

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний інститут ім.  
Ігоря Сікорського»  
Навчально-науковий комплекс «Інститут прикладного та системного аналізу»  
Кафедра математичного моделювання системного аналізу

Частина комплексної дипломної роботи на тему:

Модуль керування автоматизованої системи  
переслідування об'єкта на базі безпілотного літального  
апарата

Виконав студент групи КА-33  
Мерзляков О.Д.  
Науковий керівник:  
к.т.н Дідковська М.В.

# Об'єкт, предмет і мета дослідження

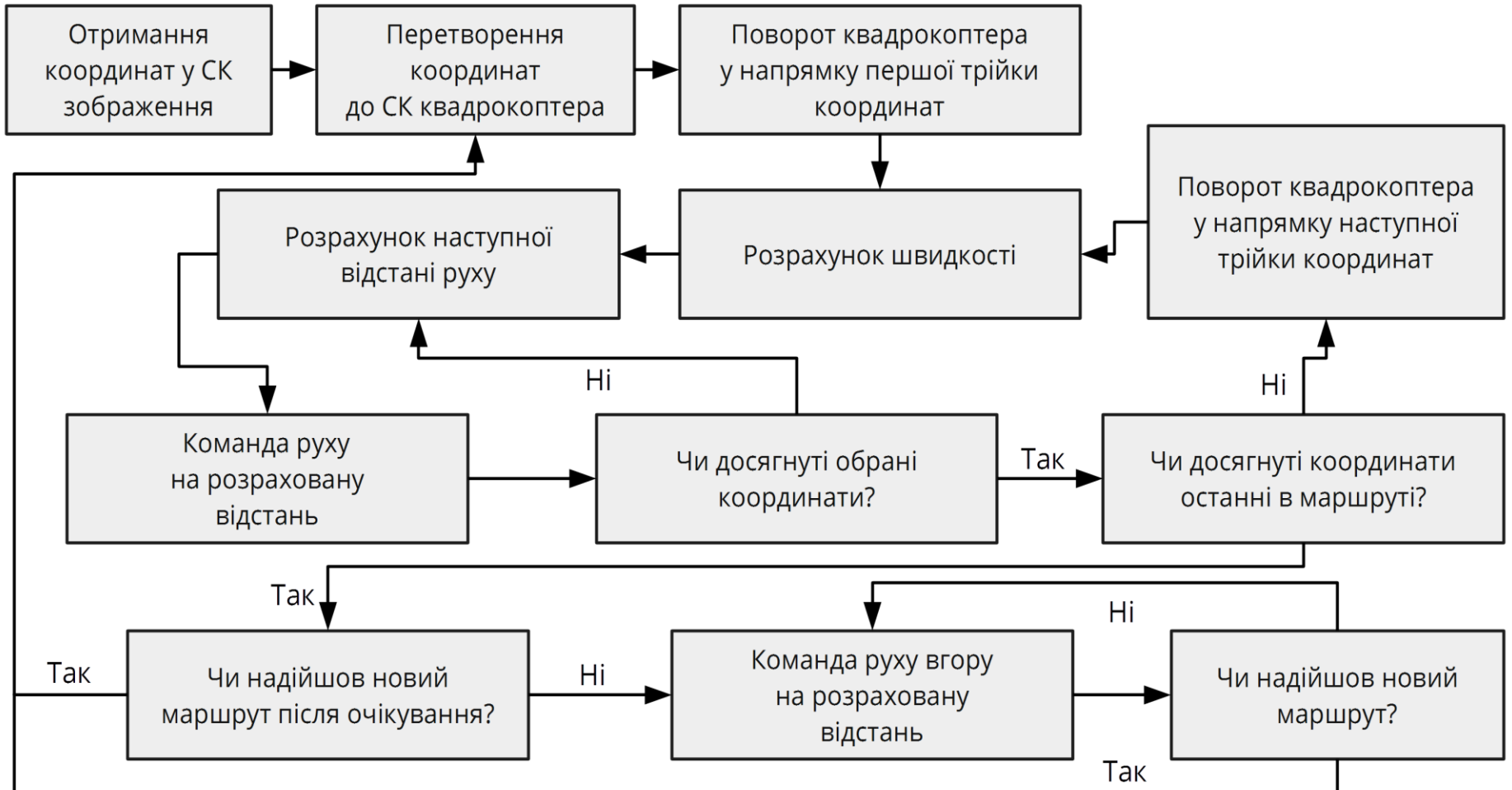
**Предметом** роботи є система керування безпілотним літальним апаратом.

