас по фазі та посиленню, що погіршує робастність та якість регулювання системи, а при подальшому збільшенні K (рис. 4) виникають періодичні коливання (система втрачає стійкість). Вплив шуму та перешкод вимірювань n також зменшується зі зростанням петльового посилення та пропорційного коефіцієнта. На рис. 5 показані перехідні характеристики замкнутої системи з І-регулятором (тобто при K = 0, Td = 0) та об'єктом другого порядку з передатною функцією

 (5)

де T = 0,1 с. При великих постійних інтеграціях Ti перехідна характеристика має вигляд, подібний до характеристики аперіодичного ланки. Зі зменшенням Ti зростає посилення регулятора, і коли на деякій частоті петльове посилення контуру з зворотним зв'язком наближається до 1, в системі з'являються коливання.

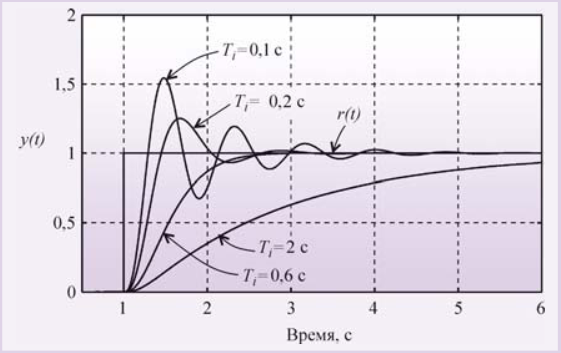


Рис. 5. Реакція на стрибок r(t) замкнутої системи 2го порядку (5) з Ірегулятором при T = 0,1 с та різних T1

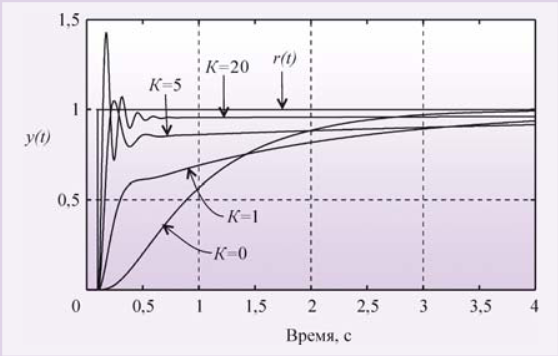


Рис. 6. Реакція замкнутої системи з ПІрегулятором на стрибок r(t) при Ti = 1 с для об'єкта виду (5) при T = 0,1 с

АЧХ ПІ-регулятора можна отримати із рис. 2, якщо відкинути праву гілка АЧХ із нахилом +20 дБ/дек. При цьому зсув фаз частотах вище 1 Гц не перевищить рівень 0°. Таким чином, ПІ-регулятор має дві істотні позитивні відмінності від І-регулятора: по-перше, його посилення на всіх частотах не може стати меншим K, отже, збільшується динамічна точність регулювання; по-друге, в порівнянні з І-регулятором він вносить додаткове зрушення фаз тільки в області низьких частот, що збільшує запас стійкості замкнутої системи. У той же час, як і в Ірегулятор, модуль коефіцієнта передачі регулятора зі зменшенням частоти прагне нескінченності, забезпечуючи тим самим нульову помилку в режимі. Відсутність зсуву фаз на високих частотах дозволяє збільшити швидкість наростання змінної керованої без зниження запасу стійкості. Однак це справедливо доти, поки пропорційний коефіцієнт K стане настільки великим, що збільшить посилення контуру до одиниці на частоті зсуву фаз 180° ( ω180).

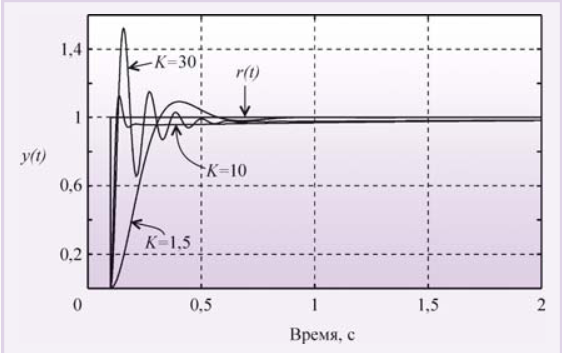


Рис. 7. Реакція замкнутої системи з ПІрегулятором на стрибок r(t) при Ti = 0,1 с для об'єкта виду (5) при T = 0,1 с

Перехідний процес у ПІ-регуляторі при різних поєднаннях Ti та K показаний на рис. 6 та 7. При K = 0 (рис. 6) отримуємо І-регулятор. Зі зростанням пропорційного коефіцієнта K з'являється додаткова помилка під час перехідного процесу (див. рис. 4), яка при подальшому збільшенні K зменшується, проте при цьому знижується запас стійкості системи, оскільки зі зростанням K збільшується посилення на частоті ω180 . Це призводить до появи загасаючих коливань на початку перехідного процесу (рис. 6). Коли величина K стає досить великий компенсації послаблення сигналу в об'єкті на частоті ω180, у системі з'являються незатухающие коливання.

Пропорційний коефіцієнт призводить до збільшення часу встановлення перехідного процесу за рівнем 0,99 порівняно з І-регулятором при тих же Ti та T (рис. 6): зі зростанням K зменшується нахил кривої y(t) при великих t; зокрема, при t = 4 з крива K = 1 проходить нижче за криву K = 0, а крива K = 5 проходить ще нижче. Пояснюється це так. Зменшення помилки e в ПІ-регуляторі досягається дією одночасно як пропорційного, так і інтегрального коефіцієнтів. Проте пропорційний коефіцієнт неспроможна звести помилку нанівець (рис. 4). Тому помилка e(t), що залишилася, зменшується з плином часу за допомогою члена, Який наростає тим повільніше, що менше e(t). У результаті запровадження пропорційного коефіцієнта, що зменшує e(t), призводить до затягування перехідного процесу.

У частотній області цей процес можна пояснити тим, що зі зростанням K нуль передавальної функції 1/KT i зміщується вліво (рис. 2), тобто розширюється область частот, де інтегральна складова дуже мала і ПІ-регулятор вироджується в чистий П-регулятор, якому властива помилка в режимі, що встановився.

У ПІД-регуляторі присутній диференціальний член, який, як випливає з рис. 2 вносить позитивний фазовий зсув до 90° на частотах вище K/Td. Це дозволяє забезпечити стійкість або покращити якість регулювання системи у випадках, коли це неможливо зробити за допомогою ПІ-регулятора. На рис. 8 показано вплив постійної диференціювання на форму відгуку замкнутої системи на стрибок r(t). Зменшення амплітуди коливань та збільшення коефіцієнта згасання зі зростанням постійної диференціювання Td пояснюється тим, що завдяки позитивному нахилу АЧХ в області ω > K/Td (рис. 2) зменшуються зсув фаз у контурі регулювання та петльове посилення.

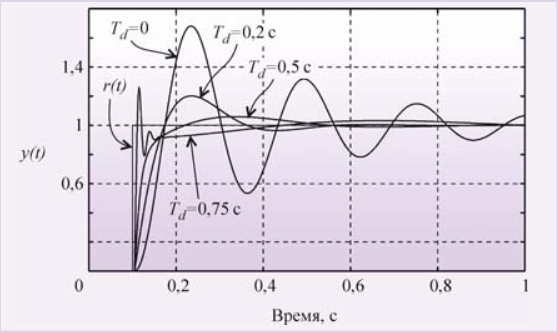


Рис. 8. Реакція замкнутої системи з ПІДрегулятором на стрибок r(t) при Ti = 0,015 с, K = 6 для об'єкта виду (5) при T = 0,1 c

Подальше збільшення постійної диференціювання призводить до зростання посилення ПІДрегулятора на високих частотах при ω > K/Td (рис. 2). Оскільки фазовий зсув, пов'язаний із транспортною затримкою, необмежено збільшується зі зростанням частоти, то в системі навіть з невеликою транспортною затримкою зі збільшенням Td завжди настає момент, коли петльове посилення на частоті фазового зсуву 180° перевищить одиницю. При цьому на перехідній характеристиці замкнутої системи спочатку з'являються коливання, що затухають (рис. 8, крива Td = 0,75 с), потім при подальшому збільшенні Td система переходить в коливальний режим.

Таким чином, зі зростанням постійної диференціювання запас стійкості замкнутої системи спочатку збільшується, потім падає.

## 2.3 Модифікації ПІД-регуляторів

Описаний у попередньому розділі ПІД-регулятор та його окремі випадки є теоретичними ідеалізаціями реальних регуляторів, для їх практичного здійснення необхідно врахувати обмеження, накладені реальними умовами застосування та технічної реалізації. Крім цього, наявність у ПІД-регуляторі лише трьох регульованих параметрів (K, Ti, Td) у ряді випадків виявляється недостатнім для отримання заданої якості регулювання, особливо для систем з великою транспортною затримкою L і для систем, в яких потрібні одночасно висока якість стеження за уставкою та висока якість ослаблення зовнішніх обурень.

Постійно зростаючі вимоги ринку до якісних показників ПІД-контролерів ініціюють появу безлічі нових модифікацій ПІДрегуляторів.

**Регулятор з ваговими коефіцієнтами при уставці**

У класичному ПІД-регуляторі сигнал помилки e дорівнює різниці між задаючою дією r і вихідною змінною об'єкта y: e = r – y. Однак якість регулювання можна покращити, якщо помилку обчислювати окремо для пропорційної, диференціальної та інтегральної складових [5] (рис. 9):

 (6)

де ep, ed, ei – помилки для пропорційної, диференціальної та інтегральної складових відповідно; b, c - настроювальні вагові коефіцієнти. Рівняння такого регулятора аналогічне (1):

 (7)

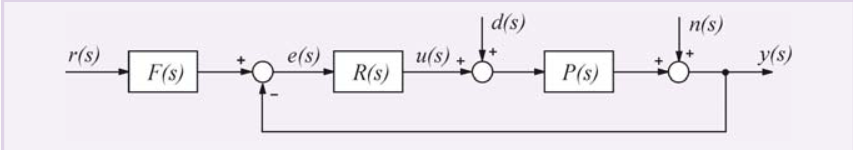
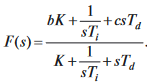


Рис. 10. Виділення блоку F(s) у структурі ПІД-регулятора

Зазначимо, що ваговий коефіцієнт при інтегральній складовій відсутній, що необхідно для забезпечення нульової помилки в режимі, що встановився. Можна довести, що регулятор представлений на рис. 9, повністю еквівалентний регулятору, показаному на рис. 10, якщо блок R(s) є класичним регулятором (4), а блок F(s) має передатну функцію вигляду:

 (8)

Структура отриманого регулятора має чудову властивість: блок F(s) не входить до контуру регулювання. Це означає, що робастність, якість регулювання, реакція на шуми і зовнішні збурення, як і раніше, будуть визначатися лише параметрами K, Ti, Td, тобто параметри b і c блоку F(s) налаштовуються незалежно від параметрів K, T i, Td.

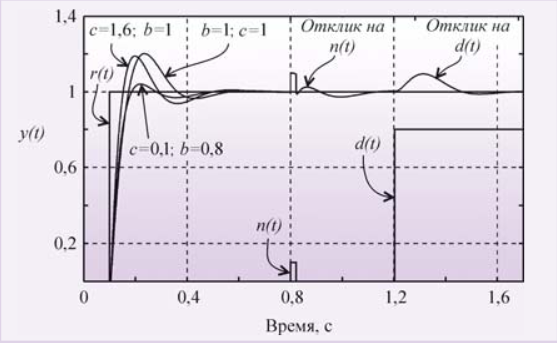


Рис. 11. Реакція замкнутої системи з регулятором на стрибок r(t) при Ti = 0,015, K = 6,  
Td = 0,2 с для об'єкта виду (5) при T = 0,1 с, L = 0,005 с (позначення відповідають рис. 10

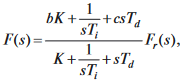
Параметри b і c визначають вид АЧХ блоку F(s) і дозволяють покращити якість реакції регулятора зміну уставки r(t). На рис. 11 показана реакція замкнутої системи з описаним регулятором при різних значеннях вагових коефіцієнтів b та c. Як очевидно з малюнка, зміна параметрів b і c впливає відгук системи на шуми n(t) і зовнішні обурення d(t).

Коефіцієнт c часто вибирають рівним нулю, щоб уникнути диференціювання випадкових різких викидів у сигналі r(t), якщо вони можливі.

Регулятор при b = 0 та c = 0 іноді називають І(ПД)-регулятором, а при b = 1 та c = 0 – ПІ(Д)-регулятором.

**Регулятор з фільтром для сигналу уставки**

Подальшим удосконаленням регулятора із структурою, показаною на рис. 10 є застосування фільтра в блоці F(s), передатна функція якого набуває вигляду:

 (9)

Де

 (10)

Тут Tr – постійна часу фільтра, яку вибирають рівною

 (11)

де mr – показник коливання системи без фільтра.

Приклад реакції системи з регулятором, що використовує формує фільтр, наведено на рис. 12.

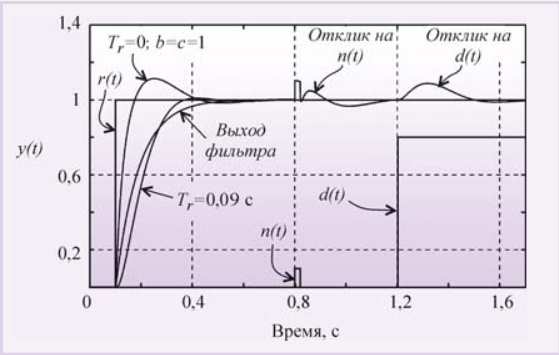


Рис. 12. Реакція системи з регулятором, який використовує фільтр (9) при T r = 0,09 с, на стрибок r(t) при Ti = 0,015 с, K = 6, T d = 0,3 с для об'єкта виду (5) при T = 0,1 с, L = 0,005 с (позначення відповідають рис. 10)

**Принцип розімкнутого управління в ПІД-регуляторах**

Регулятор можна побудувати без використання зворотного зв'язку. Якщо відомі діють на систему обурення і бажана реакція на зміну впливу, що управляє, то в деяких випадках можна знайти таку передатну функцію регулятора, при якій виходить бажана реакція системи. Перевагою такого підходу є висока швидкість реагування системи на зовнішні обурення, оскільки для вироблення керуючого впливу не потрібно чекати, поки керуючий сигнал пройде через об'єкт і повернеться в регулятор ланцюга зворотного зв'язку. Крім того, система з розімкненим управлінням у принципі не може бути нестійкою, оскільки в ній відсутня зворотний зв'язок.

Недоліком є ​​неможливість отримати високу точність при невідомих збуреннях та низькій точності моделі об'єкта, а також неможливість повної компенсації збурень для об'єктів з транспортною затримкою та проблема фізичної реалізації зворотних операторів (див. розділ «Знаходження зворотної динаміки об'єкта»).

У зарубіжній літературі системи із розімкненим управлінням називають системами із прямим зв'язком. Термін «прямий зв'язок» обраний у тому, щоб підкреслити відмінність цього від методу зворотний зв'язок. Далі обидва терміни будуть використані як синоніми.

Переваги розімкнутого та замкнутого управління можна поєднати в одному регуляторі. Найкращі характеристики системи виходять, якщо її проектувати за принципом розімкнути того управління, а зворотний зв'язок використовувати тільки для подальшої мінімізації похибки системи.

У попередньому розділі було розглянуто окремий випадок застосування прямого зв'язку (рис. 10), який реалізований за допомогою блоку F(s). Основним принципом застосування розімкнутого управління у ПІДрегуляторах є поділ завдання проектування на дві частини. Перша частина – забезпечення робастності та ослаблення впливу шумів та зовнішніх збурень – вирішується за допомогою параметрів K, T i , T d. Друга частина — забезпечення заданої реакцію керуючий вплив r(t) – вирішується з допомогою параметрів регулятора із прямим зв'язком. Регулятори, що забезпечують можливість незалежного вирішення цих двох завдань, називають регуляторами з двома ступенями свободи, і на їх умовному зображенні є два входи (рис. 13), [9].

