

## ДОДАТОК А

## Лістинг програми симуляції

```
import {
  LineChart,
  Line,
  XAxis,
  YAxis,
  CartesianGrid,
  Tooltip,
  Legend,
  ResponsiveContainer,
  Area,
  AreaChart,
  ReferenceDot
} from "recharts";
import { useCallback, useEffect, useState } from "react";
import PID from "pid-controller";
const DirectionControl = () => {
  const CustomReferenceDot = props => {
    return (
      <circle cx={props.cx} r="10" cy={props.cy} fill="gold">
        <animate
          attributeName="r"
          from="8"
          to="20"
          dur="1.5s"
          begin="0s"
          repeatCount="indefinite"
        />
      </circle>
    );
  };
};
```

```

    />
  </circle>
);
};
const [stop, setStop] = useState(true);
const [direction, setDirection] = useState(1);
const [data, setData] = useState([]);
const [DirectionSetpoint, setDirectionSetpoint] = useState(21);
const upper = 0.001;
const lower = -0.0005;
const [Kp, setKp] = useState(800);
const [Ki, setKi] = useState(200);
const [Kd, setKd] = useState(50);
const [ctr, setCtr] = useState(
  new PID(direction, DirectionSetpoint, Kp, Ki, Kd, 'direct')
);
const timeframe = 1000;
useEffect(() => {
  ctr.setSampleTime(timeframe);
  ctr.setOutputLimits(0, 10000);
  ctr.setMode('auto');
  ctr.setPoint(DirectionSetpoint);
}, [ctr, DirectionSetpoint]);
const [timer, setTimer] = useState(0)
const DirectionSimulation = () => {
  ctr.setInput(direction);
  ctr.compute();
  const newDirection =
    direction + ctr.getOutput() * upper + (1500 - ctr.getOutput()) * lower;
  setTimer(prev => prev + 1)

```

```

setDirection(newDirection);
setData((prev) => {
  return [
    ...prev,
    {
      time: timer,
      uv: direction,
      pv: DirectionSetpoint,
    },
  ];
});
console.log(data);
console.log(`Output : ${ctr.getOutput()} ; Temp : ${Math.round(newDirection *
100) / 100}`);
};
const [isStopped, setIsStopped] = useState(true)
useEffect(() => {
  let intervalId
  if (isStopped === false)
    intervalId = setInterval(DirectionSimulation, timeframe);
  if (isStopped === true)
    clearInterval(intervalId)
  return () => clearInterval(intervalId);
}, [DirectionSimulation, timeframe]);
return (
  <div style={{ display: 'flex' }}>
    <div
      style={{
        height: "50vh",

```

```

    width: "1000px",
  }}
>
<ResponsiveContainer>
  <LineChart
    width={500}
    height={300}
    data={data}
    margin={{
      top: 5,
      right: 30,
      left: 20,
      bottom: 5,
    }}
  >
    <CartesianGrid strokeDasharray="3 3" />
    <XAxis dataKey="time" />
    <YAxis />
    <Tooltip />
    <Legend />
    <Line
      type="monotone"
      dataKey="pv"
      stroke="#8884d8"
      activeDot={{ r: 8 }}
    />
    <Line type="monotone" dataKey="uv" stroke="#82ca9d" />
  </LineChart>
</ResponsiveContainer>
</div>

```

```

<div
  style={{
    height: "50vh",
    width: "1000px",
  }}
>
  <ResponsiveContainer>
    <AreaChart
      width={500}
      height={600}
      data={data.map(({ pv, uv }) => ({ pv: [pv - 1, pv + 1], uv }))}
      margin={{
        top: 50,
        right: 30,
        left: 20,
        bottom: 5,
      }}
    >
      <XAxis dataKey="name" hide />
      <YAxis hide />
      <ReferenceDot shape={<CustomReferenceDot />} fill="#82ca9d"
stroke="#82ca9d" r={10} x={data.length - 3 || 0} y={data[data.length - 3]?.uv || 0}
style={{ zIndex: 100 }} />
      <Area dataKey='pv' fill="#8884d8" stroke="#8884d8" />
      <Area dataKey='uv' fill="transparent" stroke="transparent" />
    </AreaChart>
  </ResponsiveContainer>
</div>
</div>
<div