**Мета** роботи – дослідити існуючі системи керування безпілотними літальними апаратами; запропонувати алгоритм керування квадрокоптером; написати програмний код, що буде виконувати керування.

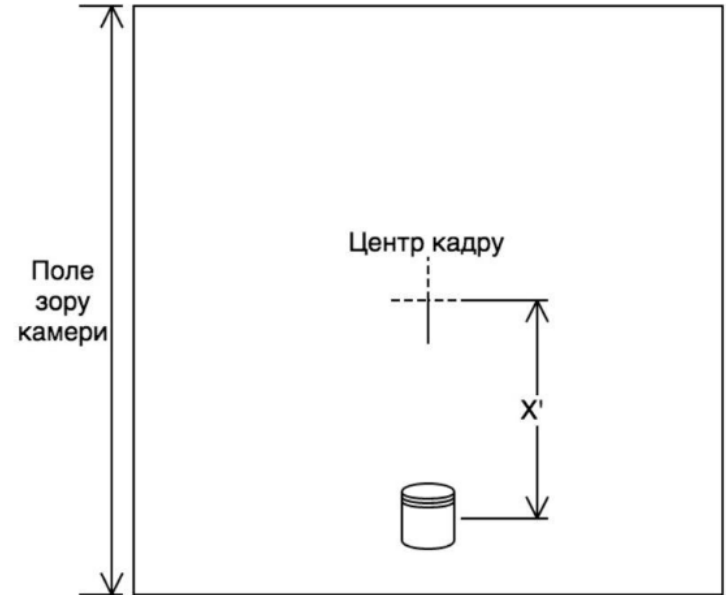
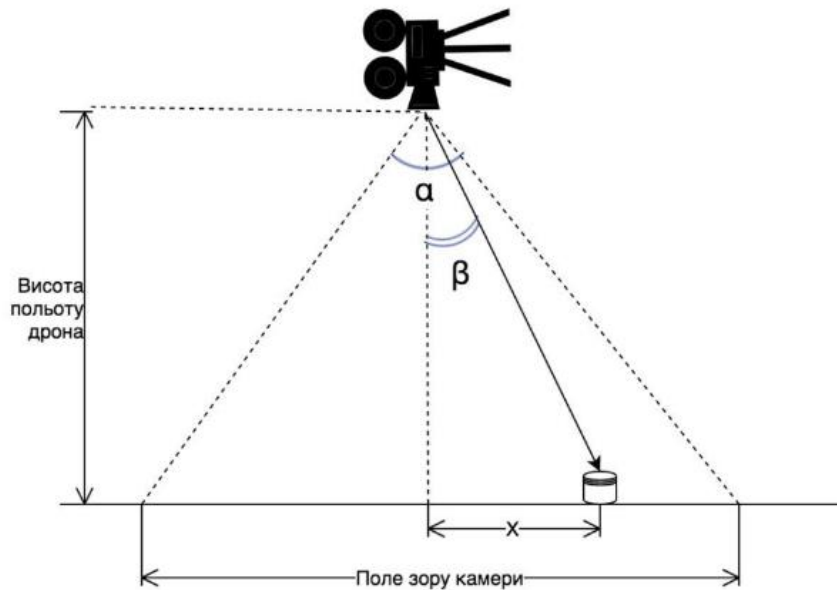
# Постановка задачі

- Проаналізувати існуючі підходи, алгоритми та методи керування безпілотними літальними апаратами
- Запропонувати алгоритм керування квадрокоптером
- Розробити схему системи керування віртуальним та реальним квадрокоптером
- Розробити програмний продукт, що буде взаємодіяти з системою розпізнавання і буде здатний керувати як реальним так і віртуальним квадрокоптером