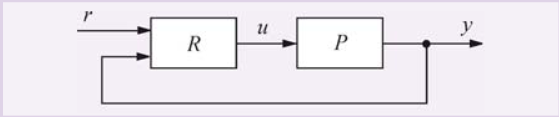


Рис. 13. Умовне позначення регулятора з двома ступенями свободи (R)

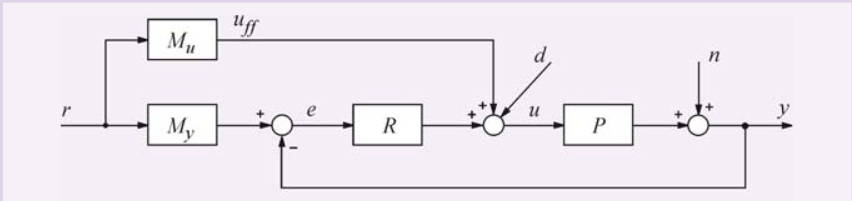


Рис. 14. Структура регулятора з двома ступенями свободи у замкнутій системі регулювання

Структура ПІДрегулятора, що використовується принцип розімкнутого управління, показана на рис. 14. Тут регулятор R спроектований, як у системі з класичним ПІДрегулятором, а передавальні функції блоків M u та M y вибираються так, щоб покращити реакцію системи на вхідний вплив r(t).

Принцип дії такої системи полягає у наступному. Зміна сигналу r(t) надходить на вхід об'єкта управління через блок Mu, минаючи ланцюг зворотного зв'язку. Передатна функція блоку Mu вибирається таким чином, щоб вихідний сигнал системи y(t) точно відповідав вхідному сигналу y(t) = r(t), тобто щоб сигнал помилки e(t) дорівнював нулю. Оскільки в реальній системі при впливі зовнішніх обурень або зміні уставки e(t) ≠ 0, то під час перехідного процесу вступає в дію звичайний ПІД-регулятор R, який за допомогою зворотного зв'язку намагається звести помилку, що з'явилася до нуля.

Безпосередньо за рис. 14 можна записати передавальну функцію системи від входу r на її вихід у вигляді:

 (12)

Тут перший член вибирають, як випливає з описаного раніше принципу дії системи, так, щоб у ідеальних умовах e = 0 і M yr = y, тобто бажаною функцією передачі системи є G = M y. Тому другий член (12) необхідно зробити рівним нулю. Цього можна досягти двома способами. Перший у тому, щоб зробити нескінченно великим петльове посилення PR. Найчастіше використовують другий шлях, який полягає у виборі такої передавальної функції Mu, щоб виконувати співвідношення PMu – My = 0, тобто

 (13)

Таким чином, на відміну від регулятора із зворотним зв'язком, у якого точність забезпечується завдяки поділу сигналу помилки на велике число (посилення інтегратора), в регуляторах з прямим зв'язком точність забезпечується шляхом компенсації помилки, тобто за допомогою операції віднімання.

Оскільки у системі, показаній на рис. 14, помилка на низьких частотах і в режимі дорівнює нулю завдяки інтегральному члену в ПІД-регуляторі R, високу точність компенсації помилки за допомогою прямого зв'язку достатньо забезпечити тільки на високих частотах. Це полегшує задачу синтезу передавальної функції Mu.

**Знаходження зворотної динаміки об'єкту**

Як випливає з (13), знаходження передавальної функції Mu необхідно знайти зворотний оператор P-1 (s). Завдяки формі алгебри зображень операторів по Лапласу, формально зробити це досить просто. Наприклад, для об'єкта з передатною функцієюзворотний оператор дорівнюватиме

 (14)

Однак такі операції наштовхуються на проблему фізичної реалізованості. Вираз (14) містить член e+sL, який є зворотним щодо ідеальної затримки, тобто є зображенням операції ідеального передбачення. З іншого боку, реалізації (14) необхідна операція ідеального диференціювання, реалізація якої також досить проблематична. Тому завдання синтезу регулятора з прямим зв'язком є ​​завданням апроксимації нереалізованої передавальної функції штучно обраної реалізованої функцією за критерієм мінімуму похибки.

**Регулятор з функцією передавання об'єкта**

У дуже простому випадку для систем з монотонним відгуком на ступінчасту дію вид переда точної функції My (s) можна вибрати збігається з нормованою передатною функцією об'єкта:

 (15)

де K p = P (0). Тоді відповідно (13)

 (16)

Недоліком такого підходу є повільна реакція замкнутої системи на зміну впливу, що задає. Перевагою є відсутність будь-яких розрахунків та налаштувань для гарантованого отримання відгуку без перерегулювання (рис. 15). Слід, однак, пам'ятати, що відгук замкнутої системи на вплив, що задає, ніяк не пов'язаний з відгуком на зовнішні обурення і шум, тому налаштування регулятора повинна бути виконана звичайними методами.

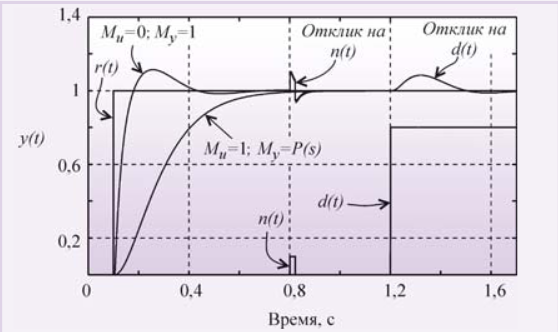


Рис. 15. Реакція системи, що має ПІДрегулятор з прямим зв'язком і передавальною функцією об'єкта, на стрибок r(t) при Ti = 0,015 с, K = 6, T d = 0,3 с для об'єкта виду (5) при T = 0,1 с, L = 0,005 с (позначення відповідають рис. 14)

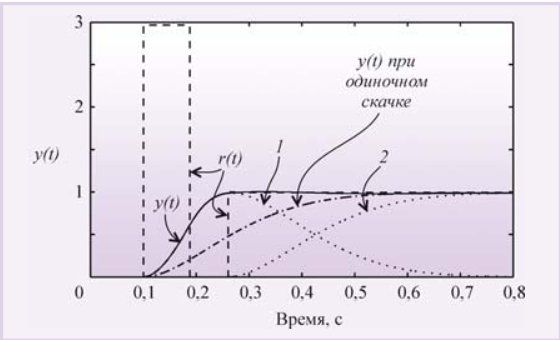


Рис. 16. Реакція системи, що містить ПІДрегулятор з прямокутним імпульсом перед сигналом уставки (див. розділ «Імпульсне керування без зворотного зв'язку»), при T i = 0,19 с, K = 0,8, Td = 0 для об'єкта виду (5) при T = 0,1, L = 0: крива 1 — реакція на прямокутний імпульс r(t); крива 2 - реакція на стрибок r(t)

**Імпульсне керування без зворотного зв'язку**

Ще одна модифікація принципу розімкнутого управління у тому, що перед подачею сигналу уставки на вхід об'єкта подають прямокутний імпульс великої амплітуди (рис. 16). Оскільки швидкість наростання реакцію прямокутний їм пульс пропорційна його амплітуді, тривалість перехідного процесу можна істотно зменшити порівняно з випадком, коли сигнал уставки подається у формі одиночного стрибка (рис. 16).

Реакція на прямокутний імпульс складається з фази наростання сигналу та фази спаду. Амплітуду імпульсу вибирають максимально можливим. Зазвичай вона обмежується потужністю виконавчих пристроїв системи. Тривалість імпульсу вибирають такий, щоб максимум реакції на імпульс дорівнював значенню уставки (одиниці при уставці у формі одиничного ступінчастого впливу). Затримку подачі ступінчастої дії вибирають так, щоб вона збіглася з моментом появи максимуму відгуку прямокутний імпульс.

У цьому методі час виходу системи режим може бути зроблено як завгодно малим, якщо використовувати імпульс досить великий амплітуди. У випадку перед подачею ступінчастого впливу можна подавати кілька імпульсів різної амплітуди і тривалості. Параметри імпульсів та ступінчастої дії вибирають, вирішуючи чисельними методами задачу оптимізації за критерієм мінімізації похибки відхилення відгуку системи від необхідної форми. Для лінійних систем параметри, отримані при оптимізації, залишаються без змін будь-яких значень уставки, якщо амплітуду прямокутного імпульсу змінювати пропорційно до значення уставки.

**Компенсація зовнішніх обурень за допомогою прямого зв'язку**

Якщо зовнішні обурення, що впливають об'єкт управління, можна вимірювати до того, як вони пройдуть на вихід системи y, їх вплив можна істотно послабити за допомогою прямого зв'язку. Прямий зв'язок дозволяє компенсувати похибку швидше, ніж зворотний виявить помилку як різницю між керованою величиною і керуючим впливом.

Раніше ми припускали, що зовнішні обурення долучені до входу системи. Таке припущення було справедливо за якісного аналізу ступеня придушення обурень за допомогою зворотного зв'язку. Однак для компенсації збурень необхідно ідентифікувати передатну функцію від точки додатка збурень до виходу системи Pd(s). При цьому об'єкт управління отримує другий вхід (вхід обурень) і описується функцією з двома аргументами u(s) і d(s):

 (17)

Одним із варіантів компенсації члена Pd(s) d(s) є використання принципу прямого зв'язку (розімкнутого управління), як показано на рис. 17. Тут F d(s) – передатна функція регулятора із прямим зв'язком. Рівняння отриманої системи можна записати безпосередньо на рис. 17 з урахуванням (17):

 (18)

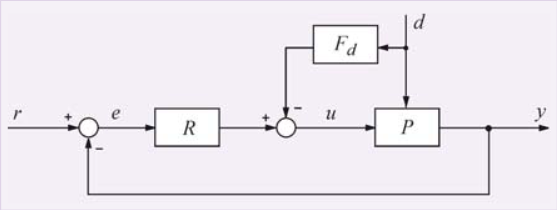


Рис. 17. Принцип компенсації впливів, що обурюють, за допомогою прямого зв'язку

Звідси випливає, що зменшити вплив зовнішніх обурень можна двома способами: збільшуючи петлеве посилення контуру із зворотним зв'язком PR або вибравши Pd – PFd = 0, тобто

 (19)

Звертання динамічного оператора пов'язане з проблемами, описаними в розділі «Знаходження зворотної динаміки об'єкта». У ряді практичних випадків досить вважати, що оператор P–1 (s) = Pd –1(0) статичний, і це значно спрощує його перебування.

В окремому випадку, коли точка додатка обурення збігається з входом об'єкта (як на рис. 3), вираз (17) спрощується до y(s) = P(s) ( u(s) +d(s)), і з ( 19) отримаємо Fd(s) = 1. Метод прямого зв'язку дозволяє компенсувати обурення до того, як воно пройде через об'єкт. Це значно збільшує загальне швидко дію системи та виключає її потенційну нестійкість.

Приклад застосування описаного методу є компенсація впливу погодних умов на промислову теплицю. Для компенсації впливу температури зовнішнього повітря, швидкості вітру, опадів необхідно встановити зовні теплиці відповідні датчики та виконати ідентифікацію передавальної функції від кожного датчика до точки вимірювання температури всередині теплиці, потім знайти зворотний оператор (19) та включити його до структури регулятора.

Правильно налаштований контролер із прямим і зворотним зв'язком дозволяє послабити вплив навантаження на керовану змінну до 10 разів (www.protuner.com, Application manual).

Недоліками методу є неможливість досить точної ідентифікації обурення та точки його застосування до об'єкта, оскільки точки розподілені у просторі, а також наявність проблеми, пов'язаної із знаходженням зворотного оператора.

## 2.4 Коректна реалізація різницевої схеми ПІД регулятора

ПІД-регулятор є найпростішим регулятором, що має ефективні апаратні аналогові реалізації і тому застосовується найбільш широко. Для своєї роботи вимагає налаштування 3-х коефіцієнтів під конкретний об'єкт, що дозволяють підібрати процес регулювання відповідно до вимог. Маючи простий фізичний зміст і простий математичний запис, застосовується широко і часто в регуляторах температури, регуляторах витрати газу та інших системах, де потрібно підтримувати певний параметр на заданому рівні, з можливими переходами між різними заданими рівнями. Зрозуміло, існують складніші регулятори, що дозволяють більш точно і швидко і з меншими перерегулюваннями виходити на задані параметри, а також враховують нелінійність або гістерезис регульованого об'єкта,

Незважаючи на свою простоту як фізичного сенсу, так і математичного запису:

https://habrastorage.org/r/w1560/getpro/habr/post_images/74c/305/0cb/74c3050cb0806a85cae4d1d8ab9b9213.png

при програмній реалізації ПІД регулятора дуже часто припускаються помилок, які зустрічаються навіть у повірених приладах автоматики. Причому перевірити якість реалізації ПІД регулятора дуже легко.

Розглянемо найпростіший приклад: терморегулятор. Для перевірки його якості найкраще підходить швидкий, малоінерційний, малопотужний об'єкт. Класика жанру: звичайна лампочка на 100Вт із прикрученою до неї тонкою термопарою (ХА). І перше, на чому слід перевіряти ПІД регулятор — деградація ПІД просто до П-регулятора. Тобто коефіцієнти інтегральної та диференціальної ставимо у нуль, а пропорційну ставимо у максимум.

Включаємо регулятор, перевіряємо: поточна температура 22 градуси, уставка 16 градусів. Лампочка не горить. Починаємо починаємо збільшувати уставку: 16.1, 16.3, 16.7, 18… 19… лампочка спалахнула. Як? Звідки? Зупиняємось – вимкнулася. Отже, ми зустрілися із першою класичною помилкою реалізації ПІД регулятора.

Невеликий математичний відступ: згадаємо ще раз інтегральний запис, вказаний вище. Ми реалізуємо її програмно, а отже дискретно. Тобто із завидною регулярністю робимо вимір вхідної величини, порівнюємо її зі уставкою, обчислюємо вплив, видаємо, повторити. А значить, треба з інтегральної форми перейти в звичайно-різницю схему. При переході зазвичай використовується перехід "в лоб":

https://habrastorage.org/r/w1560/getpro/habr/post_images/f1a/c54/1d9/f1ac541d9784004cb71ab3fbe22ad0eb.png

де E(n) = X(n) - X0(n) - тобто величина неузгодженості між поточним та заданим значенням регульованого параметра.

Використання прямої формули вимагатиме по-перше вважати і зберігати інтеграл розбіжностей за великий період, по-друге вимагає роботи з плаваючою точкою високої точності (оскільки інтегральний коефіцієнт Ki завжди < 1), або операції поділу (представляючи коефіцієнт у формі 1/Ki) великий розрядності. Все це вимагає обчислювальних ресурсів, яких в embedded зазвичай сильно обмежено ... Тому, замість реалізації прямої схеми, реалізують рекурентну формулу:

https://habrastorage.org/r/w1560/getpro/habr/post_images/ce6/eb7/243/ce6eb7243ca714979948d9aa56eb3056.png

використання рекурентної формули дозволяє скоротити обсяг обчислень та розрядність проміжних значень.

Отже, повернемося до нашого регулювальника. Отже, є об'єкт, що регулюється: лампочка. Для управління потужністю, що подається на неї, використовують простий підхід: мережа живлення (220В 50Гц) подається через симистор на навантаження. Симистор в момент переходу напівхвилі через нуль вимикається, і залишається вимкнений до тих пір, поки не буде подано сигнал на електрод, що управляє. Таким чином, чим раніше після початку напівхвилі ми подамо керуючий сигнал, тим більше енергії від цієї напівхвилі досягне керованого об'єкта. Правильно розрахувавши час для лінійності площі частини напівхвилі з часу X до кінця напівхвилі, ми отримуємо можливість видавати потужність від 0 до 100% з точністю, з якою розрахували таблицю лінеаризації.

Отже ми можемо видавати потужність від 0 до 100%. У реальних об'єктах часто не можна видавати 100% потужності — наприклад, це рясно перегоранням нагрівального елемента. Тому, всі прилади мають налаштування мінімальної і максимальної потужності, що видається на об'єкт.

Отже, після обчислення U(n) за формулою вище, додається ще обмеження результату:

if Un < Umin then Un := Umin; якщо Un>Umax then Un := Umax;

Після цього, обчислене Un і є необхідна вихідна потужність на даний момент. Та-дам! Саме ця реалізація і створює помилку, про яку написано вище.

