

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра математики  
Факультету інформатики

**Курсова робота**  
**за спеціальністю 113 Прикладна математика**  
**освітня програма «Прикладна математика»**

**УПРАВЛІННЯ РОБОТОМ-КУР'ЄРОМ В ЗАКРИТИХ ТА  
ВІДКРИТИХ СИСТЕМАХ**

Керівник курсової роботи  
асистент Вознюк Я. І.

\_\_\_\_\_ (підпис)  
“ \_\_\_\_ ” \_\_\_\_\_ 2022 р.

Виконав студент  
3-го року навчання спеціальності  
113 “Прикладна математика”  
Сердюк Федір Олексійович  
“ \_\_\_\_ ” \_\_\_\_\_ 2022 р.

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛІАНСЬКА АКАДЕМІЯ»  
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ  
Зав.кафедри інформатики,  
доц., канд. ф.-м. н.  
С. С. Гороховський

(підпис)

„\_\_\_\_\_” \_\_\_\_\_ 2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ  
на курсову роботу

студенту 3-го курсу факультету інформатики курсу  
Сердюку Федору Олексійовичу

ТЕМА Управління роботом-кур'єром в закритих та відкритих системах

Вихідні дані:

- Конструкція робота
- ПІД-регулятор для робота
- Метод налаштування ПІД-регулятора

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Календарний план

Анотація

Вступ

1 Проблематика

2 Конструкція робота

3 ПІД-регулятор

3.1 Загальна інформація про ПІД-регулятори

3.2 ПІД-регулятор для робота кур'єра

4 Налаштування ПІД-регулятора

4.1 Відомі методи налаштування

4.2 Метод налаштування для робота кур'єра

5. Результат

Висновки

Список літератури

Дата видачі „\_\_\_\_\_” \_\_\_\_\_ 2022 р. Керівник \_\_\_\_\_  
(підпис)

Завдання отримав \_\_\_\_\_  
(підпис)

**Тема:** Управління роботом-кур'єром в закритих та відкритих системах

Номер	Назва етапу курсової	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу	жовтень	
2.	Огляд технічної літератури за темою роботи	листопад	
3.	Аналіз ПІД-регулятора та методів налаштування	грудень-січень	
4.	Розробка конструкції робота та застосування ПІД-регулятора	лютий-березень	
5.	Розробка методу налаштування ПІД-регулятора	квітень	
6.	Текстове оформлення результатів роботи	травень	
7.	Захист курсової роботи	червень	

Студент Сердюк Федір Олексійович

Керівник Вознюк Ярослав Іванович

„\_\_\_” \_\_\_\_\_ 2022 р

## Зміст

<i>Анотація</i> .....	5
<i>Вступ</i> .....	6
1. <i>Проблематика</i> .....	7
2. <i>Конструкція робота</i> .....	8
3. <i>ПІД регулятор</i> .....	11
3.1 <i>Загальна інформація про ПІД-регулятор</i> .....	11
3.2 <i>ПІД-регулятор для робота кур'єра</i> .....	12
4. <i>Налаштування ПІД-регулятора</i> .....	17
4.1 <i>Відомі методи налаштування</i> .....	17
4.2 <i>Метод налаштування для робота кур'єра</i> .....	21
5. <i>Результат</i> .....	24
<i>Висновки</i> .....	25
<i>Список літератури</i> .....	26

## АНОТАЦІЯ

Метою курсової роботи була розробка робота-кур'єра. Під час роботи використовуючи середовище Unity 3D було сконструйовано робота, застосовано до нього ПД-регулятор та розроблено метод підбору коефіцієнтів до ПД-регулятора. Метод підбору включає в себе евристичні експерименти та сітку пошуку, таке поєднання робить цей спосіб швидшим та точнішим за стандартні.

**Ключові слова:** робот-кур'єр, ПД-регулятор.

## ВСТУП

Світ прямує до автоматизації ще з часів промислової революції. Впровадження у виробництво механізмів дозволило значно збільшити потужності заводів та зменшити ручну працю людей. У двадцять першому сторіччі тенденція на автоматизацію продовжилась і активно триває. Всюди, де це можливо, людей замінюють роботи. Тепер машина може робити операції, обслуговувати людей, спілкуватись, керувати машиною чи навіть літаком.

Однак на сьогодні ще багато галузей недостатньо чи взагалі не автоматизовані, хоча теоретично роботи там можуть бути. Кур'єри – теж не виключення. Крім використання доставки в закритих системах, як склади, де постійно треба переміщувати якісь об'єкти, зараз активно розвиваються служби доставки. Часто службам не вистачає потужності та доставка може тривати довгий час, тому створення робота-кур'єра допоможе компенсувати відсутність вільних людей та пришвидшити кур'єрські послуги. Зараз активно йде робота над роботами-кур'єрами, деякі компанії їх активно тестують, але на сьогодні все ще немає широкого застосування таких роботів – у наявних екземплярів присутні недоліки, які не дають їх використовувати.

Виходячи з цих тенденцій, за мету даної роботи були поставлені задачі дослідити наявні методи, які можуть бути застосовані для роботи робота-кур'єра, модифікувати їх та запропонувати свої розв'язання проблем, які можуть виникнути під час розробки кур'єра.

Використання запропонованих рішень дозволить скоротити час на розробку та налаштування роботів. Це наблизить нас до широкого застосування та автоматизацію в цій сфері.

## 1. ПРОБЛЕМАТИКА

Перш ніж думати, який алгоритм пошуку шляху застосувати, треба знати можливості робота, його конструкцію, як він пересувається, які має датчики. Тому в першому розділі було вирішено описати як робот сконструйований і алгоритм, за допомогою якого він рухається по дорозі в закритій системі. Зараз робототехніка дуже активно розвивається і існує просто безліч можливостей. Коли здійснювався пошук, які датчики можна встановити на робота, було побачено велику кількість різних продуктів різної якості і за різну ціну. Так само й з алгоритмами, останні роки дуже швидко розвивається автопілот, багато машин вже мають такі можливості, що вражають.