# Блок-схема алгоритму



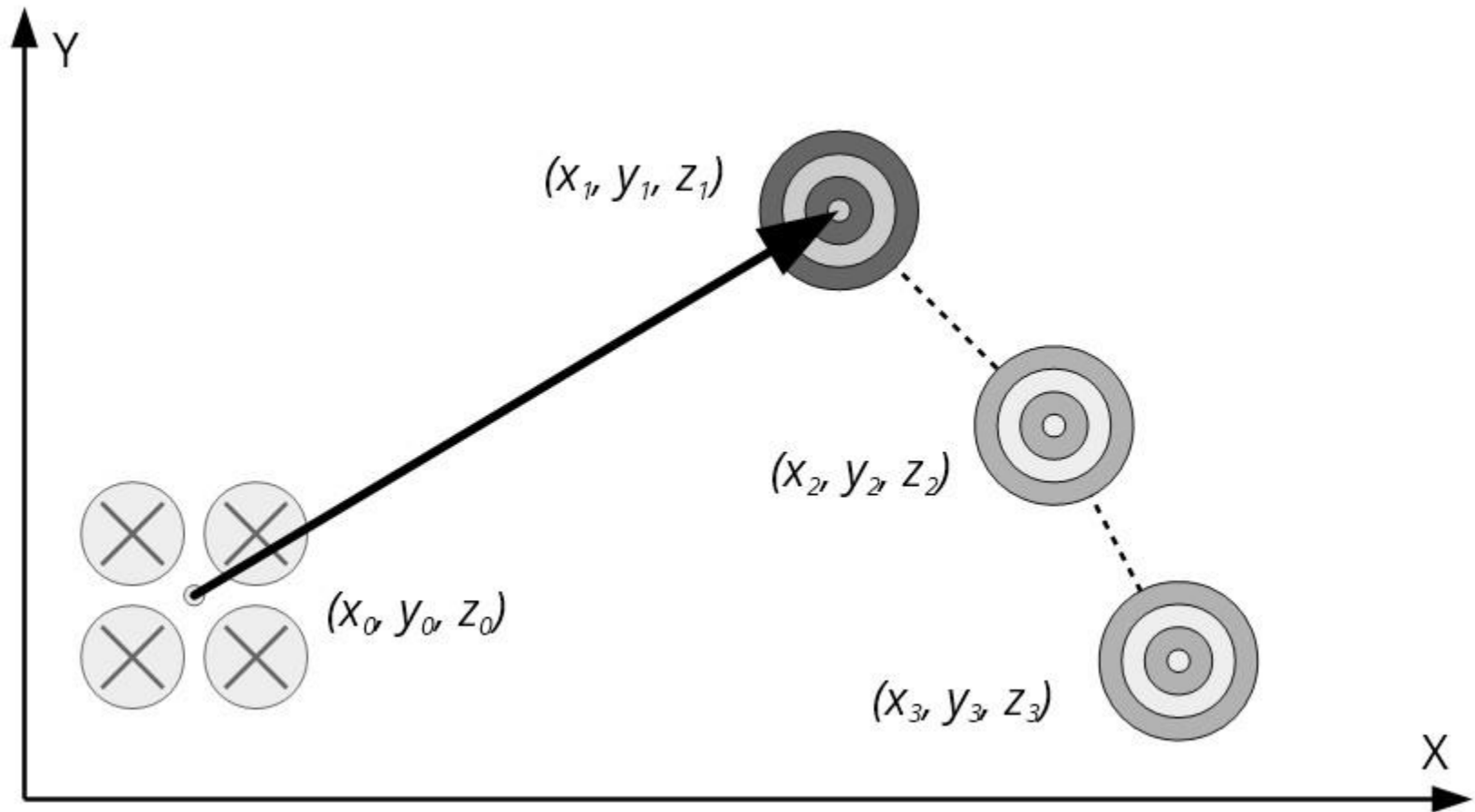
# Переведення координат між системами



$$x' = \frac{x \cdot \frac{x'_{\max}}{2}}{h \cdot \operatorname{tg}\left(\frac{\alpha}{2}\right)}$$

$$x = \frac{x' \cdot h \cdot \operatorname{tg}\left(\frac{\alpha}{2}\right)}{\frac{x'_{\max}}{2}}$$

# Вектор до цілі



# Розрахунок руху

$$v = \begin{cases} v_{max} \cdot \frac{l}{L}, l < L \\ v_{max}, l > L \end{cases}$$