```

```
style={{
  display: "flex",
  flexDirection: "column",
  width: "200px",
}}
>
<input
  type="text"
  placeholder="DirectionSetpoint"
  onChange={(e) => setDirectionSetpoint(+e.target.value)}
/>
<input
  type="text"
  placeholder="Direction"
  onChange={(e) => setDirection(+e.target.value)}
/>
<input
  type="text"
  placeholder="Kp"
  onChange={(e) => setKp(+e.target.value)}
/>
<input
  type="text"
  placeholder="Ki"
  onChange={(e) => setKi(+e.target.value)}
/>
<input
  type="text"
  placeholder="Kd"
  onChange={(e) => setKd(+e.target.value)}
```

```
    />
    <button onClick={() => setIsStopped((prev) => !prev)}>
      {isStopped ? "Start" : "Stop"}
    </button>
  </div>
</>
);
};
export default DirectionControl;
const CustomReferenceDot = props => {
  return (
    <circle cx='5' r='10' cy='16' fill='gold'>
      <animate
        attributeName="r"
        from="8"
        to="20"
        dur="1.5s"
        begin="0s"
        repeatCount="indefinite"
      />
    </circle>
  );
};
```

**ДОДАТОК Б**  
**АПРОБАЦІЯ РЕЗУЛЬТАТІВ**

Міністерство освіти і науки України (МОНУ)  
Харківський національний університет радіоелектроніки (ХНУРЕ)  
Варшавський університет сільського господарства (WULS - SGGW)  
Азербайджанський державний університет нафти і промисловості  
Національний університет «Львівська політехніка»  
Festo Didactic Україна  
Jabil Circuit Ukraine Limited  
ТОВ «Науково-виробниче підприємство «УКРІНТЕХ»»  
Факультет автоматики і комп'ютеризованих технологій (АКТ)  
Кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки (КІТАР),  
Державне підприємство «Науково-дослідний технологічний інститут приладобудування»  
Державне підприємство «Південний державний проектно-конструкторський та науково-дослідний інститут авіаційної промисловості»

# МАТЕРІАЛИ

VII-ої Міжнародної Конференції  
**ВИРОБНИЦТВО**  
&  
**МЕХАТРОННІ СИСТЕМИ 2023**  
(19-20 жовтня 2023)  
Харків, Україна



## ЗМІСТ

|  |    |
|--|----|
| <i>Vladyslav Yevsieiev</i>   |    |
| Modeling of the BEAM robot control system on the basis of a microcircuit L293D .....   | 12 |
| <i>Medovkin Mykhailo, Puhach Hanna</i>   |    |
| The development of a cryptographically secure pseudorandom number generator .....  | 15 |
| <i>Svetlana Sotnik, Anton Andreiev</i>   |    |
| QR codes in production .....   | 19 |
| <i>Софія Хрустальова, Світлана Вишванюк</i>  |    |
| Розроблення структурної схеми модуля автоматизації на базі RFID – технологій .....   | 22 |
| <i>Владислав Заїкін</i>  |    |
| Моделювання пошуку вибухонебезпечних предметів методом електромагнітної спектроскопії та радіолокації .....                    | 26 |
| <i>Karetyna Stetsenko</i>  |    |
| BEAM Robotics: Combining Biological Principles and Technological Solutions for More Adaptive and Energy-Efficient Robots ..... | 30 |
| <i>Svitlana Maksymova, Mykhailo Akopov</i>   |    |
| Selection of Sensors for Building a 3D Model of the Mobile Robot's Environment .....   | 33 |
| <i>Сергій Новоселов, Єгор Волков</i>   |    |
| Завдання автоматичного керування рухом мобільної платформи з застосуванням законів автоматички .....                           | 36 |
| <i>Сергій Новоселов, Ігор Гладков</i>  |    |
| Сучасний промисловий інтернет речей та хмарні технології .....   | 40 |
| <i>Дмитро Гурін</i>  |    |
| Вирішення задачі зворотної кінематики для рухливих кінцівок роботехнічної платформи .....                                      | 43 |

# Завдання автоматичного керування рухом мобільної платформи з застосуванням законів автоматичності

Сергій Новоселов, Єгор Волков

Кафедра КІТАР, Харківський національний університет радіоелектроніки, УКРАЇНА, Харків, пр. Науки 14, email: yehor.volkov@nure.ua

**Анотація:** В роботі розібрано основні завдання автономного пересування роботів, розглянуто оптимізація швидкості руху на заданих маршрутах, описана САК та описано ПДД-регулятор.