Було взято до уваги, що робот-помічник може стати масовим продуктом і недоцільно буде використовувати складні конструкції, дорогі датчики та важкі алгоритми. Саме тому було важливо використати ті елементи, які будуть задовольняти ціною клієнтів, яких може зацікавити робот, але яких буде достатньо для того, щоб добре виконувати свої функції. Так само й з алгоритмом, він мав бути простий і зрозумілий, щоб його можна було легко змінити і модифікувати, для використання в специфічних задачах, а також щоб він не потребував потужної та дорогої системи.

Важливою умовою також є те, щоб до робота було нескладно застосувати різноманітні алгоритми пошуку шляху, ніякі особливості конструкції та спосіб пересування по дорозі не мають обмежувати можливості пошуку шляху.

Використовуючи вищеописані вимоги та умови, ми взяли відомі конструкції та алгоритми, модифікували їх та створили робота. А згодом за допомогою нього було досліджено проблему налаштування ПД коефіцієнтів та запропоновано вирішення цієї проблеми.

## 2. КОНСТРУКЦІЯ РОБОТА

Створення конструкції – це перше, що треба було зробити перед тим як застосовувати ПДД регулятор. А також, щоб зрозуміти як працює застосування, треба знати як робот виглядає, де знаходяться його датчики. Як було написано вище, робот має бути простий, але практичний, тому його конструкція зовсім нескладна і, насправді, тема з ПДД регулятором та рухом по лінії дуже добре досліджена, тому за основу було взято одну з відомих конструкцій[1]:

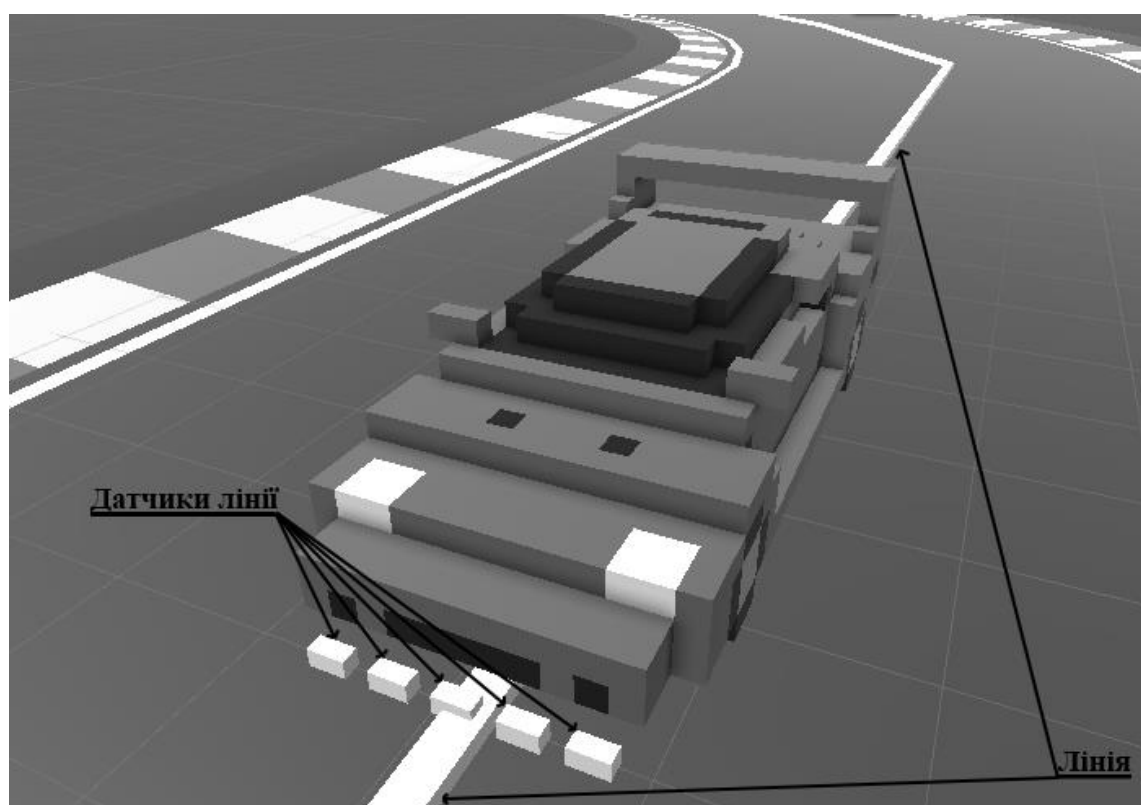


Рис. 2.1: Вигляд робота

Робот має п'ять датчиків лінії, кожен з яких може або почати бачити лінію, або закінчити. Оскільки модель була реалізована в програмному середовищі Unity 3D, було застосовано схожі методи, тобто OnTriggerEnter та OnTriggerExit, які спрацьовують, коли колізія з об'єктом (в даному випадку з

лінією) почалась або закінчилась відповідно. Схематично програма кожного з датчиків виглядає так:

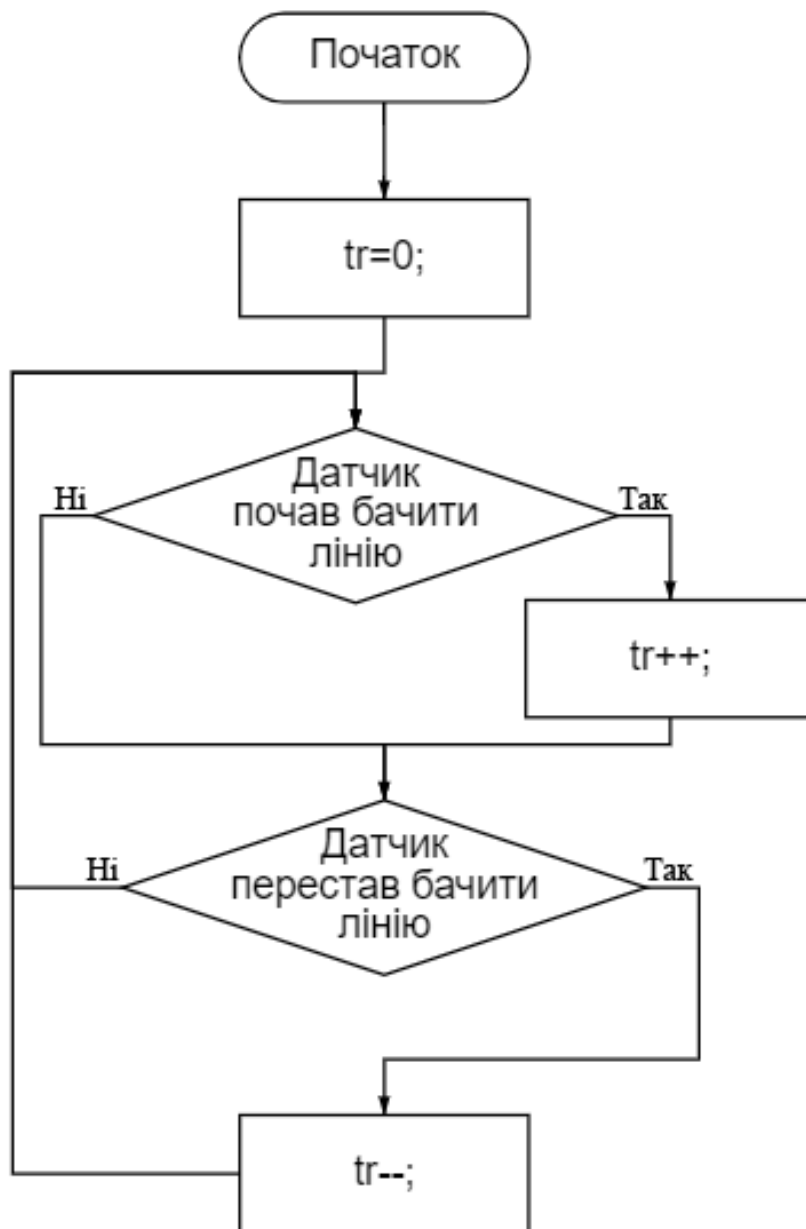


Рис 2.2 Робота датчиків

Тобто кожен датчик має змінну “tr”, яка з самого початку дорівнює нулю. Коли починається колізія з об’єктом, який має тег “route” до змінної додається одиниця, коли колізія закінчується – одиниця віднімається. Таким

чином змінна дорівнює нулю, коли нема колізії з лінією, та одному, коли колізія є. Також присутній метод “haveTrigger()” який повертає значення змінної.

Таке розташування датчиків дозволяє дізнатись, наскільки від лінії відхиляється наш робот, що дуже зручно для застосування ПІД регулятора.

### 3. ПІД-РЕГУЛЯТОР

#### 3.1 Загальна інформація про ПІД-регулятори

Системи автоматичного управління (САУ) створені для того, щоб автоматизувати як можна більше систем керування. Працює це так, що існує регулятор з заданим бажаним станом об'єкту, датчики слідкують за станом цього об'єкту і повідомляють регулятору, якщо стан має помилку, то регулятор посилає сигнал елементу керування і виправляє відхилення.

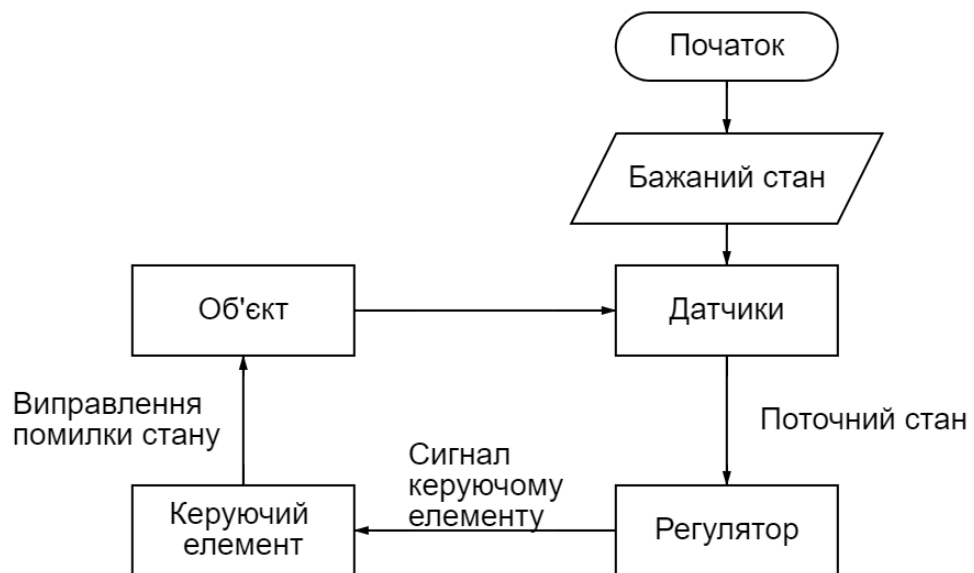


Рис. 3.1 Система автоматичного управління

Зараз найбільш розповсюджений з САУ є пропорційно-інтегрально-диференціальний (ПІД) регулятор. ПІД-регулятор складає 90-95% від усіх САУ та використовуються в клімат контролях, літальних апаратах, в військовій сфері та багато іншого. Рівняння такого регулятора має вигляд:

$$K_p e + K_d \frac{de}{dt} + K_i \int_0^t e(t) dt$$

Тобто бачимо 3 складових, перша з яких – пропорційна Р, це сама проста і основна частина ПІД-регулятора, сенс якої в тому, щоб протидіяти

відхиленню від бажаної величини. Тобто в залежності від значення помилки  $e$  корегується напрям руху так, щоб  $e$  наближалось до нуля, і чим більше відхилення, тим швидше наближення, бо множник  $K_p e$  стає більшим. Але в цій складовій є суттєвий недолік, якщо використовувати лише Р складову, регулятор не стабілізується в заданому значенні, а буде коливатись.