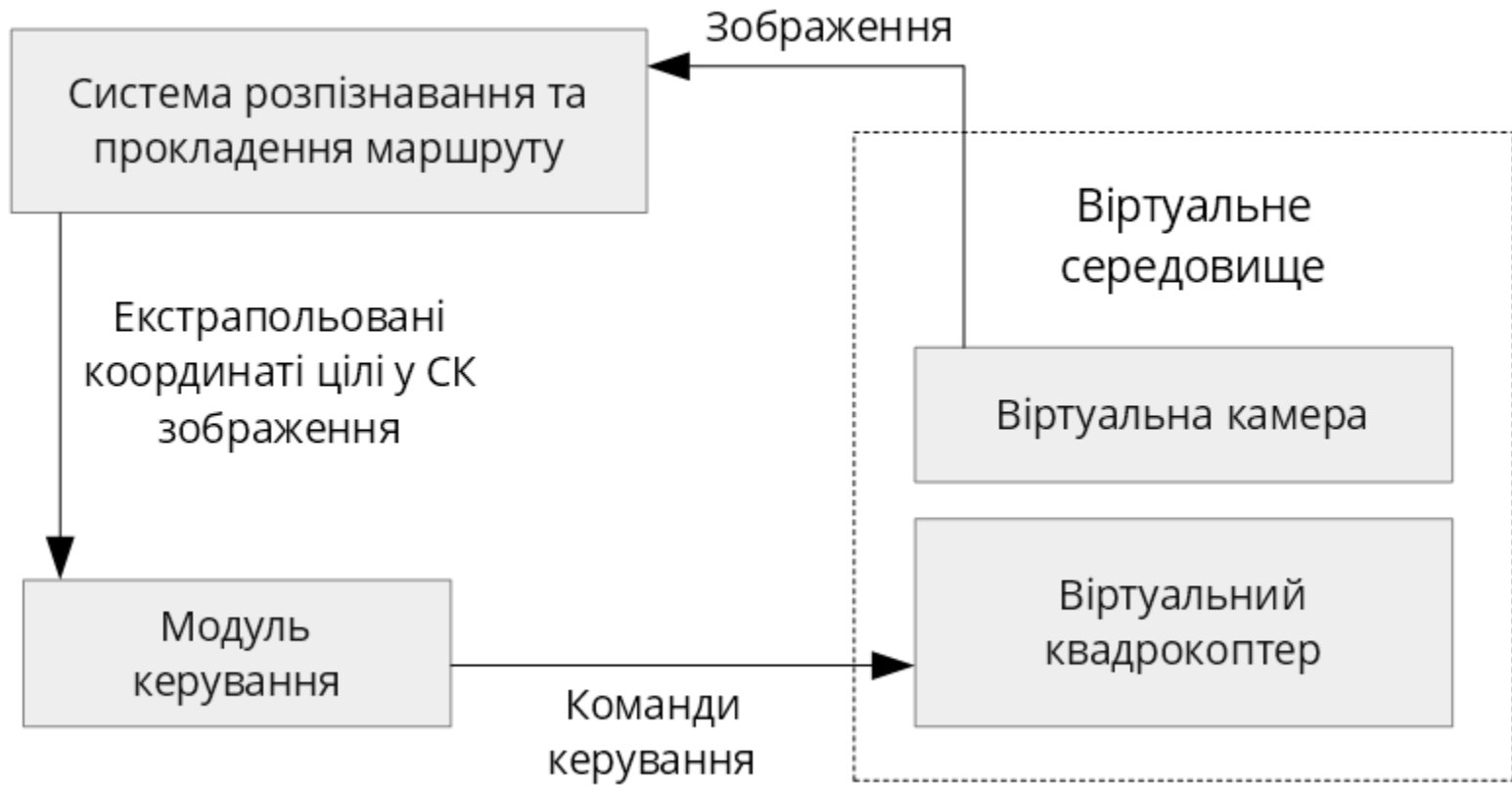
$l$  – довжина вектора до цілі,  
 $L$  – мінімальна довжина, при переміщенні на яку квадрокоптер рухається на максимальній швидкості,  
 $v_{max}$  – максимальна швидкість.