**Ключові слова:** ПДД-регулятор, закони автоматичності, керування рухом, робототехніка.

## I. ВСТУП

На даний момент тема автоматичного керування є дуже актуальною, так як все більше і більше в різних галузях починають використовуватися роботи, від стандартних роботів маніпуляторів до цілих підприємств, які повністю управляються та функціонують за рахунок різноманітних роботів, а для їхнього руху потрібно розробити певну послідовність дій.

Усі роботи рухаються по заданим координатам, які мають бути дуже точно прокладені. Адже якщо маніпулятор відхилиться від заданого маршруту на один градус, він вже опиниться в абсолютно не тому місці, в якому мав опинитися і це спричинить багато наслідків, найкритичніше з усіх буде брак.

Автоматичні машини не змогли б правильно функціонувати без налагодженої системи автоматичного пересування та керування, адже будь-яка помилка в їхньому шляху чи позиціонуванні координат призведе до тотальної дезорієнтації платформи.

## II. СИСТЕМА АВТОМАТИЧНОГО КЕРУВАННЯ (САК)

Автоматична система керування – це сукупність керованого об'єкта й автоматичних вимірвальних та керуючих пристроїв.

На основі автоматизованої системи управління ця система самодіюча і реалізує встановлені функції процесу автоматично, без участі людини (крім етапів запуску та налагодження системи). На практиці часто користуються терміном-аналогом системи автоматичного керування (САК).

Автоматичним регулюванням використовуємо підтримку на заданому рівні певної фізичної (хімічної) величини, що характеризує процес, або змінює її згідно із заданим законом.

Автоматичне керування – ширше поняття, в цьому випадку створюється повна кількість впливів на процес, вибраних з певної кількості можливих.

Автоматичні системи керування можна класифікувати за швидкими ознаками:

- інформативним принципом;
- кількість керованих параметрів і контурів;

- виглядом статичних і динамічних характеристик;
- структурними особливостями тощо.

Одним із розширених принципів класифікації автоматизованих систем керування на сьогодні при бурхливому розвитку інформаційних технологій є інформативний, на основі якого лежать особливості здобуття та використання інформації. відповідно до цього принципу виділяють систему: з повною і з неповною початковою інформацією. Перші відповідно звичайними, вони мають початкову інформацію, достатню для розв'язання поставленого завдання на період усього часу роботи системи. З неповною початковою інформацією або кібернетичними є системи, які для розв'язання поставлених завдань повинні в процесі роботи отримати додаткову інформацію, аналіз якої дає змогу сформувати необхідні команди керування. Системи водопостачання та водовідведення як об'єкти автоматизації від цих до звичайних систем з повною початковою інформацією. Їх можна розділити на дві характерні групи, що відрізняються як принципом керування, так і особливостями функціонування. Перша – це замкнуті автоматичні системи керування відносно вихідної (керованої) величини, що діють на основі принципу керування за відхиленням керованої величини, їх буде використовуватися системами із зворотним зв'язком. Друга – розімкнуті автоматичні системи керування окремою величиною, що базуються на принципі керування за збуренням.

Одним з параметрів САК можна виділити автоматичну у параметрах руху. Закони автоматичності відіграють важливу роль у регулюванні параметрів руху в різних системах. Вони використовуються для створення автоматичних керуючих систем, які здатні підтримувати або регулювати різні параметри руху, такі як швидкість, положення, кут, траєкторія та інші.

Розглянемо які існують завдання автономного пересування.