Причина банальна: у момент переходу від дикретної до звичайно-різностної схеми, ми «виносимо за дужки» операцію обчислення інтеграла, і на кожному кроці додаємо похідну до накопиченої суми U(n-1). Наклавши на неї обмеження, ми весь обчислений інтеграл фактично обнулюємо. (Ну не скільки обнулюємо, скільки приводимо до діапазону 0-100, що в цьому випадку несуттєво). Таким чином, ми диференціюємо ПІД регулятор, і залишається диференціально-прискорювальний. Що в реальності виглядає як просто диференціальний регулятор - потужність при цьому подається пропорційно до зміни уставки або регульованої величини, а не пропорційно різниці між уставкою та регульованою величиною.

**Висновок №1**: обчислення U(n) не можна обмежувати. Для обмеження потужності подається на вихідний пристрій слід заводити окрему змінну.

Тепер, коли ми завели Urn, для обмеженої потужності перезаливаємо, продовжуємо тестувати.

Включаємо регулятор, перевіряємо: поточна температура 22 градуси, уставка 16 градусів. Лампочка не горить.

Починаємо додавати уставку: 16.1, 16.4, 17, 18, 20, 22, 24 (опа! засвітилося! ура!), 28, 30, 40, 60… Краса! Працює!

Спостерігаємо за процесом - сталося приблизно на 60, бовтається трохи туди-сюди, але тримає. Начебто все красиво. Видихаємо, перевіряємо управління з ПК: задаємо 600 градусів. І… Лампочка вимикається. Як так? Уставка 600, що тече 60, а лампочка не горить?

Поки чекаємо і повільно усвідомимо, що ми явно напоролися на якийсь «Класичний Косяк №2» лампочка повільно розгорається, виходить на 100% потужності, і так і залишається — 600 градусів вона видати ну ніяк не може.

Повертаємось знову до нашої різницевої схеми. U(n) = U(n-1) + Kp \* (dE + ...). До поточного розрахункового значення впливу додається різниця нев'язки помножена на коефіцієнт пропорційності. У нас була уставка 60, температура 60, тобто нульова нев'язка. Вихідна потужність так само була нульовою. І тут разом, стрибком, уставку збільшили до 600 градусів. нев'язка різко стала 540 градусів, помножили ще й коефіцієнт пропорційності… і вилетіли за розрядність зберігання U(n). Не смійтеся використання математики з фіксованої точки, замість плаваючої точки. При різниці в 540 градусів і роботі через 1/16, при коефіцієнті пропорційності 20, отримуємо ... 540 \* 20 \* 16 = 172800, а якщо у нас 16-ти розрядний U (n), та ще й знаковий, то фактично, в результаті обчислення ми отримали A300h = −8960. Опачки. Замість великого плюсу — такий мінус.

**Висновок №2**: обчислення мають проводитися з коректною підтримкою переповнення. Переповнилося? Обмежити граничним числом, ніяк не завертати.

Отже, наростили розрядність U(n), перетранслювали, зашили, запускаємо. Лампочка ще не зовсім охолола, там 80 градусів, уставка все ті ж 600. Лампочка запалюється і тухне. Запалюється й тухне. Як так? Уставка 600, лампочка 80 і підтримує цілком собі свої 80! Як так то?! Очевидно, у нас виліз Жучок №3.

І знову лірично-математичний відступ. Отже, є наша різницева схема: U(n) = G(U(n-1), dE(n)). Ще раз: нове значення впливу є сума минулого впливу і деякого впливу, що залежить від різниці нев'язки зараз і попередній. А що таке попередній момент? А який попередній момент у попереднього? Ну, згадали школу. Доказ щодо індукції. Якщо можна побудувати доказ для K+1, вважаючи, що доказ для K вірний, І довести окремо що вірно для K=0, тоді доказ істинний. Отже, як ми вважаємо U(0)?

Рішення, що часто зустрічається: все обнуляємо, уставку зчитуємо з флешпам'яї уставки, чекаємо 1 цикл опитування, і зчитуємо X(0). Ось нуль готовий, тепер працюємо. І… І не правильно. Чому? Тому що рекуррентна формула відштовхується від змін у нев'язці. А проініціалізувавши нулем і завантаживши поточні значення, ми втратили стартові умови. Все – замість підтримки абсолютного значення температури на рівні, що дорівнює абсолютній уставці, регулятор починає тримати температуру рівну стартовій плюс різниці уставки. Тобто було 80 градусів та уставка 200, включили прилад – він тримає 80. Змінили уставку на 240 – він почав тримати 120.

Правильна ініціалізація різницевої схеми: обнулити \_взагалі все\_. Тобто

X(0) = 0, X0(0) = 0. U(0) = 0. E(0)=X(0)-X0(0)=0.

І на першому ж циклі обчислень у нас як би стрибком з'являються уставці та поточне значення:

X (1) = 80. X0 (1) = 200. U(1) = U(0)+Kp\*(E(1)-E(0)) = U(0)+Kp\*(X(1)-X0(1)-E(0)) = 0 + 20 \* (200 - 80 - 0) = 2400

Ось тепер схема працює правильно.

**Висновок №3**: коректно ініціалізуй стартові умови

Чи правильно? Ну, ну... Ще раз... Ставимо уставку 20. Чекаємо на охолодження... Вимикаємо. Вмикаємо. Отже, краса: поточна 20, уставка 20. Ставимо стрибком 600. Поїхало грітися. 100, 120 ... ставимо уставку 20. Відключилося, пішло охолоджуватися. Чекаємо трохи (120 ... 110 ... 100 ... 90 ... 80 ...) і ставимо уставку 100. Поїхало грітися ... 105 градусів, відключилося. Стоп. А чому воно тримає 105? У нас же зараз працює лише пропорційна складова. При правильній реалізації з фізичного сенсу процесу коливальний процес не може тримати уставку вище, ніж задано. Суворо нижче. А тримає на 5 градусів більше, ніж попросили. Це спостерігається Прикол №4.

Отже, згадуємо, що у нас було вище: Висновок №2: U(n) не можна обмежувати. І Висновок №3: при переповненні обмежити все-таки доведеться. Так Так. Інакше робоча точка зміщується на обмежений момент. Що ж робити? Збільшити розрядність? Добре, якщо вистачає обчислювальної потужності. А чи треба? Власне, що поганого, що ми U(n) = 9999.99, а чи не 29999.99? Загалом тільки те, що ми втратили 20000. Але зараз для роботи нам так і так треба ввалювати просто 100% потужності, правильно? Правильно. Отже, проблеми з обмеженням у полиці немає, доки ми не відходимо від межі. Таким чином, у разі переповнення треба ставити прапор, і після досягнення, наприклад, половини діапазону (тобто як U(n) після 9999.9 опустилося нижче 5000.00), наново реініціалізувати схему. Тобто відкидати історію, сказати що n=0 і див. Висновок №3. Допитливий розум вже зрозумів, що у разі повної схеми, коли всі три компоненти не дорівнюють нулю, обнулюючи в процесі ітеративний процес, ми в тому числі обнулюємо накопичений інтеграл інтегральної складової. Однак, у зв'язку з тим, що обнулюємо ми значно заздалегідь, він встигне підкопитися за час до формування залишку. Та й не зовсім коректно накопичувати інтеграл на «великих» перегонах, бо мета інтегральної складової – «вибрати» нев'язку, яку не може відпрацювати пропорційна складова окремо.

**Висновок №4**: якщо з якоїсь причини було обмежено U(n), схему слід переініціалізувати як тільки склалося враження, що схема повернулася в нормальний стан.

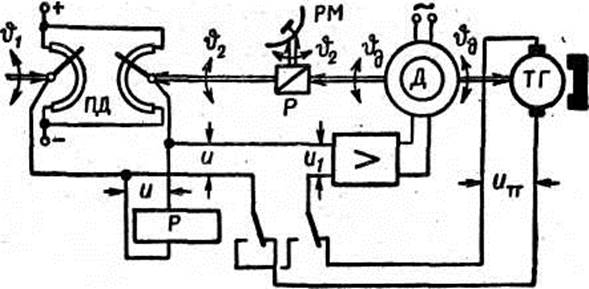
# 3 Адаптивні системи управління

## 3.1 Системи зі змінною структурою

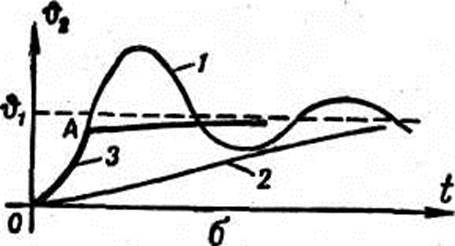
Зазвичай у аналізованих автоматичних системах сукупність елементів, які входять у керуючий пристрій, і зв'язків між ними залишаються незмінними. Однак такий спосіб управління не є єдиним та найкращим. У багатьох випадках ефективність управління істотно підвищується, якщо керуючий пристрій має можливість розривати і відновлювати зв'язки між функціональними елементами, що входять до нього, і тим самим змінювати структуру автоматичної системи в цілому.

Системи управління у яких зв'язки між функціональними елементами змінюються залежно стану системи, називаються системами зі змінною структурою (УПС).

Розглянемо конкретний приклад.



На малюнку вище зображена схема стежить системи зі змінним демпфуванням, побудована за принципом УПС. Реле Р, що входить до складу її керуючого пристрою, в залежності від величини напруги u на його обмотці може або розривати, або відновлювати зв'язок між виходом тахогенератора ТГ і входом підсилювача системи. Таким чином, слідкувальна система може мати дві лінійні структури: з відключеним та включеним тахогенератором.



Система з вимкненим тахогенератором (u1=u) слабо демпфована. Перехідний процес у ній має коливальний характер, але з його початковому ділянці керована величинамає велике збільшення (крива 1).

Система з увімкненим тахогенератором (u1=u-uТГ) сильно демпфована. Перехідний процес у ній немає перерегулювання, але збільшення керованої величини мало (крива 2).

Бажаний характер перехідного процесу виходить за рахунок поєднання корисних властивостей кожної із структур: велике приріст на початковій ділянці процесу, характерне для першої структури, та відсутність перерегулювання, властиве другій структурі.

Процес управління у системі протікає так. При великих неузгодженостях u = u1 – u2 реле відключає тахогенератор, на вхід підсилювача подається повна напруга датчика кута неузгодженості (u1 = u) і величина u2 швидко прагне значення u1 (крива 3). Коли неузгодженість стає малим, тахогенератор підключається до входу підсилювача так, що напруга uТГ віднімається від напруги u. В результаті виключається перерегулювання.

Варто окремо відзначити, що УПС — це системи із структурою регулятора, що змінюється. У ході еволюції адаптивних систем управління з'явилися системи із змінною структурою об'єкта управління, наприклад, системи автоматичного сортування листів у великих поштових центрах тощо. Але термін УПС до них не належить.

## 3.2 Екстремальні системи керування

*екстремальними*називають системи, які у процесі роботи автоматично відшукують такі управляючі на вході керованого об'єкта, у яких забезпечується безперервне підтримання показника якості системи поблизу його екстремального значення.

Основна особливість цих систем у тому, що управління них здійснюється не шляхом компенсації приватних неузгоджень (відхилень) у системі, а цілеспрямованим впливом на систему з урахуванням деякої функції її стану — показника якості системи управління.

Системи екстремального управління складніше звичайних САУ, та його доцільно застосовувати, якщо ОУ має досить ясно виражений екстремум показника якості, який залежить від управляючих впливів і зовнішніх умов.

На рис. 13.I проілюстрована залежність показника якості / від керуючого впливу та для різних варіантів зовнішніх умов.

Якщо при зміні зовнішніх умов змінюється тільки величина екстремального значення показника якості /е, а координата if, що відповідає екстремуму, не змінюється або змінюється тільки координата иЭУ відповідна екстремуму /е, а величина /е не змінюється, то можна забезпечити екстремальне значення показника якості, використовуючи стандартну САУ, тобто. САУ із постійними параметрами керуючого пристрою.

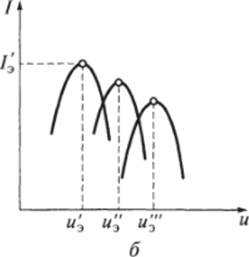
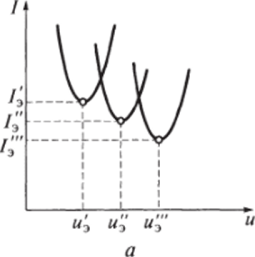
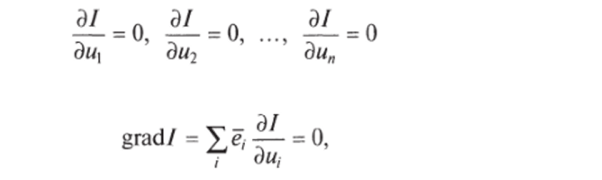
Показником якості у СЕУ можуть бути різні величини. Наприклад, для системи управління металорізальним верстатом таким показником може бути продуктивність, яка визначається масою стружки, що знімається різцем в одиницю часу, за умови, що температура різця та деталі не перевищить заданого рівня.

Рис. 13.1. Залежність показника якості I від керуючого впливу та для різних варіантів зовнішніх умов: а - з мінімальними екстремумами; б - з максимальними екстремумами

Зазвичай показник якості СЕУ є функцією кількох управляючих впливів, тобто.

https://studme.org/htm/img/39/2606/627.png

Якщо остання функція диференційована і шуканий оптимум явно не лежить на кордоні, то в точці, що відповідає екстремуму,



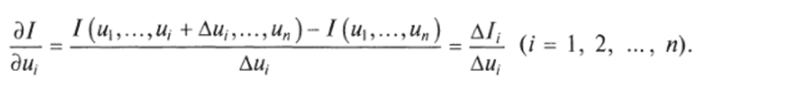
або

де е|, ..., еп — поодинокі вектори осей, якими відраховуються величини иь ..., ип.

Екстремальна система повинна забезпечити рух робочої точки по поверхні / у просторі змінних ух, уъ ..., уп до точки, в якій grad/= 0. Для здійснення цього необхідно, по-перше, визначити градієнт і, по-друге, організувати рух до точки екстремуму.

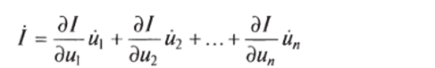
Для вирішення першої та другої задач існує багато способів. Розглянемо лише найпростіші з них.

**Методи визначення градієнта показника якості.**На заміні приватних похідних відносин кінцевих прирощень заснований метод кінцевих прирощень:



Даючи величинам w,, ..., w„ по черзі невеликі стрибкоподібні зміни, обчислюють відповідні збільшення Aw,. Точність цього залежить від характеру функції / і величини приростів Aw,.

*Метод похідної за часом*заснований на послідовній повільній зміні кожної з величин і, з постійною швидкістю спочатку в одну, а потім в іншу сторону при постійних значеннях інших величин wy. З формули для повної похідної функції / за часом



слід, якщо всі величини wy постійні, а координата w, змінюється з постійною швидкістю w,, то

https://studme.org/htm/img/39/2606/633.png

У цьому методі w, = const.

Недоліком обох розглянутих методів визначення градієнта функції є те, що вони дозволяють визначати похідні 9//3w, тільки по черзі, внаслідок чого час визначення градієнта функції залежить від числа змінюваних координат.

*Метод синхронного детектування*дозволяє одночасно знайти всі складові градієнта. При визначенні приватних похідних 3//3w, цим методом (рис. 13.2) вхідні змінні об'єкта модулюються невеликими по амплітуді гармонічними коливаннями різних частот, тобто. до сигналів w,, ..., іп додаються синусоїдальні сигнали

https://studme.org/htm/img/39/2606/634.png

різних частот з,, ..., з„.

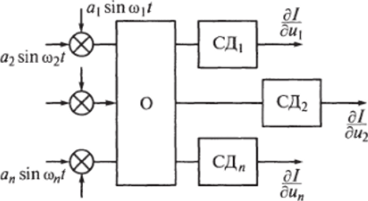
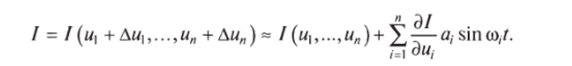


Рис. 13.2. Визначення приватних похідних методом синхронного детектування

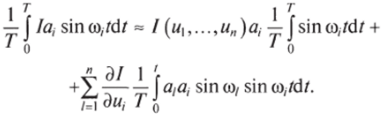
Для визначення впливу коливань кожної змінної на критерій якості використовуються синхронні детектори ЦД. Кожен детектор виконує множення величини / відповідний гармонічний сигнал tf/Sinco,/ і усереднення отриманого твору за часом. При цьому вихідні сигнали синхронних детекторів приблизно пропорційні приватним похідним функції / відповідним змінним. Дійсно, при малих амплітудах д, а2, ап функцію / можна

розкласти в ряд Тейлора по приросту Дм,.