Друга складова – інтегральна І. Вона накопичує  $e$ , тобто дорівнює сумі помилок. Якщо є якась постійна помилка – І її компенсує. Але головним недоліком є те, що якщо таких помилок немає, то ця складова навпаки буде створювати відхилення

Останньою є диференціальна D складова, вона залежить від швидкості зміни відхилення та потрібна для того, щоб протидіяти майбутнім помилкам. Тобто  $\frac{de}{dt}$  дорівнює різниці поточної і минулої помилок. Недоліком є нестійкість до шумів[2].

### 3.2 ПІД-регулятор для робота кур'єра

Далі буде описано, як працює ПІД-регулятор в роботі, який був описаний в минулому розділі. Також зазначалось, що тема ПІД-регулятора добре досліджена і створення чогось зовсім нового немає сенсу, тому за основу було взято код з того ж джерела, де бралась основа для конструкції[1].

Так як все було реалізовано в середовищі Unity 3D і багато речей відрізняється від реального життя, такі як: відсутність зовнішніх факторів, замість звичного поняття часу – кадри, переміщення відбувається в координатному просторі – треба адаптувати ПІД-регулятор для роботи віртуальному світі, тому спочатку буде показано як працює ПІД-регулятор для даного робота, щоб було зручніше пояснювати як відбувається налаштування та підбір коефіцієнтів.

Блок схема програми, за якою працює даний робот, виглядає наступним чином:

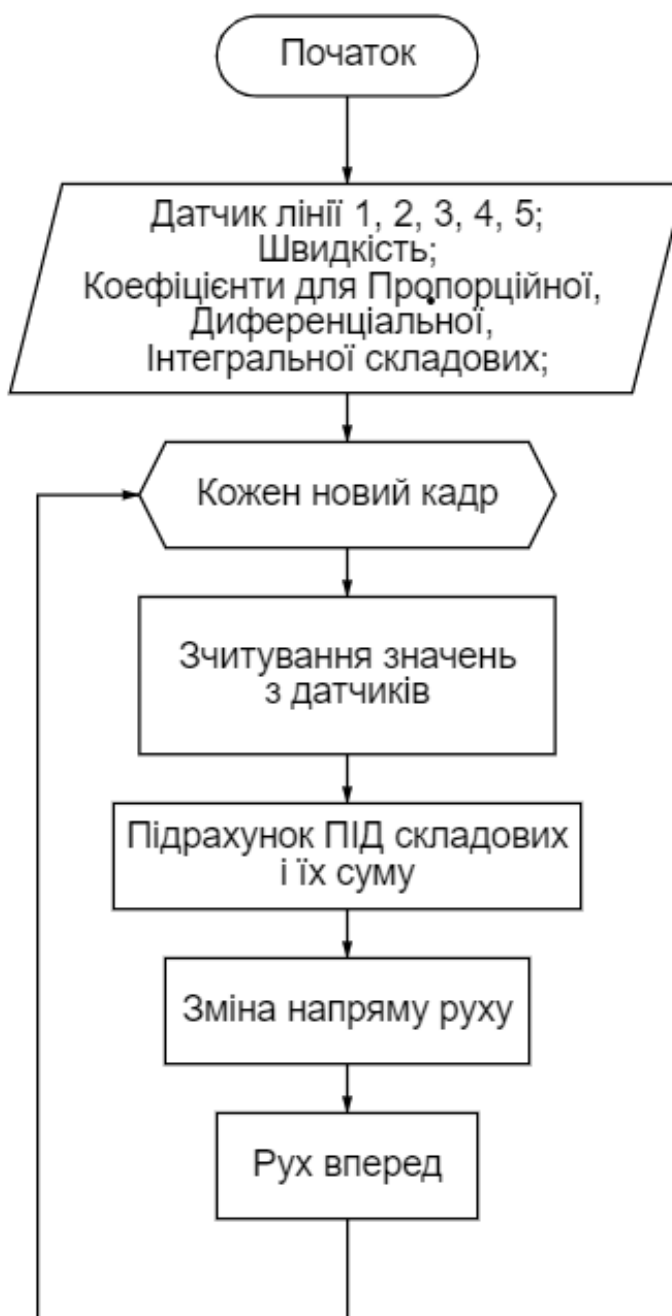


Рис 3.2 Робота ПІД-регулятора

Перед запуском робота треба дати посилання на датчики лінії, задати швидкість, за якою буде рухатись робот, та значення коефіцієнтів, оскільки час в віртуальному світі вимірюється кадрами, то кожен новий кадр буде зчитуватись датчиками поточний стан робота, рахуватись помилка і сума складових пропорційно-інтегрально-диференціального регулятора, змінюватись напрямком робота і просування його вперед. Так як в циклі майже

кожна дія містить в собі складні процеси – буде детально описано окремо всі пункти.

Першим йде зчитування даних з датчиків, одночасно з цим і рахується відхилення від бажаного результату. Наступна блок схема показує як це працює:

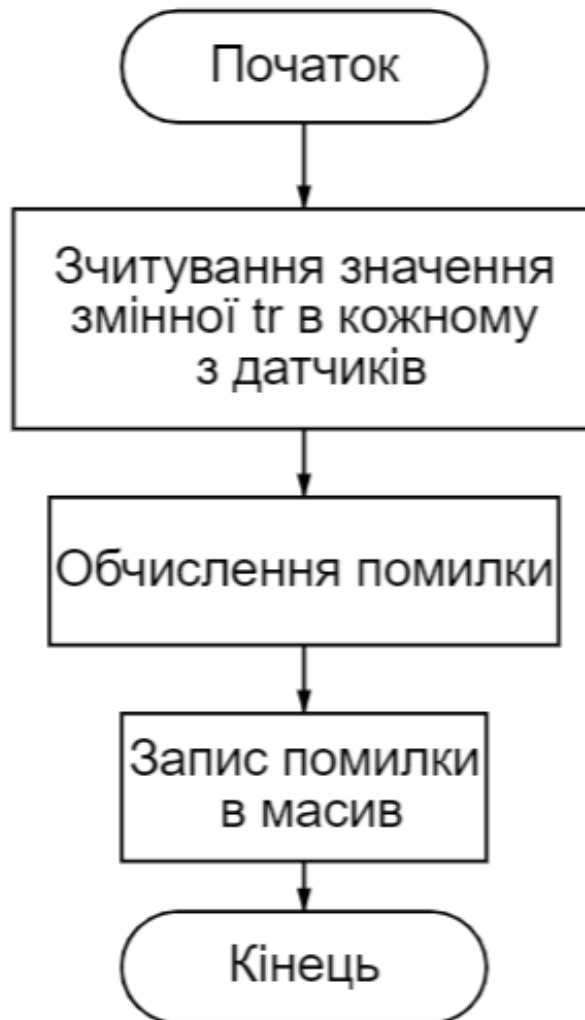


Рис 3.3 Зчитування значень з датчиків

Як було описано в минулому пункті розділу, кожен датчик має змінну, яка вказує чи бачить датчик. В залежності від того які датчики бачать лінію, обраховується помилка, чим далі від центру було помічено лінію – тим більше помилка. Масив було створено для інтегральної складової через один з недоліків, який називається насиченням. З часом сума помилок може стати

надто великою, що збиває роботу ПІД-регулятора. Результат цього було помічено при тестуванні, з часом робота починало відхиляти в одну з сторін, тому було вирішено рахувати суму не всіх помилок, а 10 останніх.

Наступним йде підрахунок ПІД складових і їхня сума. В усіх пропорційно-інтегрально-диференціальних регуляторах цей процес відбувається приблизно однаково, але для уточнення буде наведено блок схему:

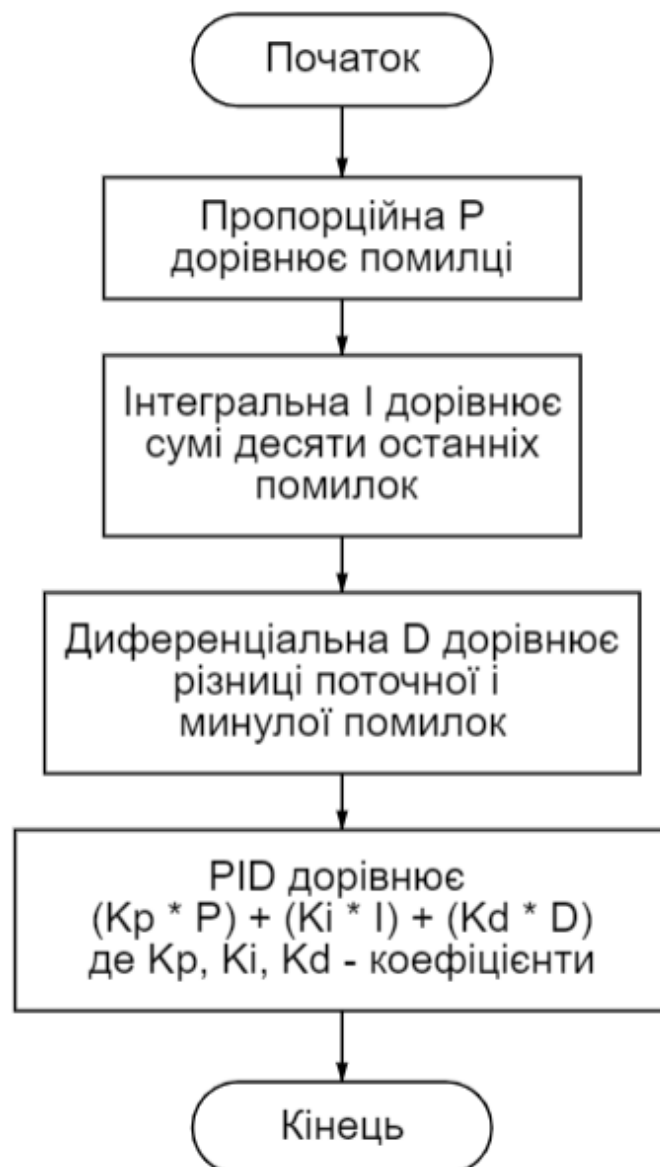


Рис 3.4 Обчислення PID

Після обчислення ПІД ідуть два останніх кроки, які можна поєднати в один пункт – пересування робота. Блок схеми роботи для демонстрації нема сенсу, бо цей пункт надто простий, тому буде показано як це виглядає в коді:

```
car.transform.Rotate(0, PID_value, 0);  
car.transform.Translate(-speed/100, 0, 0);
```

Тобто ПІД рахує, в яку сторону і на скільки треба скорегувати рух, а далі обчислюється, на яку відстань рухається робот за один кадр.

Як можна побачити, вибір саме ПІД-регулятора як САУ для нашого робота є аргументованим – він дуже універсальний та простий, не було ніяких труднощів з застосуванням його в віртуальному світі. Головною проблемою залишається його налаштування, що буде описано в наступному розділі.

## 4. НАЛАШТУВАННЯ

### 4.1 Відомі методи налаштування

ПД-регулятор – перевірена часом та добре досліджена тема. Багато наукових робіт та статей в інтернеті, він багато де застосовується. Та при дослідженні цієї теми було помічено сильний недолік – універсальна система не має універсального та зручного налаштування. Підбір коефіцієнтів залишається актуальною проблемою, існує велика кількість різних методів для пошуку коефіцієнтів, але кожен з методів має свої суттєві недоліки і немає єдиного найкращого способу, який буде однаково швидкий і точний. Зараз підбір відбувається за допомогою евристичних експериментів, які спираються на метод проб і помилок, де все залежить від людини і її досвіду до різних нейронних мереж. Нижче буде описано декілька методів, які часто зустрічаються в різних джерелах, щоб порівняти їх з запропонованим в кінці рішенням.