## III. ЗАВДАННЯ АВТОНОМНОГО ПЕРЕСУВАННЯ

Їх існує дуже багато, деякі з них приведено і описано нижче:

Навігація – однією з основних завдань є розробка ефективних алгоритмів навігації, які дозволяють переміщатися по заданій території, уникаючи перешкод і безпечно досягаючи цілей. Це включає в себе розробку алгоритмів виявлення перешкод, маршрутизації та планування руху.

Далі можна виділити датчики та сприйняття. Для успішної навігації, роботу потрібно бути обладнаним

різними датчиками, такими як лідари, камери, ультразвукові датчики та інерціальні вимірювачі, щоб надійно сприймати оточуюче середовище та стан самої платформи.

Навчання з підкріпленням – застосування методів машинного навчання, таких як навчання з підкріпленням, для поліпшення управління мобільною платформою. Це дозволяє машині вчитися на основі досвіду та вдосконалювати свої дії відповідно до поставлених цілей.

Також до завдань автономного пересування відноситься стійкість до змінних умов. Роботи можуть стикатися з різними умовами, такими як зміни погоди, освітлення та стан дороги. Вирішення проблеми включає в себе створення систем, здатних адаптуватися до змінних умов та зберігати стабільність в русі.

Ще можемо виділити такий параметр як безпека. Забезпечення безпеки під час руху мобільної платформи є пріоритетною задачею. Це включає в себе розробку систем запобігання зіткненням і безпечного гальмування, а також механізмів виявлення людей та інших об'єктів.

Останнім параметром який треба виділити є ефективність. Максимізація ефективності руху платформи, зменшення часу в дорозі та оптимізація споживання енергії також є важливими аспектами проблеми.

Вирішення цих завдань потребує дуже чіткого підходу, включаючи розробку апаратних та програмних компонентів, а також поєднання технологій штучного інтелекту та робототехніки.

Розглянемо як застосовувати закони автоматички для регулювання швидкості.

#### IV. РЕГУЛЮВАННЯ ШВИДКОСТІ

За допомогою законів автоматички можна створити системи управління, які підтримують задану швидкість руху. Прикладом цього буде адаптивний круїз-контроль у машинах. Розглянемо як він влаштований та працює.

Адаптивний круїз-контроль - це покращена система круїз-контролю, яка не тільки автоматично підтримує швидкість руху, але і дотримується безпечної дистанції до автомобіля, що йде попереду, адаптуючись під поточну ситуацію на дорозі.

На відміну від неактивної системи, яка лише підтримує задану швидкість руху, система адаптивного круїз-контролю:

- прискорює автомобіль до заданого значення;
- уповільнює його, аж до повної зупинки;
- утримує задану швидкість, забезпечуючи не тільки комфорт у тривалих поїздах, а й значно на 10% і більше скорочуючи витрату палива.

Сама система складається з трьох вузлів.

По-перше, це фронтальні датчики, розташовані в передньому бампері, рідше – за фальш решіткою радіатора. Це – «очі» круїз-контролю, тому мають бути чистими від бруду та снігу, інакше ефективність системи прагнути до нуля.

По-друге, це додаткові датчики, що відстежують, чи їде автомобіль по прямій, в гірку або з неї, на який

кут відхилене рульове колесо, що контролюють включену в КПП передачу, навіть інформацію з дорожньої розмітки та придорожніх знаків.

По-третє, це електронний блок управління адаптивного круїз-контролю. Від нього залежить як якість функціонування, та й робота взагалі. Він аналізує інформацію, що надходить, даючи команди:

- системі подачі палива змінити кут нахилу дросельної заслінки;
- гальмівному контуру збільшити або навпаки послабити силу притискання гальмівних колодок до дисків.

Принцип роботи адаптивного круїз-контролю:

- збирається вся інформація з основних та другорядних датчиків;
- блок управління у режимі реального часу безперервно її аналізує;
- залежно від заданої водієм швидкості та ситуації на дорозі – порожня, пряма траса або, навпаки, авто, що йде попереду, пригальмовує/перебудовується, дає команду на прискорення або, навпаки, уповільнення. При цьому система керування може гальмувати двигуном або зменшувати його потужність шляхом короткочасного збіднення повітряно-паливної суміші.