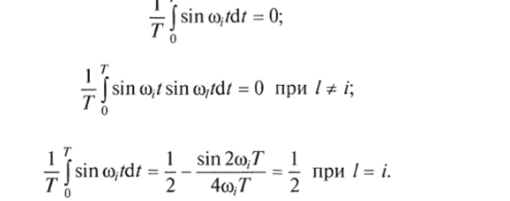
Якщо обмежитися членами першого ступеня щодо Аі„ то



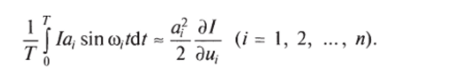
Помножуючи цю рівність на Д/Sinco,/ і середня результат за часом, отримаємо



Якщо взяти час опосередкування Т кратним всім періодам 2я/(0|, 2л/(о„, то



При будь-якому досить великому часі осреднения Т перші два інтеграли будуть як завгодно малі, а третій як завгодно близький до 1/2. Отже, формула може бути переписана як



Як пошукові коливання Пекло, системи можна використовувати не тільки синусоїдальні коливання, але і будь-які типи коливань, що задовольняють єдиній умові, щоб середні значення творів коливань різних параметрів за час опосередкування були досить близькі до нуля.

Пошукові алгоритми, які у екстремальних системах. Коли знайдено градієнт функції /, необхідно організувати рух системи до точки екстремуму, тобто. пошук екстремуму. В даний час є велика кількість методів вирішення цього завдання. Найбільш поширені такі методи: Гаусса-Зейделя, градієнта і якнайшвидшого спуску.

*Метод Гауса-Зейдія*полягає в почерговій зміні змінних ії іп та визначенні приватних екстремумів

https://studme.org/htm/img/39/2606/640.png

за умови, що решта змінних //,, ..., uk.hw\*+l, ..., іп є постійними. У цьому методі спочатку встановлюється вихідна (початкова) точка пошуку. Потім усі змінні іъ ..., іп фіксуються, а змінна їх варіюється доти, доки не буде виконана умова

https://studme.org/htm/img/39/2606/641.png

У цій точці величина їх = і0 фіксується, і починає змінюватись змінна і2 до звернення в нуль похідної d//dі і т.д. Після знаходження точки, в якій d//dw„ = 0, знову починає змінюватися uh і весь цикл повторюється доти, доки не буде знайдена точка екстремуму.

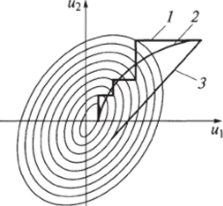


Рис. 13.3. Переріз функції двох змінних 1(ІІІ2): 1—3 — траєкторії

На рис. 13.3 показані перерізи функції двох змінних /(іь і2). Шляхи руху до точки екстремуму за методом Гауса — Зейделя, або, як його іноді називають, методом послідовної зміни змінних відповідає ламана лінія (траєкторія I на рис. 13.3), що складається з взаємно-перпендикулярних прямих ділянок.

*Метод градієнта*передбачає одночасну зміну всіх змінних, так що точка, що зображає, рухається в напрямку, близькому до напрямку вектора градієнта. Рух може здійснюватися безперервно чи дискретно.

На рис. 13.3 траєкторія руху методом градієнта зображена траєкторією 2 Ця крива нормальна до поверхні рівних значень /(и,, и2) = = const.

*Метод якнайшвидшого спуску*є узагальненням методів Гауса-Зейделя та методу градієнта. У ньому рух відбувається у напрямку вектора градієнта до того часу, поки нс звернеться на нуль приватна похідна функції / у цьому напрямі. Потім знову визначається градієнт і рух відбувається вздовж нового вектора градієнта до звернення в нуль похідної цього напрямку до тих пір, поки не буде досягнута точка екстремуму. На рис. 13.3 траєкторія, що відповідає методу якнайшвидшого спуску, зазначена цифрою 3.

Градієнтний метод і метод якнайшвидшого спуску забезпечують якнайшвидше досягнення точки екстремуму. При великих відхиленнях зображуючої точки від екстремуму доцільно використовувати метод якнайшвидшого спуску, а за малих - метод градієнта.

Класифікація СЕУ. В даний час є велика кількість видів екстремальних систем, і класифікацію їх можна проводити за різними ознаками.

За джерелом коливань, що використовуються для обчислення градієнта функції / (за способом пошуку), СЕУ поділяються на системи з авто коливальним та примусовим пошуком екстремуму.

Якщо пошукові коливання генеруються самою системою, СЕУ називають екстремальною системою з авто коливальним пошуком екстремуму. Якщо пошукові коливання вводяться від джерела, спеціально передбаченого цієї мети, то СЕУ називають екстремальною системою з примусовим пошуком екстремуму.

За кількістю змінних, яких залежить функція якості, СЕУ поділяються на одномірні і многомерные. Можлива класифікація СЕУ за методом визначення градієнта, методом руху до екстремуму, а також за іншими ознаками.

Розглянемо як приклад принцип роботи СЕУ з автоколивальним пошуком екстремуму/е, схема якої показана на рис. 13.4 а.

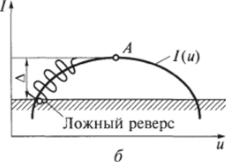
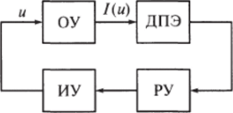


Рис. 13.4. Принцип роботи СЕУ з авто коливальним пошуком екстремуму: а - схема СЕУ; б - «хибні реверси» через широкосмугові перешкоди

На датчик показника екстремуму ДПЕ подається виміряне значення показника якості, який безперервно змінюється (рухається) у напрямку екстремуму за допомогою монотонної зміни вхідної змінної та об'єкта управління ОУ. При досягненні показником якості / екстремального значення Е//Еі = 0 датчик фіксує (запам'ятовує) значення е, що відповідає екстремуму показника якості, але ніяких сигналів, що управляють, не видає. Величина і навіть продовжує змінюватися у той самий бік (для певності нехай і збільшується). Коли різниця між поточним значенням / і величиною /е стає рівною деякою заздалегідь заданою величиною (порогом) А (Л = |/- /е|), датчик видає імпульс, що спрацьовує реверсуючого пристрою РУ. При цьому виконавчий пристрій ІУ змінює напрямок зміни величини і (і почне зменшуватися), і величина 1(і) знову прагне екстремального значення. У СЕУ такого виду виникає автоколивальний процес біля положення, що відповідає екстремуму/.

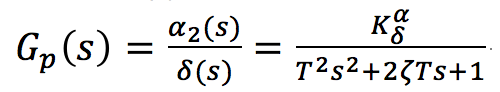
Розглянута СЕУ є найпростішою. Недоліком є ​​можливість «хибних реверсів» через широкосмугових перешкод (рис. 13.4, б). Через перешкод реверс може відбуватися не близько точки, що відповідає екстремуму показника якості (точка А), а в будь-якій іншій точці, де градієнт вимірюваного значення показника якості перетворюється на нуль. Показник якості у цій системі відхилений від екстремуму. Рівень перешкод визначає розмір порога Л.

# 4 Нечіткі регулятори

## 4.1 Постановка задачі

Розглянемо синтез цифрового ПІД-регулятора та нечіткого регулятора для системи управління ракетою по кутку атаки. Методом математичного моделювання визначимо процеси у системі та дамо порівняльну оцінку якості системи при використанні синтезованих регуляторів.

Прийнявши за вихідну координату ракети кут атаки:https://habrastorage.org/r/w1560/webt/0f/mj/ch/0fmjchqikcazeias1tse8u7c8os.png, а за вхідну координату кут повороту кермаhttps://habrastorage.org/r/w1560/webt/d2/-s/b5/d2-sb559am7mjtf7yirvtyqlccc.pngвизначимо передатну функцію ракети у вигляді:

,

де:

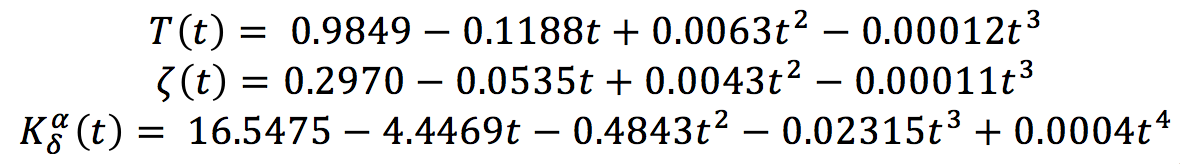
https://habrastorage.org/r/w1560/webt/vt/xj/jm/vtxjjmsll6wmsapumoegkejidiu.png- Коефіцієнт перетворення ракети,

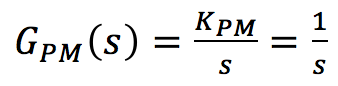
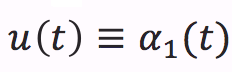
https://habrastorage.org/r/w1560/webt/pz/kw/tu/pzkwtuzlgjpzotjziqctz8gechg.png- Коефіцієнт демпфування,

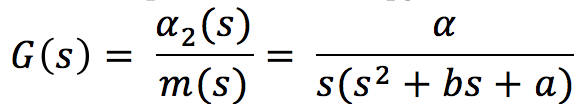
https://habrastorage.org/r/w1560/webt/of/2p/zg/of2pzgmhnca7maoqcqt_xoz1a_i.png- Постійна часу.

Тут і далі «передавальна функція» використовується не в класичному класичному визначенні, як відношення перетворень лапласа.

При дослідженні системи управління припустимо, що залежність параметрів ракети від часу польоту визначаються так:

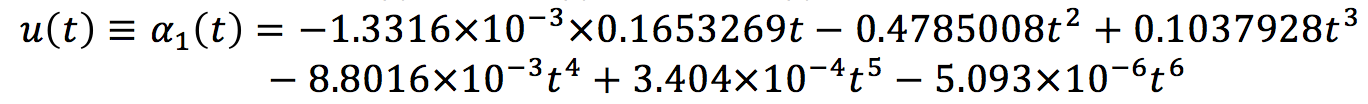


Для спрощення розрахунків, кермовий механізм опишемо передавальною функцією інтегруючої ланкиВ цьому випадку вхід системи- Заданий кут атаки, вихід системиhttps://habrastorage.org/r/w1560/webt/0f/mj/ch/0fmjchqikcazeias1tse8u7c8os.png- відпрацьований ракетою кут атаки, m(t) - керуючий сигнал на виході регулятора, а об'єкт управління описується загальною функцією передачі:



(В об'єкт управління включені аналогові кермовий механізм та ракета).

Закон зміни вхідного впливу заданий поліномом:



Необхідно розробити регулятор, який забезпечує відпрацювання вхідного впливу за допомогою ПІД-регулятора та регулятора на базі нечіткої логіки.

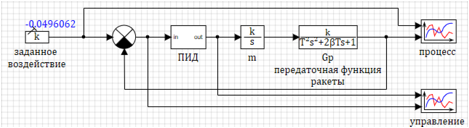
Здійснити вибір коефіцієнтів регуляторів.

Зробити порівняння перехідного процесу з ПІД-регулятором і з регулятором на основі нечіткої логіки.

## 4.2 Динамічна модель об'єкту

Створимо динамічну модель серед структурного моделювання. Сама схема моделі представлена ​​малюнку 1.

Заданий вплив задається у вигляді блоку константа, як параметри визначається змінна зі скрипта. Параметри перехідної функції задаються як змінних.



1. Схема динамічної моделі ракети.

## 4.3 Налаштування регулятора

Блок ПІД є субмодель (рис. 2), в якому використовується стандартний блок «Дискретний ПІД-регулятор». Частота дискретизації обрана рівною 0.001 с.

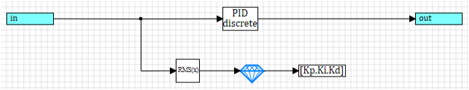


Рисунок 2. ПІД регулятор зі схемою налаштування.

Параметри регулятора задаються як імена глобальних сигналів проекту Kp, Ki, Kd. Це дозволяє змінювати параметри під час моделювання та налаштувати регулятор.

Для налаштування регулятора використовувався блок Оптимізація, критерієм оптимізації є мінімум середнього квадратичного відхилення.

Блок оптимізації здійснює оптимізацію по всьому перехідному процесу. Результат оптимізації – вектор із трьох коефіцієнтів, який спрямовується до блоку «Запис до списку сигналів», де обчислені значення передаються в сигнали і, відповідно, змінюються значення коефіцієнтів ПІД. Для налаштування регулятора ми визначаємо наступні параметри оптимізації:

Вихідні значення всіх коефіцієнтів 1.

Діапазон для підбору заданий від -50 до +50

Точність підбору 0.001

Максимальне середньоквадратичне відхилення після оптимізації 0.01

У цьому випадку блок оптимізації розрахував наступні оптимальні значення коефіцієнтів:

**Kp = -1,7498597; Ki = 17.891995; Kd = 11.606602**.

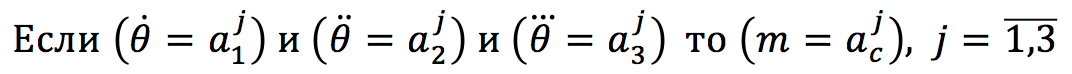
За таких коефіцієнтів середньоквадратичне відхилення у заданому перехідному процесі склало 0.008738090

|  |  |
| --- | --- |
| https://habrastorage.org/r/w1560/webt/ab/c7/ht/abc7htr6emfjjaigwy9i_mbagbg.png 3. Перехідний процес. | https://habrastorage.org/r/w1560/webt/ae/r1/jp/aer1jp59s1gpppaj6t0sxr9gu1k.png Рисунок 4. Управління. |

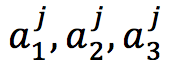
## 4.4 Регулятор на основі нечіткої логіки

Основне переваги регулятора з урахуванням нечіткої логіки – це простота і наочність формування правил управління об'єктом.

Наприклад, у книзі «Нечіткі регулятори в системах автоматичного моделювання» правила нечіткого регулювання для управління ракети по кутку атаки описані у вигляді математичного виразу:

де,https://habrastorage.org/r/w1560/webt/0f/n_/ll/0fn_llp68_vbqeoqtskel5w8dhw.png- Помилка системи, швидкість змінюючи (перша похідна) помилки, прискорення (друга похідна) помилки;

*m*- Управляючі вплив на об'єкт;

-лінгвістичні оцінки помилки, швидкості зміни помилки (першої похідної) помилки та другої похідної помилки, що розглядаються як нечіткі множини, визначені на універсальній множиніhttps://habrastorage.org/r/w1560/webt/ef/3w/uw/ef3wuwzgk1e3vzejo-ycvue4voo.png;

https://habrastorage.org/r/w1560/webt/2q/vc/ax/2qvcaxq_ckob6pxxlw3glhtb9cg.png- лінгвістичні оцінки керуючого впливу на об'єкт, що вибираються з терм-множин змінної m

Читач може запитати: це як розуміти?

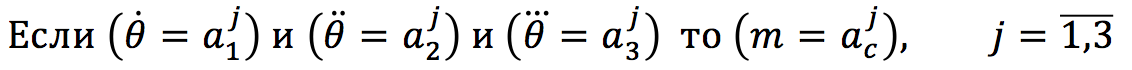
Іноді в мене закрадаються сумніви, що автори-математики самі розуміють те, що вони написали. За розумними математичними оборотами схована велика таємниця правил нечіткого регулювання. Ось вона:

**багато – зменшуй**

**норма – не чіпай**

**мало - збільшуй**

Якщо перекласти з пташиної мови математики російською, то вираз

означає буквально таке:

Якщо більше норми та відхилення зростає і швидкість зростання збільшується, то зменшуємо.

Якщо норма і не змінюється і швидкість постійна, то не змінюємо.

Якщо менше норми падає і швидкість падіння збільшується, то збільшуємо.

Якщо розуміти те, що реально ховається за математичним туманом, то можна підходити до створення регуляторів більш усвідомлено та отримати цікавіші результати.