$$v = v_0 + at$$

$v_0$  – швидкість квадрокоптера на минулому кроці

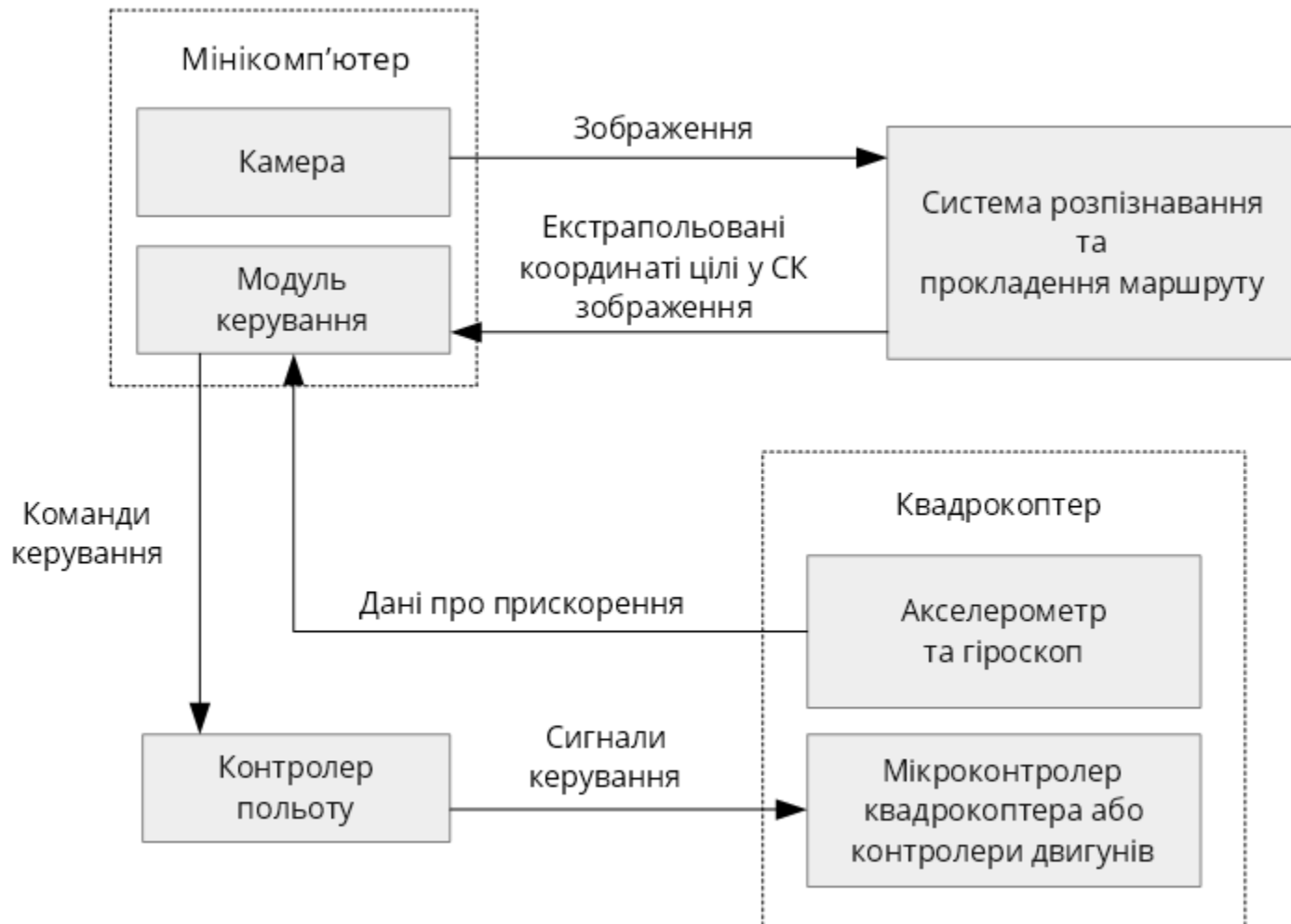
$$d = vt$$

# Принципова схема у випадку керування віртуальним квадрокоптером





# Принципова схема у випадку керування реальним квадрокоптером





# Результати роботи

The screenshot displays the V-REP PRO EDU interface. The main 3D view shows a simulated environment with a floor, a table, a chair, and a potted plant. A target (bullseye) is placed on the floor. A blue line indicates the path of a quadcopter, starting from a starting point and moving towards the target. The interface includes a top toolbar with various simulation controls, a left sidebar with a scene hierarchy, and a bottom terminal window.

**Scene Hierarchy:**

- pyquadsim (scene 1)
  - Camera
  - DefaultCamera
  - DefaultLightA
  - DefaultLightB
  - DefaultLightC
  - diningChair
  - diningTable
  - diningTable0
  - indoorPlant
  - Path

**Terminal Output:**

```
real=(0.84, -0.90, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01)
real=(0.85, -0.92, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01)
real=(0.87, -0.94, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01)
real=(0.88, -0.95, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01)
real=(0.90, -0.98, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01)
real=(0.91, -0.99, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01)
real=(0.93, -1.02, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01)
real=(0.95, -1.03, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01)
real=(0.96, -1.06, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01)
real=(0.98, -1.07, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01)
real=(1.00, -1.09, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01)
real=(1.01, -1.11, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01)
real=(1.03, -1.13, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01)
real=(1.04, -1.15, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01)
real=(1.06, -1.17, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01)
real=(1.07, -1.18, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01)
real=(1.09, -1.21, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01)
real=(1.11, -1.22, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01)
real=(1.13, -1.25, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01)
real=(1.14, -1.26, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01)
real=(1.16, -1.28, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01)
real=(1.17, -1.30, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01)
real=(1.19, -1.32, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01)
```