Першим буде описано метод, де людина самотійно здійснює підбір коефіцієнтів, використовуючи лише досвід і експерименти. При тому, що існують нейронні мережі та безліч різних методів – метод проб і помилок залишається найпопулярнішим з усіх способів. При дослідженні цього методу було знайдено різні інтерпретації цього шляху налаштування, які відрізняються часом і точністю, тому буде описано основний, найбільш вживаний:

Починається все з налаштування пропорційної П складової, тобто інтегральна і диференціальна мають бути вимкнені. Починаючи з невеликих значень, треба поступово збільшувати коефіцієнт. В контролері, де працює лише П, завжди є якесь відхилення, тому збільшувати коефіцієнт треба до тих пір, поки постійна помилка не стане мінімальною:

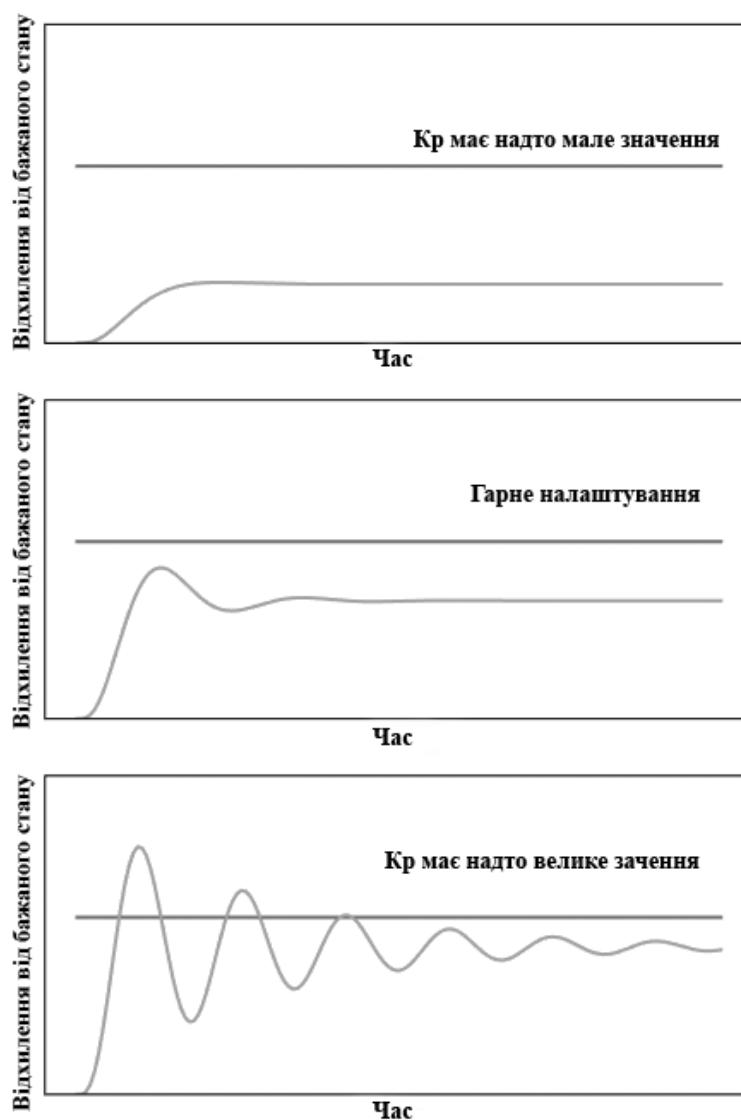


Рис 4.1 Налаштування П складової

Далі до П складової додається інтегральна і контролер починає працювати як ПІ-регулятор. Інтегральний компонент гарантує, що після початкової швидкої реакції П-компонента залишок похибки керування компенсується з часом. Тут налаштування відбувається навпаки, спочатку встановлюється велике значення і поступово знижується. Знижувати треба до тих пір, поки час, за який процес стає стабільним, не стає найменшим:

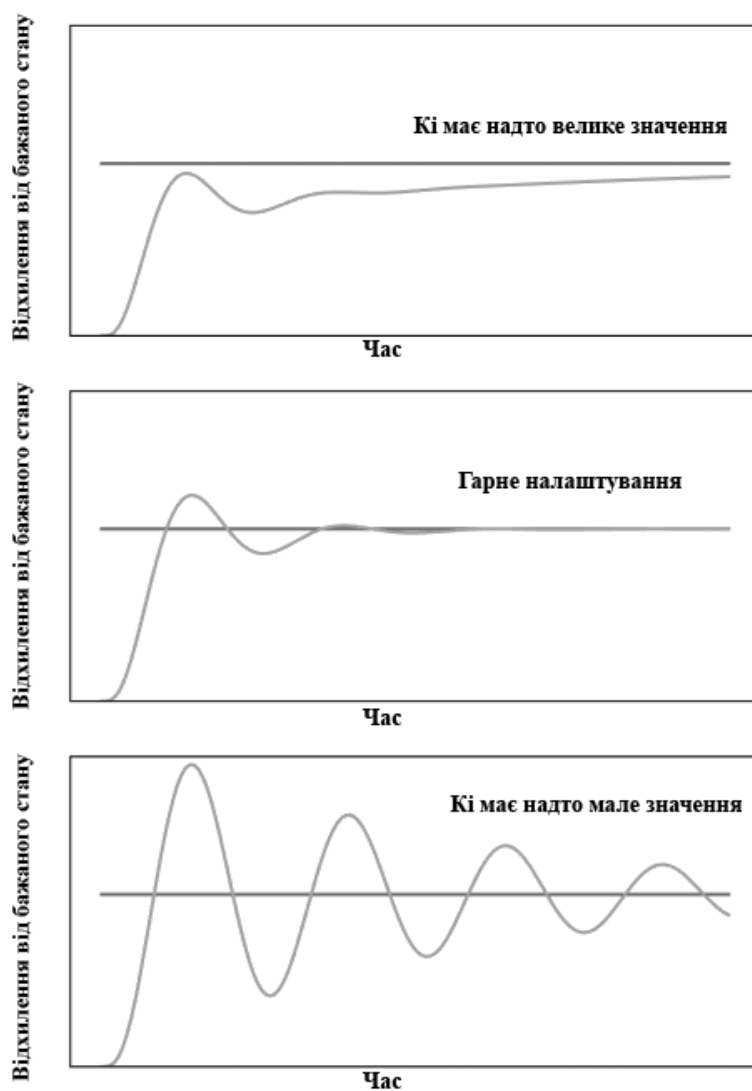


Рис 4.2 Налаштування І складової

Останньою додається диференціальна складова, яка призначена для протидії відхиленням від цільового значення, які прогнозуються у майбутньому. Починається все з малих величин, які поступово треба збільшувати. Чудово підбраною величиною буде вважатись така, при якій коливання майже зникнуть:

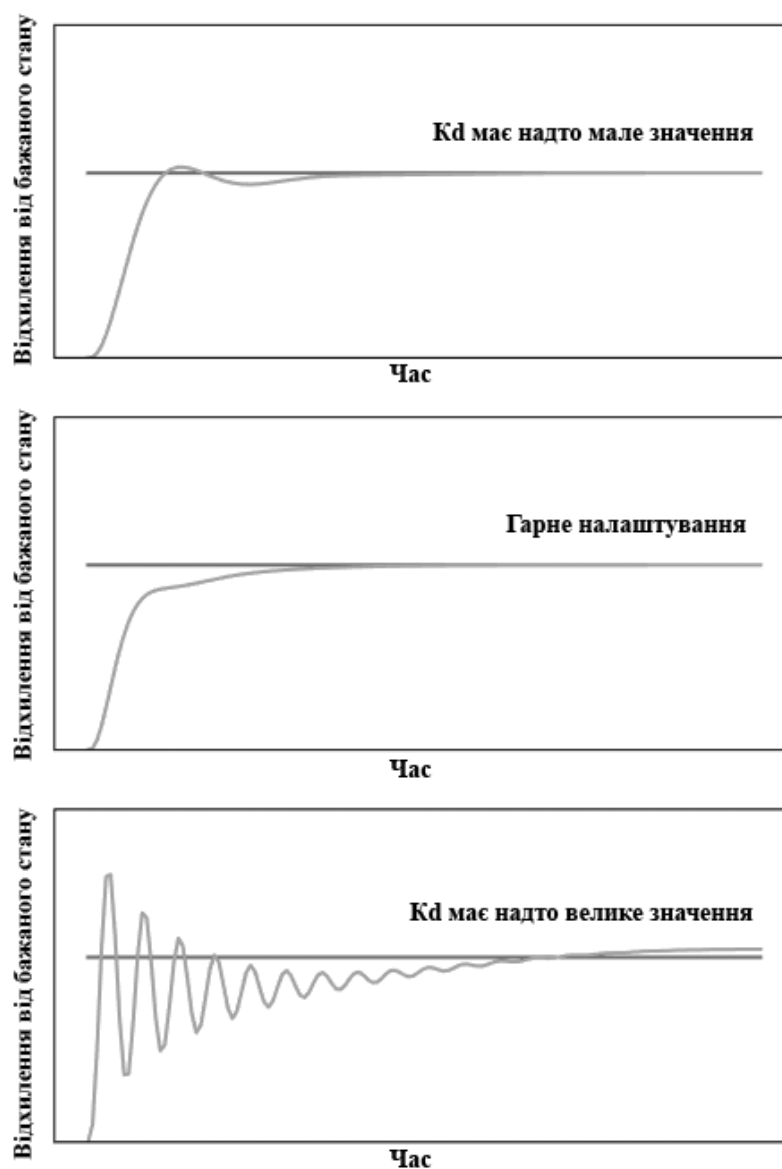


Рис 4.3 Налаштування D складової

Хоч цей метод і популярний та має багато переваг, але в ньому і вистачає недоліків. Швидкість і точність цього методу залежить лише від людини та її досвіду. До того ж, всі люди схильні помилятися, тому ідеально налаштувати ПІД-регулятор таким шляхом скоріш за все не вийде.[3]

Далі мова піде про системи автоматичного налаштування параметрів ПІД-регулятора. Тема автоматизації є актуальною не тільки в налаштуванні регулятора, всюди шукають способи знизити витрати людських сил. Хоч і пропорційно-інтегрально-диференціальний регулятор використовується давно та методи автоматичного налаштування активно досліджуються, але поки нема простого і універсального шляху автоматичного налаштування, та

зазвичай регулятори, які налаштовані вручну, мають кращу якість. Також недоліком є те, що потрібен потужний пристрій або багато часу, наприклад для застосування нейронних мереж.

Методів автоматичного підбору вистачає, але зазвичай вони розроблені для конкретних випадків. Одним з видів є табличне налаштування, в ньому коефіцієнти обираються з таблиці, коли змінюються навколишні умови. Іншим методом є використання нейронних мереж, для цього їй треба навчити підбирати коефіцієнти на основі схожих робіт, які налаштовувала людина. [2]

Обидва методи мають у нашому випадку недоліки. В першому має бути готова таблиця для різних умов, чого у нас нема. В другому випадку треба навчити нейронну мережу на основі інших випадків, але такий підбір коефіцієнтів буде не точним, а навчання такої нейронної мережі, щоб вона нормально працювала, буде потребувати багато часу та даних.

#### **4.2 Метод налаштування для роботи кур'єра**

Коли здійснювалось дослідження налаштування ПД-регуляторів, спочатку було взято курс на повністю автоматичний підбір та нейронні мережі. Але через недоліки, описані вище, було прийнято рішення змінити метод. Оскільки обидва способи мають свої недоліки та переваги, було вирішено поєднати їх так, щоб взяти найкраще. Працює це так, що спочатку людина за допомогою методу евристичних експериментів виявляє припустимі граничні значення, які потім вдосконалюються за допомогою програми. Найоптимальнішим і доречним в даній ситуації буде метод пошуку у сітці. Це простий, робочий метод, який буде краще працювати для даної нескладної задачі, ніж нейронні мережі чи інші методи, бо вони потребують більше ресурсів і часу.

Використовуючи описаний вище метод спочатку було знайдено за допомогою підбору початкові значення, що зайняло приблизно пів години і мало такий результат:

Kp	0.5
Ki	0.001
Kd	5
Speed	3

Рис 4.4 Підібрані за допомогою методу проб і помилок коефіцієнти.