Також регулювання швидкості, тісно пов'язане з проходженням траєкторії руху мобільної платформи. Вона дозволить точніше слідувати побудованому маршруту, що підвищить оптимізацію нашого руху.

Роботи і автономні транспортні засоби використовують закони автоматички для слідування заздалегідь визначеним траєкторіям. Це може включати в себе коригування траєкторії на основі зворотного зв'язку від датчиків і камер. Розглянемо використання ПД-регулятора для відстеження траєкторії роботом, але перед цим розберемося як він працює, а також де він може застосовуватися.

#### V. ПД-РЕГУЛЯТОР

Пропорційно-інтегрально-диференціальний (ПД) регулятор – пристрій в керуючому контурі зі зворотним зв'язком. Використовується в системах автоматичного управління для формування сигналу керуючого з метою отримання необхідних точності і якості перехідного процесу. ПД-регулятор формує керуючий сигнал, що є сумою трьох доданків, перший з яких є пропорційним різниці вхідного сигналу і сигналу зворотного зв'язку (сигнал неузгодженості), друге - інтегралу сигналу неузгодженості, третє - похідної сигналу неузгодженості.

ПД регулятор – один із найпоширеніших автоматичних регуляторів. Він настільки універсальний, що застосовується практично скрізь, де потрібне автоматичне керування.

ПД регулятор складається з трьох складових (рисунк 1): пропорційної  $P$ , що інтегрує  $I$  і диференціє  $D$ , формується просто як сума трьох значень, помножених кожна на свій коефіцієнт. Ця сума після обчислень стає керуючим сигналом, який подається на керуючий пристрій.

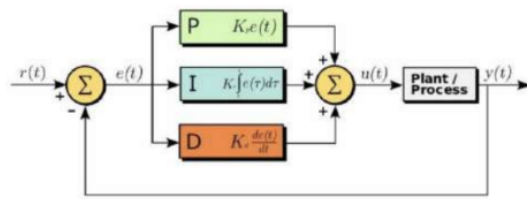


Рис. 1. Схема принципу роботи ПІД-регулятора

$K_P$ ,  $K_I$  і  $K_D$  це і є ті коефіцієнти, які потрібно налаштувати для роботи ПІД-регулятора. Значення тут можуть бути різні, це залежить від конкретної системи. Будь-який коефіцієнт може дорівнювати нулю, і в такому випадку анулюється вся його компонента. Тобто регулятор можна перетворити на П, І, Д, та інші поєднання. Різні системи вимагають різного підходу, саме тому ПІД-регулятор такий універсальний.

Пропорційна складова є різницею поточного значення з датчика та установки. Ця різниця називається помилкою регулювання, тобто як далеко знаходиться система від заданого значення. Виходить чим більше помилка, тим більше буде керуючий сигнал і тим швидше система наводитиме керовану величину до заданого значення. П складова є основною в ПІД-регуляторі, регулятор може непогано працювати тільки на ній одній. Р складова виправляє помилку в даний час.

Інтегральна складова просто підсумовує в саму себе ту саму помилку, різницю поточного і заданого значення, помножену на період дискретизації системи, тобто на час, що минув з попереднього розрахунку  $dt$  - фактично бере інтеграл від помилки за часом. У самому регуляторі це ще множить на коефіцієнт  $K_I$ , яким налаштовується різкість цієї складової. В інтегральній складовій буквально накопичується помилка, що дозволяє регулятору з часом повністю її усунути, тобто привести систему до заданого значення з максимальною точністю. І складова виправляє минулі помилки, що накопичилися.

Диференціальна складова являє собою різницю поточної та попередньої помилки, поділену на час між вимірюваннями, тобто на ту ж  $dt$ , яка має загальний період регулятора. Іншими словами – це похідна від помилки за часом. Фактично Д складова реагує на зміну сигналу з датчика, і чим сильніше відбувається ця зміна, тим більше значення додається до загальної суми. Д дозволяє компенсувати різкі зміни в системі і при правильному налаштуванні запобігти сильному перерегулюванню та зменшити розгойдування. Коефіцієнт Д дозволяє налаштувати вагу, чи різкість даної компенсації, як та інші коефіцієнти регулюють свої складові. Д складова в першу чергу потрібна для швидких систем, тобто для систем із різкими змінами. Д складова виправляє можливі майбутні помилки, аналізуючи швидкість.