## 4.5 Трохи теорії

Для вирішення завдання регулювання кута атаки ми повинні з безперервної величини відхилення отримати три терми – менше, норма, більше. Те саме потрібно зробити для першої похідної відхилення і другої похідної відхилення. Це перший етап нечіткого висновку – фазифікація.

Щоб отримати терми, ми повинні встановити числове значення параметра для кожного терму. Наприклад: "Мало" = -1; "Норма" = 0; "Багато" = 1. Для фазифікації будемо використовувати трикутні функції. Функції зростають у міру наближення до заданої величини і зменшуються в міру видалення. Два варіанти трикутних функцій наведено на малюнку 5:

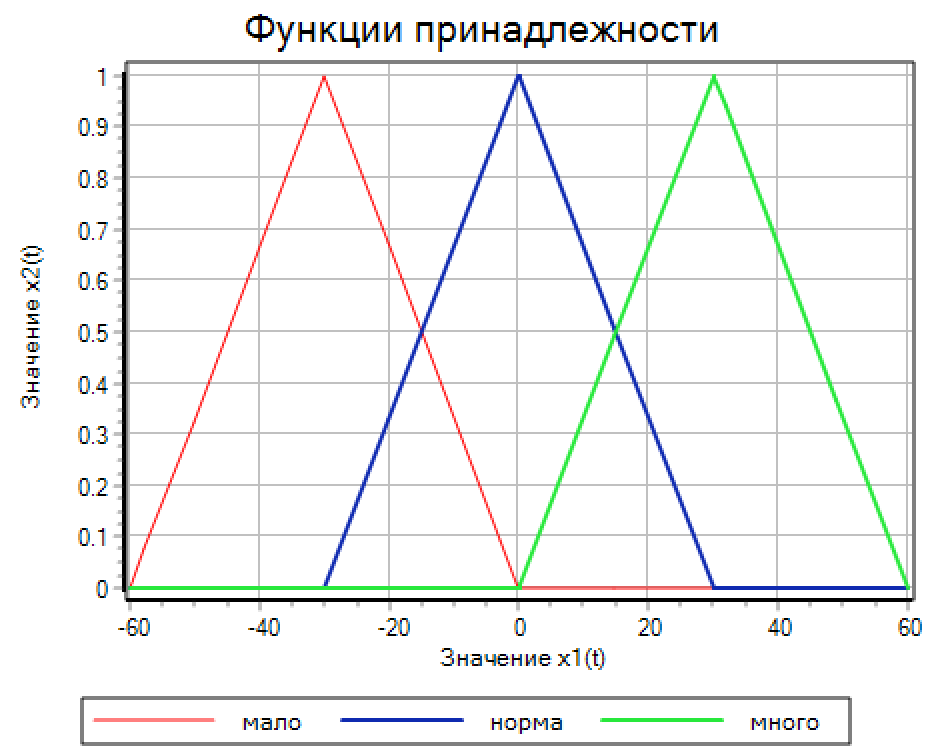
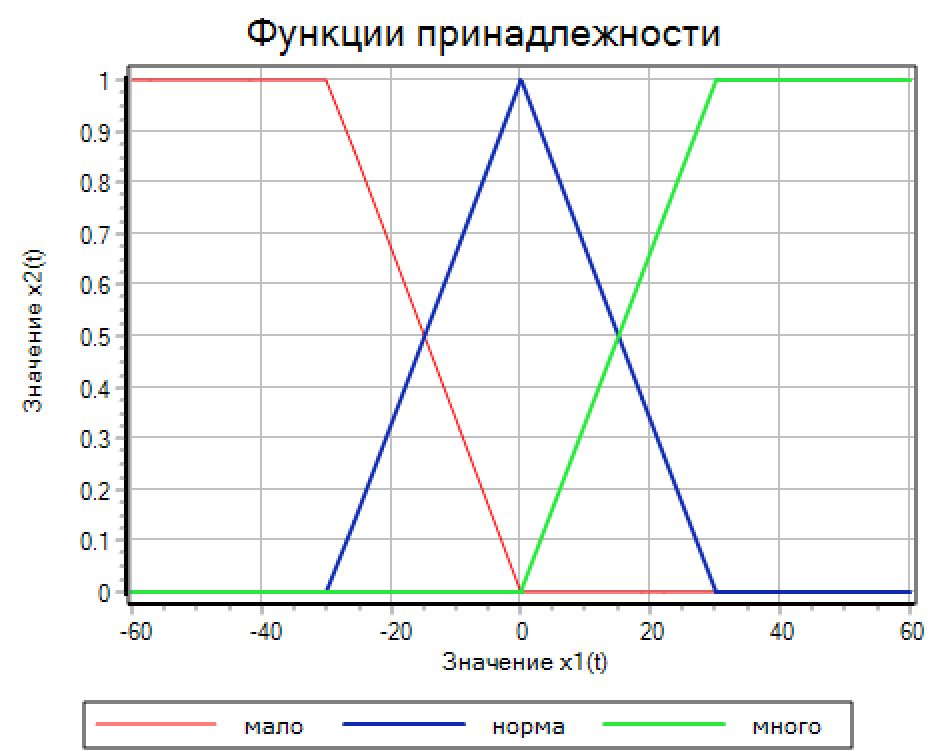


Рисунок 5. Трикутні функції приладдя.

Знаючи величину відхилення (х1), ми можемо знайти значення функції приналежності для термів більше (червона лінія), норма (зелена лінія), менше (синя лінія). Величини будуть в діапазоні від 0 до 1.

Зверніть увагу, що на лівому графіку є крайні функції, не зовсім «трикутні». Якщо розглядати з погляду абстрактної математики, то функції на правому графіку «красивіші». Але якщо згадати «головну таємницю правил нечіткого висновку», то лівий графік правильніший. Справді:

Розглянемо правило "Мало - додай", якщо в нас значення -1, то "мало" = 1 (червона лінія) вірно для обох графіків. А якщо ми маємо значення -2? За логікою ми теж маємо «додавати». На лівому графіці при -2 так і є: "мало = 1", але на правому графіку у нас "мало" = 0, що явно не вірно. Те саме справедливо для правила «Багато – зменшуй».

Фазифікація «чесними» трикутними функціями може призводити до того що при виході величини за діапазон визначення функцій ми отримуємо 0, для всіх термів, що, своєю чергою, може привести до відсутності впливу на об'єкт.

Зворотне завдання – дефазифікація. Для розрахунку впливу потрібно виконати зворотне перетворення – ми маємо значення функцій приналежності зменшувати, не змінювати, збільшувати в діапазоні (0...1) (трикутні функції) та діапазон впливів, які ми можемо зробити, і ми повинні з трьох термів отримати одне число - конкретний вплив.

Отримати можна вплив можна у різний спосіб, наприклад, по центру маси фігури. На малюнку 6 наведено стан регулятора, де значення термів зменшувати 0.3 не змінювати 0.6 і збільшувати 0.8 при діапазоні регулюючого впливу -30..30 результуючий вплив = 4.1.

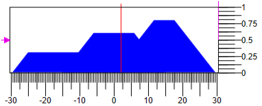
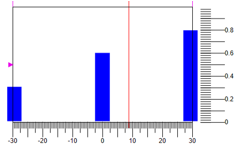


Рисунок 6. Дефазифікація керуючого впливу

Інший варіант дефазифікації – у центрі мас точок. На малюнку 7 наведено варіант, де при тих же значеннях термів і діапазону регулювання ми отримуємо інший варіант відповіді 8.82:



7. Дефазифікація методом центру маси точок.

Треба розуміти, що крім способу виведення, результат впливає також форма функції власності. Наприклад, можна вибрати такі трикутні функції, у яких основа трикутника однакова, відрізняються лише вершини. (Див. мал. 8).

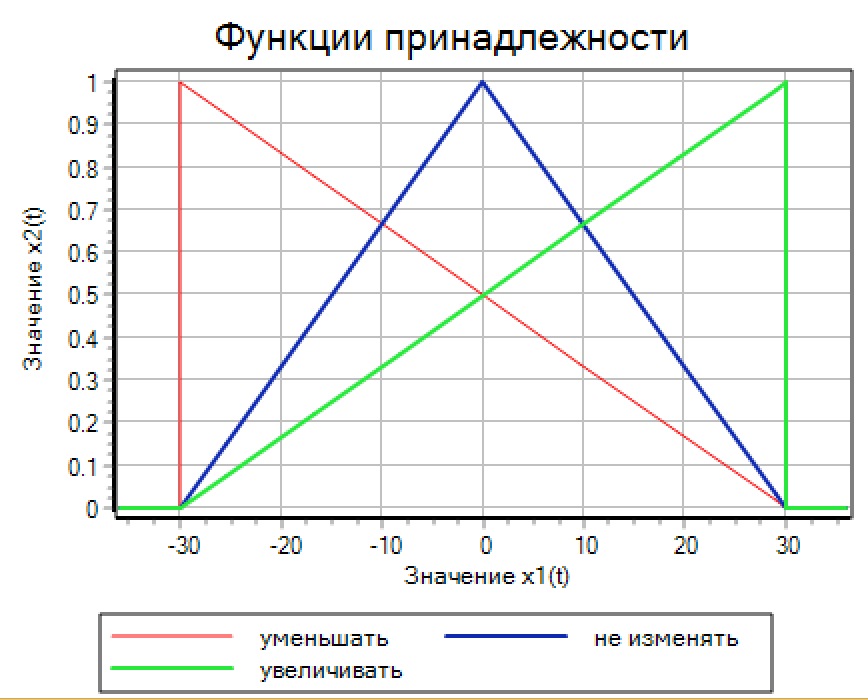


Рисунок 8. Трикутні функції з однією основою.

В цьому випадку результат фазифікації при таких же значеннях термів зменшувати 0.3, не змінювати 0.6 і збільшувати -0.8 при діапазоні регулюючої дії -30, 30 результуючий вплив = 5.27.

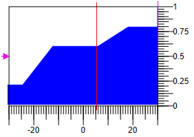


Рисунок 9. Дефазифікація шляхом розрахунку площі.

Озброївшись таємними знаннями про нечітку логіку, створимо модель регулятора. Модель ракети залишаємо таку ж, як і для ПІД-регулятора (див. рис. 1), а ось у субмоделі регулятора зберемо схему, зображену на малюнку 10.

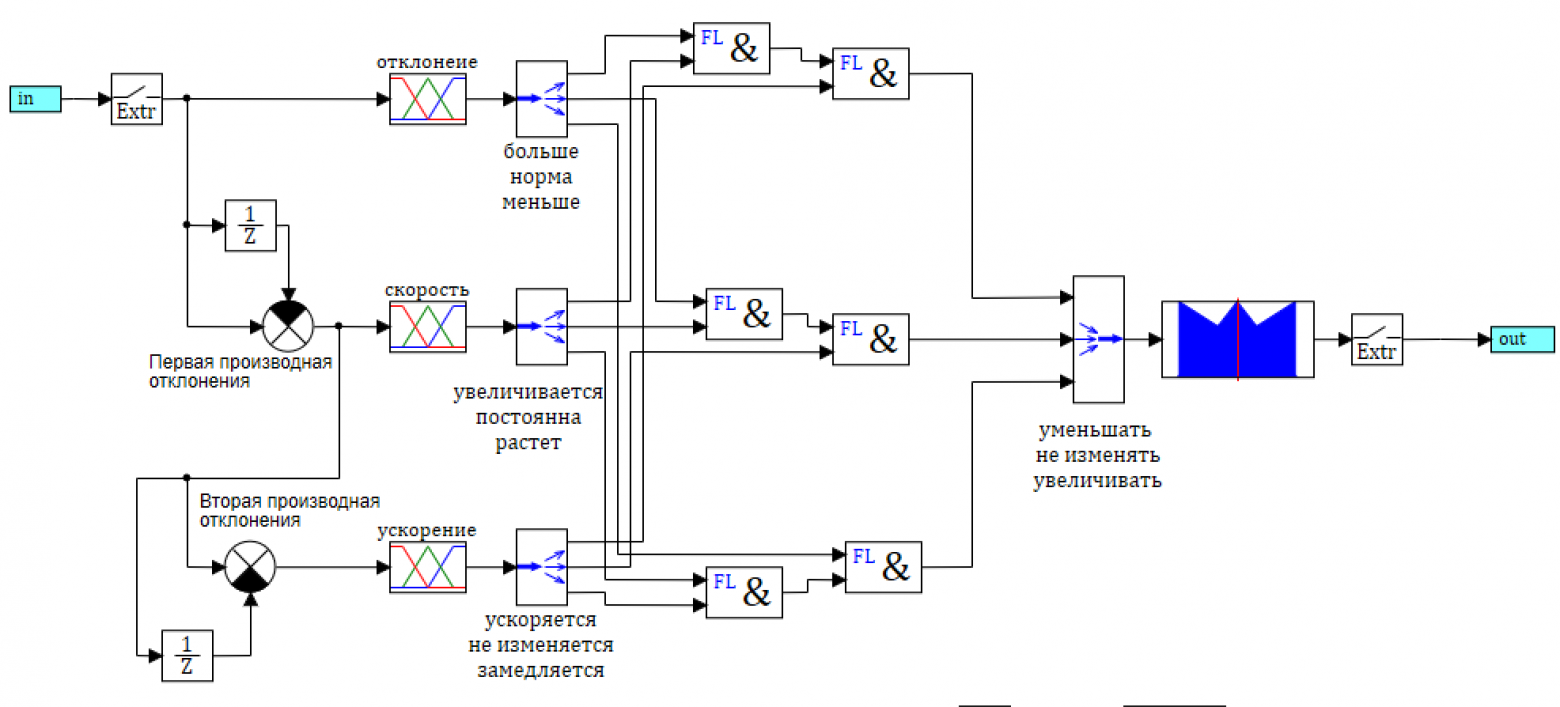


Рисунок 10. Схема регулятора з урахуванням нечіткої логіки.

На вхід у регулятор подається неузгодженість між заданим кутом атаки та реальним (виміряним). Після входу стоїть блок «Екстраполятор», який забезпечує перетворення безперервного сигналу на дискретний із заданим періодом дискретизації (0.001 с – такий самий, як у дискретного ПІД-регулятора).

Після цього відбувається обчислення першої та другої похідної відхилення. Для цього ми обчислюємо різницю між поточним значенням та значенням із затримкою на період квантування, ділимо її на час затримки (коефіцієнт у порівнювальному блоці). Таким чином, ми отримуємо три входи: помилка системи, швидкість зміни (перша похідна) помилки, прискорення (друга похідна) помилки.

Значення вхідних змінних перетворюються на блоки фазифікації трикутними функціями. Для кожної змінної отримуємо три лінгвістичні змінні (загалом дев'ять).

Блоки «Демультиплексор» розводять вектора в лінгвістичні змінні на формування правил. На схемі назви змінних підписано у порядку їх розпорядження у векторах.

Відхилення в нашому випадку – це різниця заданого та виміряного, якщо негативне значення – значить кут атаки більший за заданий, ми повинні зменшувати. І відповідно навпаки, якщо відхилення позитивне, то виміряний кут менший від заданого, ми повинні збільшувати.

**(Більше – зменшуй, менше – збільшуй, норма – не чіпай).**

Вихід має три лінгвістичні змінні «зменшувати», «не змінювати», «збільшувати». Мультиплексор збирає значення вектор і віддає в блок нечіткого виведення. Тепер, коли ми маємо всі змінні, ми можемо записати правила нечіткого виведення у вигляді схеми.

*Якщо більше норми*і відхилення зростає і швидкість зростання збільшується => зменшуємо.

Якщо норма не змінюється і постійна => не змінюємо.

Якщо менше норми і падає, і швидкість падіння збільшується => збільшуємо.

Усі лінгвістичні змінні у правилах у нас пов'язані через логічні блоки «і» та підключені до виходів. Як видно з малюнка 10, схема логічна нечіткого виводу практично не відрізняється від звичайної логічної схеми, тільки використовуються блоки нечіткої логіки.

Аналогічно налаштування ПІД-регулятора ми використовуємо блок оптимізації.

Залишається питання із параметрами блоків.