Після того як були знайдені приблизні коефіцієнти, почався етап знаходження коефіцієнтів за допомогою програми. Щоб вибрати найкращу з усіх комбінацій треба спочатку створити систему оцінювання і, насправді, в ПД-регуляторі вона вже готова – це відхилення. Чим більше робот знаходиться в бажаному стані – тим менше відбувається помилок. Тому найкраща комбінація буде та, де найменша сума помилок. Далі було створено сітку пошуку:

```
private float[] Kp = { 0.45f, 0.46f, 0.47f, 0.48f, 0.49f, 0.5f, 0.51f,
0.52f, 0.53f, 0.54f, 0.55f };
private float[] Ki = { 0.0005f, 0.0006f, 0.0007f, 0.0008f, 0.0009f,
0.001f, 0.0011f, 0.0012f, 0.0013f, 0.0014f, 0.0015f };
private float[] Kd = { 4.5f, 4.6f, 4.7f, 4.8f, 4.9f, 5f, 5.1f, 5.2f, 5.3f,
5.4f, 5.5f };
```

Це масиви, в яких знаходяться значення в околі знайдених попередньо коефіцієнтів. Після цього, для кожної з комбінацій проводились тести, робот на кожній ітерації проходив три кола і рахував суму помилок, а потім писав, на яких коефіцієнтах проводився тест та які він мав результати.

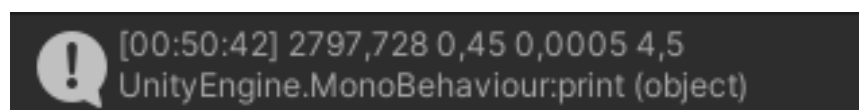


Рис 4.5 Перший тест з результатами

В кінці, коли робот закінчував всі тести, було демонстровано найкращій результат з найменшою сумою помилок:

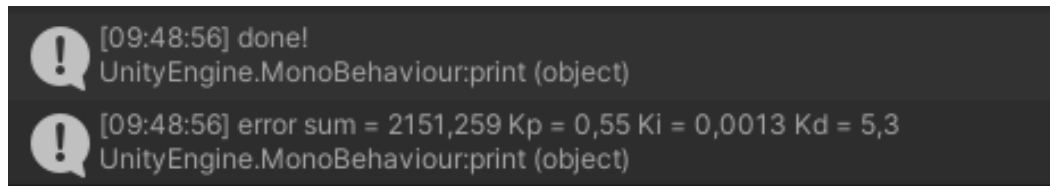


Рис 4.6 Результат

Дивлячись на останні два зображення, можна зробити декілька висновків: тести тривали майже дев'ять годин, що з одного боку багато, але швидше, ніж якщо використовувати інші методи автоматичного підбору; в результаті всі коефіцієнти збільшились; можна побачити велику різницю між сумою помилок в першому тесті і в результаті; і основне – результат відрізняється від того, що вийшов при методі евристичних експериментів.

Метод, що був описаний, має свої недоліки. В деяких ситуаціях такий час – це надто довго, тут використовувались людські потужності, не завжди має можливість проводити стільки тестів. Але в даній ситуації цей метод добре підходить завдяки своїм перевагам. Такий метод має перевагу над автоматичним підбором коефіцієнтів – він швидший та точніший. Над методом проб і помилок теж є перевага, фінальний результат відрізнявся від того, що був після ручного підбору, а це показує більшу точність описаного методу.

## 5. РЕЗУЛЬТАТ

Перш за все про конструкцію робота. Вона вийшла проста, це легко можна реалізувати як в віртуальному, так і у реальному світі. Його простота дозволяє не тільки легко застосувати до нього ПІД-регулятор, а й модифікувати та створити щось більш технологічне. В наступних розділах буде розказано про алгоритми пошуку шляху і цей робот, якщо до нього додати нові датчики, зможе виконувати ці задачі.

Коли обиралась система автоматичного управління, майже одразу був обраний пропорційно-інтегрально-диференціальний регулятор. ПІД є найбільш поширеною САУ, а це означає, що є багато матеріалу на цю тему. Крім того ця система є зрозумілою, її легко застосовувати та модифікувати. Саме через ці переваги був обраний цей контролер.

Хоч тема ПІД-регуляторів й не нова, дослідження все ще проводяться. До таких речей, як підбір коефіцієнтів, досі нема універсальних рішень. Запропоноване в цій роботі рішення також не є універсальним, але добре підходить саме в цій ситуації. При застосуванні відомих методів були свої недоліки і переваги, тому використання переваг кожного з методів дозволило створити свій спосіб підбору коефіцієнтів.

## ВИСНОВКИ

У цій роботі було досліджено актуальну на сьогодні тему, пов'язану з робототехнікою. Роботи-кур'єри активно розвиваються, тому було прийнято рішення бути частиною цього розвитку та запропонувати рішення, які можуть покращити роботів і зробити їх більш простими.

У процесі виконання роботи було досліджено різні методи, якими зараз користуються в робототехніці. У багатьох напрямках немає стандартних рішень, які б працювали в більшості умов, тому на кожен випадок потрібно робити щось нове або сильно модифікувати наявні рішення. Всі рішення проблем робились максимально простими і зрозумілими, щоб їх можна було адаптувати та модифікувати під різні проблеми.

Попри те, що рішення, які були описані в роботі, добре працюють та мають багато переваг в нашій ситуації, тема не може вважатись закритою. Простір для досліджень залишається великим, стандартних рішень багатьох проблем немає, тому потрібно над цим багато працювати, а основою для цього може слугувати ця робота.

## ДЖЕРЕЛА

1. Line Follower Using PID, [Електронний ресурс] <https://beginnertopro.in/post/331>
2. Пунов Є. А. Адаптивний пропорційно-інтегрально-диференціальний регулятор у роботизованих системах: магістерська дисертація [https://ela.kpi.ua/bitstream/123456789/40376/1/Punov%20\\_magistr.pdf](https://ela.kpi.ua/bitstream/123456789/40376/1/Punov%20_magistr.pdf)
3. Practical PID tuning guide, [Електронний ресурс] <https://tlk-energy.de/blog-en/practical-pid-tuning-guide>