Для налаштування регулятора потрібно варіювати коефіцієнти:

– при збільшенні  $K_P$  збільшується швидкість виходу на встановлене значення, збільшується

сигнал, що управляє. Чисто математично система не може прийти рівно до заданого значення, так як при наблизенні до установки П складник пропорційно зменшується. При подальшому збільшенні  $K_P$  реальна система втрачає стійкість та починаються коливання.

– при збільшенні  $K_I$  зростає швидкість компенсації помилки, що накопичилася, що дозволяє вивести систему точно до заданого значення з плином часу. Якщо система повільна, а  $K_I$  занадто великий - інтегральна сума сильно зростає і відбудеться перерегулювання, яке може мати характер коливань, що не затухають, з великим періодом. Тому інтегральну суму в алгоритмі регулятора часто обмежують, щоб вона не могла збільшуватися та зменшуватися до нескінченності.

– при збільшенні  $K_D$  зростає стабільність системи, вона дає системі змінюватися дуже швидко. У той же час  $K_D$  може стати причиною неадекватної поведінки системи та постійних стрибків сигналу керуючого, якщо значення з датчика вагається. На кожну різку зміну сигналу датчика Д складова буде реагувати зміню керуючого сигналу, тому сигнал з датчика потрібно фільтрувати.

## VI ЗАСТОСУВАННЯ ПІД-РЕГУЛЯТОРУ У РОБОТОТЕХНІЦІ

ПІД – це алгоритм керування, який знаходиться в електричному приводі та секції керування мобільної платформи. Для автоматичної навігації та управління зазвичай існує три контури керування: крутний момент, швидкість і положення. Як правило, привід керує контуром крутного моменту, а «контролер» керує швидкістю та положенням.

Погане налаштування може мати низку шкідливих наслідків для програми рухомої платформи, найочевиднішим з яких є те, що система не стабільна чи точна. Погано налаштована система може задавати невірний маршрут і невірно регулювати швидкість руху автоматизованого роботу. Окрім неточності, погане налаштування може призвести до поломки машини, а також до перегріву та пошкодження двигунів і приводів.

Справлятися з мінливими навантаженнями є дуже складним завданням для мобільної платформи. Правильне налаштування важливе, але в екстремальній ситуації може знадобитися динамічна налаштування, щоб впоратися з цією проблемою.

## VII. ВИСНОВКИ

В даному матеріалі були висвітлені завдання, що виникають при автоматичному управлінні рухом автоматичної платформ, описано САК та її ознаки, розглянуто ПІД-регулятор та як він працює, а також його застосування у робототехніці.

## ПЕРЕЛІК ПОСИЛАНЬ

- [1] AplusTopper [Електронний ресурс]: Automation Advantages And Disadvantages | What are The Top Advantages and Disadvantages of Automation? – Режим доступу:

- <https://www.aplustopper.com/automation-advantages-and-disadvantages/>
- [2] В.О. Клебан, В.Г. Парфьонов, А.А. Шалито,  
«ПОБУДУВАННЯ СИСТЕМИ  
АВТОМАТИЧНОГО  
УПРАВЛІННЯ МОБІЛЬНИМ РОБОТОМ НА  
ОСНОВІ АВТОМАТНОГО ПІДХОДУ», Жовтень  
2017.
- [3] Crenganis, M., Bologa, O., Developing a mobile  
platform for EUROBOT 2015, Buletinul AGIR.  
Supliment, Vol. 1, pp. 230-235, 2015.
- [4] Tan, K. K., Wang, Q. G., Hang, C. C., Advances in  
PID control, Springer Science & Business Media,  
(2012).

**ДОДАТОК В**  
**ДЕМОНСТРАЦІЙНИЙ МАТЕРІАЛ**