## 4.6 Синтез регулятора на основі нечіткої логіки

У житті нічого не дається задарма і тому простота правил регулювання компенсується кількістю параметрів, що описують функції власності. Насправді, якщо для ПІД-регулятора потрібно підібрати три коефіцієнти, то у разі нечіткої логіки, тільки для однієї трикутної функції потрібно 3 числа для вершин. Якщо для кожної вхідної змінної нам потрібно 3 функції приналежності + 3 для вихідного, виходить, що нам потрібно задати 3 x 3 x 3 + 3 x 3 = 36 параметрів!

Але не все так сумно. Для першого наближення та первинного налаштування можна все спростити.

Зробивши деякі припущення для початкового налаштування регулятора:

Задаємо симетричність функцій щодо нуля, тоді замість двох чисел для максимуму і мінімуму можна задати одне – Мах, і, відповідно, діапазонно буде визначено як [-Мах… Мах].

Задаємо рівномірний розподіл функцій, тоді можна розрахувати положення всіх вершин трикутників виходячи із заданого діапазону.

Для трьох функцій координати вершин визначаться як Max, 0, Max.

Задаємо, що основа трикутника всіх функцій власності однакові.

Таким чином, замість 36 незалежних параметрів ми повинні задати лише 4, максимальне відхилення від 0 для трьох вхідних змінних та одного виходу, а саме:

**uMax**- Амплітуда керуючого впливу (-uMax ... uMax);

**deltaMax**- Максимальне відхилення (-deltaMax ... deltaMax);

**divMax**- максимальне похідне відхилення (-divMax ... divMax);

**div2Max**– максимальна друга похідна відхилення (-div2Max… div2Max)

У функціях фазифікації та нечіткого виводу використовуємо ці сигнали для розрахунку параметрів з урахуванням прийнятих припущень.

Налаштування регулятора, запропоновані у книзі Гостева В.В. "Нечіткі регулятори в системах автоматичного моделювання", для випадку фазифікації трьома функціями приладдя пропонуються такі параметри:

**uMax = 30**- Амплітуда керуючого впливу;

**deltaMax = 0.01**- максимальне відхилення;

**divMax = 0.07**- максимальна похідна відхилення;

**div2Max = 1**- Максимальна друга похідна відхилення.

## 4.7 Порівняння перехідних процесів

На графіці перехідних процесів збіг заданого впливу та отриманого результату практично повний:

|  |  |
| --- | --- |
| https://habrastorage.org/r/w1560/webt/my/o6/fd/myo6fd8zpwbybqogtysuygzdvik.png Рисунок 11.а Перехідний процес. ПІД регулятор | https://habrastorage.org/r/w1560/webt/im/q6/vw/imq6vwffyfnd-m7shrkoikxfkhy.png Рисунок 11.б Перехідний процес. Нечітка логіка |

Явні відмінності можна подивитися на графіках отриманого відхилення та впливу, що управляє:

|  |  |
| --- | --- |
| https://habrastorage.org/r/w1560/webt/0l/tx/k4/0ltxk42ygfgvnkxmeki5uicnpju.png Малюнок 12.а. Управління. ПІД регулятор | https://habrastorage.org/r/w1560/webt/tb/tc/ic/tbtcic53qspbuvnb_-2vaigztfa.png Малюнок 12.б. Управління. Нечітка логіка |

З порівняння малюнків видно, що нечіткий регулятор забезпечує меншу помилку і краще відпрацьовує перехідний процес.

Порівняємо перехідні процеси в системі, якщо задати ступінчасту керуючу дію. Результати малюнку 13:

|  |  |
| --- | --- |
| https://habrastorage.org/r/w1560/webt/0e/ev/-g/0eev-ghwaketpmvy6a2z18pqbi8.png Малюнок 13.а. Ступінчаста дія. ПІД регулятор. | https://habrastorage.org/r/w1560/webt/8s/42/qy/8s42qywxhbxjnahqhy64jydjg8k.png Рисунок 13.б. Ступінчаста дія. Нечітка логіка. |

Для ступінчастого впливу регулятор на основі нечіткої логіки забезпечує кращу якість перехідного процесу. ПІД-регулятор, налаштований автоматично на плавний процес, викликає коливання з перерегулюванням вдвічі перевершують задану сходинку.

## 4.8 Налаштування регулятора на основі нечіткої логіки шляхом оптимізації

Спробуємо підібрати параметри нечіткого регулятора методом оптимізації, як і ми підбирали їх для ПІД-регулятора. Як критерій задамо середньо квадратичне відхилення менше 0.001.

Потрібно зазначити, що цей метод не зовсім правильний, оскільки для професіоналів зрозуміло які кути і які швидкості є максимальними і мінімальними для кожного концертного виробу, що дозволяє ставити обмеження на параметри, що оптимізуються, більш усвідомлено, ми ж задаємо параметри за замовчуванням і дивимося, що виходить.

Метод оптимізації з параметрами за замовчуванням розрахував такі значення діапазонів параметрів оптимізації:

**uMax = 19.377**- Амплітуда керуючого впливу;

**deltaMax = 1.095**- максимальне відхилення;

**divMax = 0.01**- максимальна похідна відхилення;

**div2Max = 2.497**- Максимальна друга похідна відхилення.

У разі простої оптимізації по відхилення отримані параметри забезпечують задану точність, проте при цьому викликають високочастотні коливання керуючого впливу.

Перехідна функція та керуючі дії представлені на малюнку 14.а

|  |  |
| --- | --- |
| https://habrastorage.org/r/w1560/webt/gj/sk/5f/gjsk5fx_hwzmboowifccw8i8iyo.pngМалюнок 14.а. Нечітка логіка. Налаштування щодо відхилення. | https://habrastorage.org/r/w1560/webt/_d/sh/ck/_dshckcuwe5jlfm-ttvhkoqxvgo.pngРисунок 14.б. Нечітка логіка. Налаштування щодо відхилення та кількості спрацьовувань. |

Для того, щоб поліпшити перехідний процес, можна додати критерій оптимізації кількість перемикань регулятора з негативного на позитивне значення регулюючого впливу (схема на рис. 15).

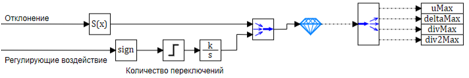


Рисунок 15. Схема для оптимізації за двома критеріями.

Розрахунок методом оптимізації за двома критеріями дає такі значення параметрів:

uMax = 19.714 - амплітуда керуючого впливу;

deltaMax = 1.0496 - максимальне відхилення;

divMax = 0.01 – максимальне похідне відхилення;

div2Max = 1.7931 – максимальна друга похідна відхилення.

Видно, що при додаванні в критерій оптимізації числа спрацьовувань вдалося скоротити частоту перемикання регулятора (див. рис. 14.б). Таким чином можна сказати, що метод оптимізації працює, навіть коли ми нічого не знаємо про фізику об'єкта і просто підбираємо числові параметри, не замислюючись про їхній фізичний сенс.

## 4.9 Створення власного регулятора на основі нечіткої логіки

Вище ми створили регулятор за вже готовою і досить простою схемою, всі терми лінгвістичних змінних з'єднали логічним оператором І. Оскільки у нас кількість термів на входах і виходах однакова, це найпростіше і очевидне рішення.

Спробуємо зробити регулятор, у якого на виході не три терми, а, наприклад, 5: зменшувати швидко, зменшувати, не змінювати, збільшувати, збільшувати швидко. А на вході ті самі.

Змінимо логіку роботи регулятора, спочатку максимально спростивши алгоритм управління.

Запишемо правила:

1) Якщо більше і зростає => скорочувати швидко.

2) Якщо більше =>зменшувати.

3) Якщо норма => не змінювати.

4) Якщо менше => збільшувати.

5) Якщо менше і зменшується => швидко збільшувати.

В цьому випадку у нас для вихідної змінної 5 термів (5 трикутних функцій). Приймаємо, що вони розташовані рівномірно між -uMax та +uMax.

Приймаємо, що трикутні функції складені таким чином, що, коли функції приналежності терму набирають максимального значення, сусідні функції набувають нульових значень (див. рис. 5).

Як параметри для виводу будуть використовуватися лише відхилення та швидкість зміни відхилення.

Для прискорення розрахунків використовуємо фазифікацію виведення методом центру тяжкості точок (див. мал. 7).

Схема регулятора в цьому випадку буде виглядати як показано на малюнку 15.

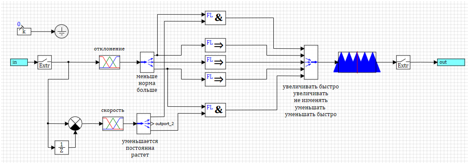


Рисунок 15. Спрощений регулятор на основі нечіткої логіки.

Замість діапазону другої похідної оптимізатора буде братися значення для термо «збільшувати». Спроба налаштувати подібний регулятор методом оптимізації показує, що регулятор налаштовується, але якість регулювання управління системою бажає кращого.

Найкращий результат показаний малюнку 16.

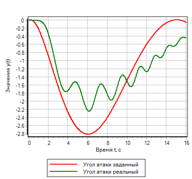


Рисунок 16. Перехідний процес спрощеного регулятора.

Видно, що регулювання відбувається, проте зовсім не так, як нам хотілося б. Справа в тому, що ми здійснюємо вплив, коли відбулося відхилення. Спробуємо увімкнення регулювання в моменті, коли у нас відхилення в нормі, але швидкість показує, що воно зростатиме або зменшиться.

1) Якщо менше і зменшується => швидко збільшувати.

2) Якщо норма і збільшується => зменшувати.

3) Якщо норма => не змінювати.

4) Якщо норма і зменшується => збільшувати.

5) Якщо більше і зростає => скорочувати швидко.

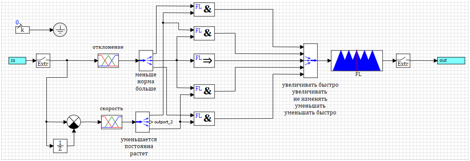


Рисунок 17. Управління швидкості зміни відхилення.

Результати роботи регулятора налаштованого методом оптимізації представлені на рисунках 18а та 18б.

|  |  |
| --- | --- |
| https://habrastorage.org/r/w1560/webt/ol/nm/iu/olnmiuamnqvze-nhtsy-fnfhbhk.png Малюнок 18.а. Перехідний процес. | https://habrastorage.org/r/w1560/webt/7c/px/cr/7cpxcru-79am4butmtzcysqvwuu.png Малюнок 18. Управління |

Управління за швидкістю зміни відхилення значно покращило перехідний процес. Однак, якщо уважно подивитися на набір логічних правил, то бачимо, що відхилення не бере участі в управлінні. Якщо дати ступінчасту дію, контролер управління не буде формувати керуючу дію. На рисунках 19 наведено приклад перехідного процесу при ступінчастому керуючому впливі, видно, що регулятор не видає керуючого впливу, хоча відхилення одно 1.

|  |  |
| --- | --- |
| https://habrastorage.org/r/w1560/webt/aa/4d/of/aa4dofh8aozrx7efjdbeouueppe.png Малюнок 19.а. Перехідний процес. Сходинка | https://habrastorage.org/r/w1560/webt/ec/c_/fq/ecc_fq250svxhfhlmzeyg6_otiy.png Малюнок 19.б. Управління. Сходинка |

Для того щоб відпрацювати швидкі відхилення, додамо керуючий вплив при відхиленні. Збільшуватимемо, якщо менше і зменшуватимемо, якщо більше. Оскільки в наборі правил вже є правила, за яких ми зменшуємо та збільшуємо, ми використовуємо логічний оператор або:

1) Якщо менше і зменшується => швидко збільшувати.

2) Якщо (норма і збільшується) чи більше => зменшувати.

3) Якщо норма => не змінювати.

4) Якщо (норма та зменшується) або менше => збільшувати.

5) Якщо більше і зростає => скорочувати швидко.

Схема регулятора за цими правилами представлена ​​малюнку 20.

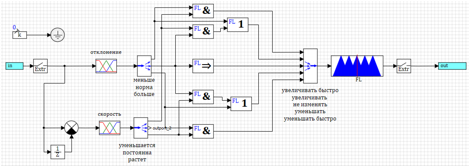


Рисунок 20. Регулятор з керуванням з відхилення та швидкості зміни.

В результаті модифікації, якість перехідного процесу при плавному впливі практично не змінилася, проте при ступінчастому вплив регулятор почав відпрацьовувати сходинку і приводити кут атаки ракети до заданого (див. рис. 21).

|  |  |
| --- | --- |
| https://habrastorage.org/r/w1560/webt/s1/vm/ch/s1vmcha5gih196zh6wcs_gftyam.png Малюнок 21.а. Перехідний процес. Сходинка | https://habrastorage.org/r/w1560/webt/hg/et/c8/hgetc8w8bmosszmw1mkzanwtdli.png Малюнок 21.б. Управління. Сходинка |

Насамкінець, давайте ще раз «покращимо» наш регулятор.

Спробуємо використати другу похідну відхилення, для початку дії, до того як змінилося відхилення та його швидкість. Справді, при додатку сили у нас з'являється прискорення, на яке вже можна реагувати.

Спробуємо додати до закону регулювання замість першої похідної швидкості, другу похідну. Надаватимемо додатково регулюючий вплив у випадку, коли у нас друга похідна показує, що буде відхилення. Загальні правила виглядатимуть практично так само, тільки в дужках у нас три терми, відхилення та швидкість у нормі, а відхиляється друга похідна:

1) Якщо менше і зменшується => швидко збільшувати.

2) Якщо (норма і постійна та розганяється) або більше => зменшувати.

3) Якщо норма => не змінювати.

4) Якщо (норма і постійна та сповільнюється) або менше => збільшувати.

5) Якщо більше і зростає => скорочувати швидко.

Схема даного регулятора наведена малюнку 22. Для економії місця на схемі логічні висловлювання «і», записані у правилах у дужках, обчислюються субмоделі, позначеної «&».

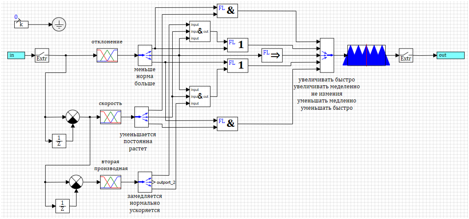


Рисунок 22. Регулятор нечіткої логіки з контролем другої похідної.

Після підбору параметрів методом оптимізації щодо відхилення та кількості включень, отримані наступні параметри:

uMax = 27.4983 - амплітуда керуючого впливу;

deltaMax = 0.0433 - максимальне відхилення;

divMax = 0.0966 – максимальне похідне відхилення;

div2Max = 1.0828 – максимальна друга похідна відхилення.

Перехідний процес показаний малюнку 23. Видно, що отриманий регулятор має найкращими показниками з усіх розглянутих вище, але заданого впливу. Відхилення і управляючі впливу – мінімальні з усіх розглянутих у цьому тексті.

|  |  |
| --- | --- |
| https://habrastorage.org/r/w1560/webt/br/x2/to/brx2tofnkadd9ouxkwczrmstc7m.png ***Малюнок 23.а. Перехідний процес.*** | https://habrastorage.org/r/w1560/webt/qj/d4/sm/qjd4smlgahbfb71rsmadbdomxbe.png ***Малюнок 23.б. Управління.*** |

## Висновки

Регулятор на основі нечіткої логіки може забезпечити більш високу якість перехідного процесу для управління ракетою, ніж ПІД-регулятор.

Налаштування регулятора на основі нечіткої логіки можна здійснювати засобами оптимізації.

Регулятора на основі нечіткої логіки забезпечує більшу гнучкість у налаштуванні та кращу якість перехідного процесу. Але потребує налаштування більшої кількості параметрів.

# 5 Історія штучного інтелекту

Штучний інтелект - це одна з найцікавіших ідей людини. Про нього написано багато дисертацій та ще більше робіт у жанрі наукової фантастики. Про нього можна нескінченно сперечатися і так не дійти єдиної думки. Він скрізь і водночас ніде. А все тому, що не існує єдиної думки щодо того, що ж собою представляє або повинен представляти Штучний Інтелект.

При згадці цих двох слів хтось згадає однойменний фільм Стівена Спілберга, хтось книги Азімова про позитронні роботи. Інші скажуть: «Опустіться на землю і розплющте очі, адже Штучний Інтелект уже серед нас». При цьому деякі вказують на комп'ютерні ігри, в яких віртуальні супротивники намагаються наслідувати своїми діями реальними, їх навіть називають також ІІ. Можливо, старшому поколінню згадається КАІССА і подібні до неї програми, які вже в змозі обіграти навіть чемпіона світу з шахів. Для когось всілякі програмні майстри та пошукові системи є ІІ.