**Terminal Status:** Raven — java — sbt run — 80x24

**Bottom Bar:** ABB IRB 140.ttm [string "SCRIPT Quadricopter"]:195: in main chunk



# Результати роботи

The image shows a V-REP PRO EDU simulation window titled "pyquadsim". The main 3D view displays a dining room environment with a yellow table, a chair, and a potted plant. A quadcopter robot is positioned above the table, with a blue path indicating its trajectory. A target symbol is visible on the floor. The interface includes a top toolbar with navigation and simulation controls, a left sidebar with a scene hierarchy, and a bottom-left data log window.

Scene hierarchy:

- pyquadsim (scene 1)
  - Camera
    - DefaultCamera
    - DefaultLightA
    - DefaultLightB
    - DefaultLightC
  - diningChair
  - diningTable
  - diningTable0
  - indoorPlant
  - Path

Data log (Raven - java - sbt run - 80x24):

```
real=(6.79, -0.14, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(6.80, -0.12, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(6.82, -0.10, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(6.83, -0.08, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(6.85, -0.06, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(6.86, -0.05, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(6.88, -0.02, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(6.89, -0.01, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(6.91, 0.02, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(6.92, 0.03, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(6.94, 0.06, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(6.95, 0.07, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(6.97, 0.10, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(6.98, 0.11, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.00, 0.14, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(7.01, 0.16, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.03, 0.18, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(7.04, 0.20, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.06, 0.22, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(7.07, 0.24, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.08, 0.26, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(7.10, 0.28, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.11, 0.30, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
```

ABB IRB 140.ttm [string "SCRIPT Quadcopter"]:195: in main chunk

# Результати роботи

V-REP PRO EDU - pyquadsim - rendering: 8 ms (166.7 fps) - SIMULATION RUNNING

pyquadsim

Scene hierarchy

- pyquadsim (scene 1)
- Camera
  - DefaultCamera
- DefaultLightA
- DefaultLightB
- DefaultLightC
- diningChair
- diningTable
- diningTable0
- indoorPlant
- Path

```
real=(7.15, 1.43, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.15, 1.46, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(7.15, 1.48, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.15, 1.51, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(7.15, 1.53, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.15, 1.56, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(7.15, 1.58, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.14, 1.61, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(7.14, 1.63, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.14, 1.66, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(7.14, 1.68, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.14, 1.71, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(7.14, 1.73, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.13, 1.76, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(7.13, 1.78, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.13, 1.81, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(7.13, 1.83, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.13, 1.86, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(7.13, 1.88, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.13, 1.91, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(7.13, 1.93, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
real=(7.12, 1.96, 3.73) horizVel=0.02, vertiVel=0.00 acceleration=0.01
real=(7.12, 1.98, 3.73) horizVel=0.03, vertiVel=0.00 acceleration=-0.01
```

ABB IRB 140.ttm [string "SCRIPT Quadcopter"]:195: in main chunk

# Аналіз результатів

- Віртуальне середовище гарно себе показало як засіб для попередніх випробувань та навіть як засіб для повноцінної розробки
- Розроблений алгоритм керування показав себе гарно, керуючі квадрокоптером достатньо плавно і без аномалій (на кшталт ривків) під час слідування за ціллю та її пошуку
- Через невелику швидкість цілі і незначного недоліка віртуального середовища, квадрокоптер не може в точності вийти на швидкість цілі у конкретному прикладі

# Висновки

- Були проаналізовані існуючі підходи, алгоритми та методи керування квадрокоптерами та іншими безпілотними літальними апаратами
- Був запропонований алгоритм для модуля керування квадрокоптером. Було розроблено дві схеми системи керування безпілотним літальним апаратом: для реального та віртуального квадрокоптера.
- Був розроблений програмний продукт, що взаємодіє з системою розпізнавання середовища і прокладення маршруту і виконаний як абстрагований інтерфейс, що здатний керувати як реальним так і віртуальним літальним апаратом.

## Подальші шляхи розвитку

- Додати можливість повертати квадрокоптер. Це дозволить повертатися у сторону цілі. Однак, для цього буде необхідно додатково переробити перевід координат з однієї системи в іншу.
- Враховувати кут нахилу квадрокоптера під час польоту на великих швидкостях, що підвищить маневреність.
- Додати можливість змінювати прискорення, що дозволить збільшити плавність польоту при різких змінах відстані між точками маршруту.