Це питання одне з тих, на які важко дати однозначну відповідь. Більше того, чим більше починаєш у ньому копатися, тим більше зустрічаєш розбіжностей, протиріч та ще більше питань. Визначення начебто має бути, але тільки приховано воно під горою різних наук, зверху - пристойний шар фантастики, далі йде невеликий прошарок філософії, оман і суб'єктивного романтизму, ну і на самому верху - псевдонаука і реклама, що кричить.

Як видно, до істини не так легко дістатися. У різних людей різна думка з цього приводу, і тому складно сформулювати щось спільне. Щоб розібратися в цьому питанні, подивимося, що з цього приводу скаже історія. Виявляється, що сама назва Штучний Інтелект вигадали не фантасти, а вчені. Але про все по порядку.

Справа була в далекому 1956 (на той час вже існувала серія книг Азімова про роботи). У США, у Стенфордському університеті відбувся семінар, на якому запропонували термін Штучний Інтелект. Учасники того семінару неясно уявляли, що це таке і для чого це потрібно. Темою семінару були логічні завдання та способи автоматизації їх вирішення. Початок було покладено.

Через 25 років Барр і Файгенбаум запропонували таке визначення, яке ніким не заперечується майже два десятки років.

Штучний інтелект (ІІ) — це область інформатики, що займається розробкою інтелектуальних комп'ютерних систем, тобто. систем, що мають можливості, які ми традиційно пов'язуємо з людським розумом, - розуміння мови, навчання, здатність розмірковувати, вирішувати проблеми і т.д.

Ось вони основні властивості ІІ - розуміння мови, навчання та здатність мислити. Це мета, якої мають прагнути його майбутні творці. Визначення дуже лаконічне, якраз у стилі математиків, але зупинимося поки що саме на ньому.

Як це зазвичай буває, «майбутні творці» від початку розділилися на два табори: на кібернетиків і нейрокібернетиків.

Кібернетики відразу відмовилися від ідеї повного копіювання розумових процесів людини. Їм достатньо того, щоб на ті самі питання (вхідні дані) пристрій і людина давали однакові відповіді (вихідні дані). А як це відбувається, які механізми та алгоритми знаходяться усередині машини, не має значення. Тому це й назвали кібернетикою «чорної скриньки».

Як тільки комп'ютери набули поширення, кібернетики освоїли їх і відразу ж взялися за справу. Почали вони із невеликих ігор. Іграшки були на зразок хрестиків-нуліків, Scrabble (Ерудит), де перебіг комп'ютера розраховувався шляхом перебору. При знаходженні відповіді перебір припинявся, а при незнаходженні... Виходив комбінаторний вибух або російською мовою — CPU 100%.

Незважаючи на те, що деякі завдання і вирішувалися таким методом, у мене все ж таки не повернеться мова назвати це Штучним Інтелектом, адже людина, яка не перебирає в голові всілякі варіанти своїх дій перед тим, як зробити що-небудь.

Розробники звернули на це увагу та впровадили в алгоритм пошуку спеціальну технологію – евристику. Її суть у тому, щоб обмежити простір перебору спеціальними умовами. Для кожної ситуації вони є різними.

Наприклад, комп'ютер намагається знайти слово, в якому обов'язково має бути буква "б" (Scrabble). Якщо він намагатиметься перебирати абсолютно всі варіанти з літер, то це займе багато часу і буде неефективно. Адже, зустрівши комбінацію з літер «БРП», безглуздо вже підбирати наступну четверту літеру, тому що не існує слів, які починалися б з «БРП». Тому для такої гри завдання евристики полягає у знаходженні неможливих поєднань літер та припинення пошуку, коли він натрапить на таке поєднання.

Евристичний пошук використовується і зараз, зокрема, в деяких антивірусних сканерах.

Інші розробки, якими займалися кібернетики приблизно в той же час, - це доказ теорем. Але, якщо з математичними завданнями комп'ютер справлявся без будь-яких проблем, всі вони легко піддавалися програмної формалізації, то з більш побутовими завданнями вже виникали труднощі. Зауважимо, що такі системи до звичайної людини мають не більше відношення, ніж і попередні, тому що більшість з нас не має видатних здібностей у логічних міркуваннях.

Час минав, технології розвивалися. І ось, у першій половині 70-х почали з'являтися більш цікаві системи, здатні розпізнавати природну мову та спілкуватися нею. Одна з перших у цій галузі - SHRLDU, створена Виноградом (Winograd) у 1972 році.

Вона розуміла пристойну кількість англійських слів, обмежену сферою застосування - світом, створеним з деталей дитячого конструктора. Весь словниковий запас, усі питання, на які можна було отримати нормальну відповідь, були пов'язані лише з цим світом. Інші подібні до неї системи також були обмежені якоюсь сферою застосування. Про самонавчання не йшлося.

Пізніше з'явилися більш просунуті системи, які могли підтримувати розмову майже всі теми. Але так можна стверджувати лише з великою натяжкою, оскільки володіння темами було досить поверхневим. Якщо поставити потребує більш глибоких знань або складно сформульоване питання, то програма відмовлялася його перетравлювати і починала повторювати за вами постійно перепитуючи. Головний її недолік — відсутність повноцінного самонавчання. Отже, системи цього типу досі залишають бажати кращого.

Ще одне з пізніших розробок — системи уявлення знань. Вони були частиною експертних систем. Ці системи створюються у тому, щоб у ролі порадника. Вони можуть аналізувати ситуацію, знаходити вирішення проблеми, проводити діагностику і все інше в тому ж дусі.

Але й тут не все так гладко. Для кожної такої мети потрібно створити свою експертну систему. Крім того знання такої системи завжди стосуються лише однієї вузької сфери застосування. Ну і насамкінець — організація знань у кожній системі своя, оскільки вона безпосередньо залежить від цілей та змісту. Так що про створення системи, яка могла б бути експертом у різних галузях, не може бути й мови.

Що ж, це були лише деякі з розробок кібернетиків. Я спеціально не став тут писати про розвиток роботів, оскільки досягнення у цій галузі й так широко відомі. Кожна з цих розробок досягла великих успіхів. Якщо це розпізнавання мови, воно вже на високому рівні, мислення експертних систем теж не відстає, не дарма ж в нього вкладається стільки грошей. Навчання зараз теж реалізовано, хоча не так добре — воно знову ж таки сильно обмежене областю застосування. Я ще повернуся до навчання, коли говоритимемо про нейромережі.

Як бачимо, знайти кандидата в ІІ нам не вдалося. Зараз йдеться швидше про «деякі характеристики Штучного Інтелекту», ніж про сам ІІ. Усі його складові вже розроблено. Здавалося б, що нам заважає взяти та об'єднати їх у одну взаємопов'язану систему? Наприклад, взяти нейрокомп'ютер, закласти в нього спрощену модель нашого світу (експертну систему) та наділити здатністю розуміти природне мовлення?

Ми можемо вже сьогодні так вчинити. Але чомусь мені здається, що ми матимемо не зовсім те, чого хотілося б. А тепер поставте собі таке запитання — чого ви чекаєте від ІІ? Чи збігаються ваші очікування з визначенням Барра та Файгенбаума, якому вже 27 років?

У той час, коли кібернетики ще тільки починали грати у свої іграшки, інші серйозніші вчені, працювали над більш серйозною проблемою. На підставі нейрофізіологічних даних про будову нейронів, клітин нашого мозку вони намагалися відтворити їх структуру. Це було за кілька років, до того самого семінару, де вперше заговорили про ІІ.

Ці серйозні вчені були повністю впевнені в тому, що єдина річ, здатна мислити, — це наш мозок. Тому все, що має мати таку здатність, має відтворювати структуру нашого мозку. Смілива заява, особливо, якщо враховувати те, що про процеси мислення вони мали досить невиразне уявлення, хоча ні — не уявлення, а просто гіпотезу про те, що мислення людини працює за допомогою її нейросетей.

Тобто, знаючи структуру нейрона, створили свою спрощену його копію і, знаючи, як нейрони об'єднуються в мережі, спробували на основі цих знань розробити логіку, а далі - комп'ютер. А тут працюють уже лише одні математики. Ніхто не каже, що людина користується своїми нейронами так само, як і нейрокомп'ютер. Просто інженери замінили класичні напівпровідники штучними нейронами та намагаються побудувати нову логіку. Про психологічний бік процесу поки що ніхто не замислюється.

Принцип роботи нейромереж дещо відрізняється від напівпровідникових мікросхем – перші не використовують двійкові коди. Але секретною зброєю нейромереж є те, що нейромережі, за чутками, здатні навчатися. Спробуємо розібратися у цьому питанні.

Отже, ми маємо мережу (одношарову або багатошарову), що складається зі спрощених моделей нейронів. Є група вхідних нейронів і, відповідно, група вихідних. При подачі інформації на вхід сигнали пробігають по мережі, посилюючись і слабшаючи, від нейрона до нейрона. Вони можуть йти безпосередньо від входу до виходу, але можуть робити петлі, повертаючись на кілька нейронів назад. Можуть навіть безладно рухатися по всій мережі, але сенс у тому, щоб на виходах сформувався правильно оброблений сигнал.

Якщо на входи подали коефіцієнти квадратного рівняння, то на виході ми маємо отримати його рішення. Якщо на входах зображення рукописного тексту, то на виході має сформуватись його розпізнаний варіант.

На початку роботи, коли нейромережа ще не налаштована, від неї немає жодного спрямування. Щоб вона, крім своєї загадкової назви, була ще й корисна у вирішенні завдань, її треба навчити. Цей процес починається з того, що завдання вирішується за допомогою якихось зовнішніх засобів. Це потрібно для того, щоб зрештою отримати набори питань та правильних відповідей. Далі ці питання подаються на вхід, а на вихід.

Розглянемо, як цей процес проходить одного нейрона. На його входи приходять вхідні сигнали, а на вихід сигнал, про який заздалегідь відомо, що він правильний. Нейрон у свою чергу формує свій вихідний сигнал, підсумовуючи вхідні та застосовуючи до суми спеціальну функцію.

Через війну свій вихідний сигнал нейрона першому етапі відрізняється від зовнішнього. Якщо розбіжність спостерігається, то застосовуються спеціальні алгоритми, які змінюють рівень вхідних сигналів (їхня вага), поки обидва вихідні сигнали не будуть близькі.

Але тут потрібно бути обережним, тому що якщо вони будуть дуже близькі, то нейромережа «перевчиться». І тоді, якщо її завдання складається з, наприклад, розпізнавання тексту, то нейромережа зможе розпізнати лише почерк якоїсь конкретної людини, а інших — ні. А якщо ці два сигнали сильно відрізнятимуться, то нейромережа «недоучиться». Т. е. не зможе розпізнати взагалі нічий почерк.

Щоб такого не сталося, розробник, який займається навчанням, встановлює своєрідну похибку налаштування. Якщо різниця між вихідними сигналами нічого очікувати перевищувати значення цієї похибки, процес навчання вважатимуться завершеним.

Ось якось так ... Насправді все набагато складніше, тому що в мережі багато нейронів. Мережі бувають різні, і ніхто поки що не знає, якою вона насправді має бути, більше того, ніхто не знає, скільки в ній має бути нейронів.

Яким би складним і неймовірним все це не здавалося, але перші нейромережеві комп'ютери були побудовані ще 50 років тому. І хоча вони були не дуже потужні, вони працювали. Основні відмінності таких комп'ютерів від звичайних полягають у самонавченні (хоча я б віддав перевагу терміну «самоналаштовування») і паралельним обчисленням.

Саме тому на них покладають такі великі надії, оскільки наші звичайні процесорні комп'ютери всі обчислення роблять послідовно. Паралельні обчислення дають новий поштовх у розвитку пошукових програм, криптоаналізі та інших областях, пов'язаних з перебором великої кількості інформації.

Зараз нейромережі застосовуються для розпізнавання образів, тексту, промови. Є навіть такі, на основі яких збудовані цілі програмні комплекси, які замінюють собою трейдерів на біржі. Випускаються плати розширення з нейромережами, які потім можна вставити у стандартний комп'ютер і з набором необхідного софту працювати з ними самому. Вони чудово справляються зі своїми завданнями, хоча вони все ще у процесі вдосконалення. Але чи можна на їх основі створити штучний інтелект?

Так, нейромережа здатна на самонавчання, але для цього потрібно заздалегідь знати всі відповіді. Метод спроб і помилок тут не прокотить. Так що самонавчанням це можна назвати лише з великою натяжкою. Її можна налаштувати на вирішення певного завдання. Вона тоді з нею справлятиметься, але більше ні з чим. Щоб її перевчити для нових цілей, потрібна людина, і не просто людина, а експерт у цій новій галузі. Сама нейромережа не здатна вивчити будь-що.

Фішкою нейрокібернетиків було те, що вони намагаються відтворити мозок із його процесами мислення. Але єдине, чого вони досягли, це те, що частково повторили структуру одного лише типу його клітин — нейронів. При цьому прикрутили до цього свої алгоритми, які до нашого мислення не мають жодного стосунку.

У результаті отримуємо ту саму кібернетику «чорного ящика», з входом, виходом і чорти ніж усередині. Процес нашого мислення немає нічого спільного з роботою нейромережі, оскільки нейромережі не користуються аналізом, синтезом, порівнянням, дедукцією. Також вони не зважають на такий фактор, як психіка людини. При відтворенні нейронів опускаються їхні властивості, пов'язані з усім цим, тому процес реалізується лише на поверхневому рівні.

Творці нейромереж, які все ж таки намагаються крім логіки навернути якусь «мову програмування», засновану на психології людини, зіткнулися з серйозною проблемою. Вони намагаються знайти загальну теорію психології, яку легко було б формалізувати. Але справа в тому, що підходів тут десятки, і що ще цікавіше, так це те, що не один з них не посилається на фізіологію, тобто на те, що движком нашої психіки є наші нейромережі. А якщо зв'язку такий і немає?

Теоретична психологія — це самостійна система зі своїми елементами. Вона ніяк не прив'язана до фізіології, тому ця «мова програмування», ця формалізація нашої психології може бути реалізована як на нейромереж, так і на іншій платформі, або взагалі описана математично. Тому нейромережі не такі вже й незамінні, якщо можна обійтися без них.

Настав час повернутися до того питання, яке я поставив наприкінці першої частини, а саме про те, чи настільки добре існуюче визначення Штучного Інтелекту. Начебто всі складові вже створені, але чогось все-таки не вистачає цьому Франкенштейну. Має бути щось, що вдихне в нього життя.

Головне тут зрозуміти, для чого нам потрібен Штучний Інтелект. Вирішувати логічні завдання? Для цього вже може бути звичайний комп'ютер. Розпізнавати образи чи мовлення? Такі технології також вже існують. Може, є якесь ще завдання, яке досить складне? Я думаю, що і для неї знайдеться якась технологія чи розробка. Будь-які наші проблеми можна вирішити окремо і без залучення Штучного Інтелекту. Тоді навіщо він нам потрібний?

Я підозрюю, що не для якогось конкретного завдання. Не будемо ходити навкруги і обманювати себе. Скажемо прямо, що хочемо, щоб штучний інтелект був максимально наближений до людського.

Щоб він був такий же алогічний, мав інтуїцію, здатність народжувати ідеї, приймати рішення, відчувати, співпереживати, щоб при спілкуванні з ним створювалося відчуття, що спілкуєшся з особистістю. Щоб він мав свій погляд на світ, щоб він міг сперечатися і погоджуватися чи не погоджуватися з опонентом. Щоб він був уважним, мав здатність дружити, зберігати секрети, брехати, поважати і недолюблювати, щоб він мав почуття гумору. Щоб він міг кохати.

Мені здається, що це визначення ближче до суті.

## 5.1 Умовні позначення

Після 56-го, коли світ оголосили першу концепцію Штучного Інтелекту, витекло багато води. Вчені, а за ними і прості обивателі розділилися на кілька напрямків. І в кожного з них свій погляд на питання, своє визначення, критерії. У когось в основі лежить думка, що ІІ вже винайдено, у когось навпаки — ІІ винайти неможливо. Для довідки можна було б викласти ці визначення, але я не бачу сенсу займатися копіпастом з тієї ж вікіпедії. Це нічого не дасть. Більшість людей мають свій погляд на це питання і відстоюватимуть те, що відповідає їхнім поглядам.

Тому я скористаюся цим правом і в короткому вигляді викладу свою думку, те, що я маю на увазі, коли пишу «Штучний Інтелект». Загалом це відповідає визначенню «сильного штучного інтелекту», яке запровадив Джон Серль. У ньому говориться, що ІІ повинен бути не моделлю розуму людини, а самим розумом, тією мірою, якою людський розум — це розум. Так, формулювання досить заплутане, але суть, гадаю, ви вловили.

## 5.2 Список невикористовуваної літератури

Кожен поважаючий себе доповідач повинен мати подібний розділ, от і я теж думаю – чим я гірший! Тільки сенс цього розділу трохи протилежний із цілком об'єктивних причин.

Тут я хотів би пояснити моє висловлювання з прологу щодо того, що «я нічого путнього в мережі не знайшов». Насамперед, жодне з основних напрямів досліджень я не вважаю перспективним для створення Штучного Інтелекту (див. Умовні позначення) у найближчому майбутньому. Для інших цілей – на здоров'я, багато що вже зараз використовується.

Деякі напрямки від кібернетики були розглянуті в моїй першій статті[Картина без художника](http://habrahabr.ru/blog/artificial_intelligence/37841.html)». У ній опис технологій перемежовуються з моїми коментарями, кожен з яких дає зрозуміти, чому з даною технологією ми не досягнемо мети. Во[другий статті](http://habrahabr.ru/blog/artificial_intelligence/37989.html)об'єктом критики стала нейрокибернетика. Вже за традицією в ній міститься опис мого погляду на цей напрямок та роздуми на тему, чому для створення ІІ копати треба в іншому напрямку.

За кадром поки що залишилися ідеї фантастів та різноманітні стереотипи. Розглянемо найпопулярніші їх.

Отже, на перше місце серед оман щодо ІІ я поставив серійні моделі роботів, популярних у науково-фантастичних фільмах 90-х. Сюди ж можна віднести більшість роботів шановного мною Азімова і, наприклад, С-3PO. Об'єднав я їх за наступним критерієм — всі вони створені з уже зашитим, запрограмованим на певні дії в певних умовах інтелектом, який або неспроможний сам переписати свою програму (інваріантний), або здатний, але дуже незначно. При цьому кожен з таких роботів має кінематографічну пам'ять.

У чому ж марення цього образу, який є метою кібернетиків? Якщо ці створіння описуються недалекими, то я не проти таке цілком можливо. Але це не ІІ. Для недоумкуватого робота можна написати якийсь алгоритм, але до людського розуму буде дуже далеко. А ось написати програму, яка наділяє робота такими властивостями, що він проходить тест Тьюринга і взагалі розмірковує не гірше за людину, неможливо. Це все одно, що розробити програму навчання людини дошкільного віку, щоб за тиждень (місяць, рік) вона опанувала знання та досвід дорослої людини.

Ну, і на другому місці у мене стоять фантастичні мега-комп'ютери: Скайнет, БПЛА із «Стелс», Вікі з «Я, Робот», Сет із «Універсальний солдат 2». Думаю, список ви можете продовжити і самі. У принципі вони не дуже відрізняються від розглянутих трохи вище роботів, але їх сприймають не як щось людиноподібне, а як програму, яка може бути запущена на комп'ютері. Багато хто сподівається, що подібні штуки висітимуть на серверах з веб-пошуковиками, на рідній машині у складі будь-яких IDE або взагалі у вигляді ОС. При цьому від них чекають на деякі суто людські властивості: мислення, здатність вирішувати нетривіальні завдання, робити складний вибір тощо.

Що тут не так? Справді, на вигляд все красиво і переконливо. З іншого боку, у подібних гіпотетичних системах хочуть використовувати лише деякі з людських властивостей, інші ж через їхню марність не вітаються та ігноруються. Я, наприклад, сприймаю цю ситуацію так: «Мені подобаються яблука, тому виведете мені такий сорт яблунь, які б складалися з одних яблук». Тобто ми хочемо взяти частину інтелекту, і щоб ця частина не дивлячись ні на що була повноцінним інтелектом.

Я пропоную на цьому зупинитись. Але якщо виникне така необхідність, спливе якась відома теорія чи гіпотеза, про яку я не згадав, то вона цілком може опинитися тут, серед інших у списку літератури, що не використовується.

## 5.3 Основні положення

Певною мірою вони випливають із попереднього розділу. Наш Людський ІІ повинен мати максимальну кількість властивостей свого творця. Після запуску він повинен мати ті ж знання, тобто майже ніякі, і ті ж навички, які можна в цілому назвати здатністю до самонавчання. Далі він працюватиме автономно.

Процес навчання при цьому проходитиме в тій самій послідовності, як і в людини, починаючи зі знайомства з навколишнім світом, навчання мови, далі йде розвиток та ускладнення картини світу. Всі ці етапи мають відбуватися так само, як і в людини. Тому важливим критерієм є спілкування з людьми. Без нього ІІ просто не зможе розвиватись. Звідси випливає, що після того, як ІІ буде створено та запущено, знадобиться кілька років на його навчання. Строго кажучи, він і надалі точно продовжуватиме вчитися, якщо будуть необхідні для цього умови.

Можна було б подумати, що набагато ефективніше спочатку впорядкувати якийсь блок інформації спеціальним чином, а потім просто скопіювати його в ІІ. Але розум людини працює з інформацією набагато складніше, ніж пошуковик. ІІ повинен сам, тими самими способами, як і людина, добути цю інформацію, а потім упорядкувати її знову так само як людина.

Я не можу поки що точно сказати, з якою швидкістю відбуватиметься навчання. Адже найважливішу інформацію особливо «в ранньому віці» ІІ отримає із спілкування з людьми як будь-яка дитина. Тож на першому етапі саме це обмежуватиме швидкість навчання. Далі, коли він досягне розумового розвитку хоча б 15-20 річної людини, він зможе багато інформації почерпнути з книг, статей та інших текстових носіїв інформації. При такому розвитку ця інформація може вплинути на його погляди, дати знання, які він зможе взяти і використати.

Ось тут і запитання — з якою швидкістю відбуватиметься навчання? Чи може обмеження швидкості навчання людини є наслідком особливостей устрою його розуму? Наскільки можна прискорити процес за тієї ж функціональності та ефективності? Створені мною часткові моделі поки що не настільки досконалі, щоб дати відповіді на ці запитання. Але вже зараз можна сказати, що варіант, у якому ІІ після запуску зайде в Інтернет, за пару хвилин порозумнішає, а ще за пару — запустить ядерні ракети, відпадає. Яким ми його навчимо, таким він стане.

Щодо навчання, то може виникнути питання — а чи зможе ІІ перевершити людину? Забудемо на хвилину пророцтва фантастів і розсудливо оцінимо ситуацію. У разі інтелект треба сприймати як щось конкретне. Але конкретні інтелекти у різних людей різняться. Є генії, є ідіоти і решта нормальних і не дуже люди. Так що Штучний Інтелект може бути як на рівних серед не найдурніших представників людства, так і перевершувати їх. Адже він проходить такий самий шлях від народження, має ті самі стимули і здібності до пізнання, у нього так само організовані спогади. Тож задайте собі питання, що заважає Вам стати найрозумнішим у світі?

Якщо ми створюватимемо ІІ ідентичним по будові з розумом людини, то ми автоматично наділимо його всіма властивостями та функціями, характерними для людини. Сюди відносяться і почуття, і інтуїція, здатність брехати та прощати – все, що відноситься до особливостей розуму.

Створити щось, що функціонує точно як наш інтелект куди простіше, ніж зображувати з себе бога і з нуля винаходити систему, яка змогла б з моменту запуску автономно (без втручання в архітектуру) розвиватися. Ось коли ми зробимо Людський ІІ, коли зрозуміємо, що як працює, тоді й знатимемо, що і як потрібно оптимізувати, якщо взагалі щось оптимізується.

## 5.4 Як це буде виглядати

Все описане в попередньому розділі — це не більше ніж філософія, ідея, яка має пронизувати всі етапи розробки та обов'язково бути присутньою в кінцевому результаті. Тому, як буде виглядати цей кінцевий результат і присвячується цей розділ.

***Платформа***

Ми вже вирішили, що ми матимемо максимальну функціональну схожість в архітектурах розумів ІІ та людини. Це означає, що абсолютно все, що стосується розуму людини, має бути присутнім і в Штучному Інтелекті. Підкреслю, що подібність має бути функціональною.

Якщо нам потрібно щось «абсолютно» людяне, то тут можуть виникнути деякі проблеми. Людина живе як своїм інтелектом, думками, мріями. Його голову носить тіло, яке постійно дається взнаки і втручається в наші священні думки і мрії. Однак без цього втручання ми б не були собою, оскільки багато процесів в організмі так чи інакше впливають на наше Я. Щось впливає лише поверхово, наприклад, коли хворий зуб заважає нам зосередитися, але є також і такі речі, чиї впливи дещо глибше, як у гормонів. Не будемо також забувати про вроджені страхи, інстинкти тощо. Якщо спробувати скопіювати все це різноманіття, то доведеться, крім самого інтелекту, створити ще й точну копію всього людського організму, і в результаті ми отримаємо людину. Я особисто не бачу в цьому сенсу (хіба що самоствердження) і не бачу можливості досягти цього «неприродним» шляхом. Все ж таки ми не боги і навіть не НЛО.

Отже, рівень абстракції моделі (системи) має бути вищим, ніж рівень клітин (нейронів). У такому разі створена модель буде платформонезалежною. Пріоритетною платформою є комп'ютер. Рівень абстракції повинен бути таким, щоб всю систему можна було б реалізувати об'єктно-орієнтованою мовою. Оскільки будь-яку інформацію можна закодувати з допомогою тексту, то, щоб організувати пам'ять, необхідно лише розробити спосіб модуляції (кодування) і розмітки. Головні плюси такого рішення: економія на дорогій позаземній техніці завдяки використанню звичайних комп'ютерів із звичайними жорсткими дисками для зберігання пам'яті та економія на програмістах, що працюють із позаземною мовою програмування, на користь старого доброго С та подібних до нього.

***Взаємодія***

Із платформою визначилися. Тепер приступимо до опису властивостей, якими вона повинна мати в першу чергу. Наведу думки Minski на рахунок Штучного Інтелекту:

Дослідження в галузі штучного інтелекту спрямовані на розробку програм, що вирішують такі завдання, з якими зараз краще справляється людина, оскільки вони вимагають залучення таких функцій людського мозку, як здатність до навчання на основі сприйняття, особливої ​​організації пам'яті та здатності робити висновки на основі суджень.

Ось воно – здатність до навчання на основі сприйняття. Щоб інтелект отримував інформацію, щось має постачати його нею. У людини цю функцію виконують органи чуття. Але відтворити кожен із них буде важко. Однак ця складність у жодне порівняння не йде з проблемою зберігання отриманої інформації. Адже її потрібно не просто записувати та відтворювати, а виконувати над нею складніші дії, для чого вона повинна особливим чином інтерпретуватися та зберігатися, так як вона зберігається у людини.

Які з розробок у цій галузі зараз існують? Напевно, найважливіша з них і, мабуть, тому найрозвинутіша — це зір, а саме розпізнавання образів. Але рівень її розвитку не дозволяє використовувати її повною мірою, тобто так, як ми. Тож перше, від чого нам доведеться відмовитись, — це зір. Більшість людей зір — це основне джерело інформації, тому необхідно знайти гідну заміну.

Подумаємо, навіщо вже зараз можна спробувати організувати нормальну «людську» пам'ять. Можливо, багато хто скаже «текст» – і матиме рацію. Хоча в людини немає такого окремого органу, але ж ми вирішили, що подібність буде не тупо один в один. Вже зараз існують системи зберігання знань як саме тексту. Звичайно в такому вигляді, як вони зараз знаходяться, використовувати їх не можна, тому що, знову ж таки, людина їх аналоги використовує по-іншому, але це вже дещо.

До текстової форми спілкування можна віднести і частково — слух. Після невеликого дослідження літератури з сліпих дітей з'ясувалося, що обов'язковою умовою для формування мови є наявність не текстової інформації, що надходить від органів чуття. Поки що ми не визначилися з жодним з них, але виходить, що не можна навчитися мови без інформації, що надходить від органів чуття, тому що слова не можуть чіплятися одне за інше і триматися тільки один за одного (…і тут мені згадалися системи знань …). Обов'язково має надходити якась інформація крім слів, щоб їх було з чим пов'язувати в пам'яті.

***Сприйняття***

Як було зазначено раніше, реалізувати якийсь із людських органів почуттів поки неможливо, хіба що лише зовні. Отже, нам потрібно вигадати абсолютно новий орган, який даватиме інформацію про зовнішній світ.

Зізнаюся, спочатку я мала ідею використовувати віртуальний світ, створений на комп'ютері у вигляді тривимірної інтерактивної моделі. Така собі Матриця для ІІ. Тим більше, що таких зараз багато, чого тільки одна Half Life 2 варта. Він буде простіший за наш, і зір організувати буде простіше. Але навіть тут виникне купа проблем, тому що ми не знаємо, як у людини організована зорова пам'ять, а з'ясовувати це методом спроб та помилок на такій складній моделі буде дуже довго, навіть для такого щодо простого світу.

Не обов'язково, щоб остаточна версія ІІ була забезпечена лише одним органом почуттів. На першому етапі основним завданням цього органу є допомога у розробці та налаштуванні механізмів навчання, пам'яті та інших, щоб вони відповідали людським. Звідси і випливають основні вимоги щодо нього.

Він має бути простим, але дає широкі можливості у взаємодії з ІІ. Він не повинен своєю складністю відволікати від роботи над налаштуванням пам'яті, змушуючи натомість займатися громіздкими алгоритмами розпізнавання та методами інтерпретації для подальшого зберігання. При цьому технології, на основі яких він буде побудований, повинні бути достатньо вивчені та розвинені, щоб його використання не довелося винаходити.

І ось одним сонячним днем ​​через півтора кілометри і півтора альбоми я зрозумів, що це має бути. Векторна графіка.

Так-так-да, вона проста, безкоштовна і в неї ціла купа переваг. Хоч це й суто комп'ютерна річ, вона може бути спрощеним аналогом людського зору і частково навіть дотику. Вона дає величезну свободу дій, оскільки можна сформувати ціле оточення з різних предметів із різними кольорами. Подумавши ще трохи, можна створити динамічні предмети, дозволивши їм рухатись, змінюватись, взаємодіяти, народжувати при цьому нові предмети.

У додатку до реального світу предмети будуть аналогами іграшок для маленького ІІ, маніпулюючи з якими він отримуватиме різну інформацію. Ці об'єкти можуть наслідувати реальні предмети і деякі їх властивості, щоб можна було вчити ІІ їх назв. Коротше місце для фантазії тут величезний. З однією лише такою системою можна досягти значно більшого, ніж просто розвитку однорічної дитини. Адже у векторі можна організувати і цифри, навіть музику.

Ну от тепер портрет можна вважати завершеним. Зовні наш людяний штучний інтелект буде виглядати, як віконце, в якому є область для малювання, поле для введення тексту, кнопочки.

Може зовні це й нагадує гібрид Corel Draw та тамагучи, але насправді це чудова платформа для випробування та доопрацювання різних ідей, яких, треба сказати, вже маса.

## 5.5 Епілог

Тим, хто дочитав до кінця і вже забув про те, що було спочатку, присвячується.

Це не науково-фантастична ідея та не прогноз. Ця концепція замислювалася і розвивається у тому, щоб її можна було реалізувати. Я не заперечую, що ІІ може бути реалізований іншими способами. Просто на мій погляд, який я як міг докладно виклав, інші напрями вимагатимуть набагато більших витрат як у часі, так і в грошах.

У принципі, основна частина того, що написано в цій статті, сформувалася ще кілька місяців тому. Так що більшу частину часу я займався тим, що намагався зрозуміти, як повинен працювати Штучний Інтелект (див. Умовні позначення), як мають бути влаштовані його системи, які методи та науки допоможуть розібратися у цьому. Незважаючи на те, що до готових моделей поки далеко, все ж таки є багато міркувань та ідей. Якщо читачі мають таке бажання, я напишу статтю на цю тему.