КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра системного аналізу та теорії прийняття рішень

Кваліфікаційна робота
на здобуття ступеня бакалавра
за спеціальністю 124 Системний аналіз
на тему:

ПОБУДОВА ТА АНАЛІЗ МОДЕЛІ ДЛЯ ОПТИМАЛЬНОГО ПЕРЕХОПЛЕННЯ ТА УРАЖЕННЯ ДИНАМІЧНИХ ОБ'ЄКТІВ

Виконав студент 4-го курсу
Крутов Василь Васильович

(підпис)

Науковий керівник:
доцент, кандидат фізико-математичних наук
Зінько Петро Миколайович

(підпис)

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент

(підпис)

Роботу розглянуто й допущено до захисту на засіданні кафедри системного аналізу та теорії прийняття рішень
«01» червня 2023р.,
протокол № 13
Завідувач кафедри
Наконечний О. Г.

(підпис)

Київ – 2023

Зміст

РЕФЕРАТ **4**

ВСТУП **5**

Актуальність проблематики 5

Мета й завдання роботи 6

РОЗДІЛ 1
ЗАСОБИ ПЕРЕХОПЛЕННЯ ТА УРАЖЕННЯ ДИНАМІЧНИХ ОБ'ЄКТІВ **8**

1.1 Загальні підходи до перехоплення повітряних цілей 8

1.2 Різновиди засобів перехоплення та ураження повітряних цілей 9

1.3 Конструкція ракети протиповітряної оборони 10

1.4 Система наведення ракети протиповітряної оборони 12

РОЗДІЛ 2
ДИФЕРЕНЦІАЛЬНІ ІГРИ **14**

2.1 Диференціальні ігри як різновид конфліктно-керованих процесів 14

2.2 Загальні підходи до розв'язку диференціальних ігор 15

РОЗДІЛ 3
МАТЕМАТИЧНА МОДЕЛЬ ГРИ ПЕРЕСЛІДУВАННЯ **16**

3.1 Опис гри переслідування 16

3.1.1 Описання простору гри 16

3.1.2 Математична модель руху переслідувача 17

3.1.3 Математична модель руху втікача 18

3.1.4 Основне рівняння гри 19

3.2 Формулювання та розв'язок задачі переслідування 20

3.2.1 Формулювання задачі 20

3.2.2 Регресивний підхід до розв'язку 20

3.2.3 Знаходження ціни гри 23

3.2.4 Знаходження оптимального керування та стратегії . . . 28

РОЗДІЛ 4
АНАЛІЗ РЕЗУЛЬТАТІВ ГРИ **31**
4.1 Моделювання різних початкових моментів гри 31
4.2 Аналіз результатів проведених ігор 40

ВИСНОВКИ **42**

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ **44**

ДОДАТОК А **45**
Код програми на Python 45
Код програми на R 55

РЕФЕРАТ

Обсяг роботи 60 сторінок, 19 ілюстрацій, 7 джерел посилань.

ВТІКАЧ, ГРА ПЕРЕСЛІДУВАННЯ, ДИФЕРЕНЦІАЛЬНА ГРА, ОПТИМАЛЬНА СТРАТЕГІЯ, ОПТИМАЛЬНА ТРАЄКТОРІЯ, ПЕРЕСЛІДУВАЧ, ПОВІТРЯНА ЦІЛЬ, РАКЕТА ПРОТИПОВІТРЯНОЇ ОБОРОНИ, СИСТЕМА НАВЕДЕННЯ, УПЕРЕДЖЕННЯ, ЦІНА ГРИ.

Об'єктом роботи є процес розв'язування диференціальної гри переслідування для двох гравців із застосуванням програмного коду.

Предметом роботи є математична модель, що описує диференціальну гру переслідування, та знайдені для неї оптимальні стратегії гравців та ціна гри. Предметом роботи також є програмний застосунок, що допомагає в пошуку розв'язку, а також моделює різні початкові ситуації диференціальної гри та візуалізує її результати.

Метою роботи є побудова математичної моделі, що описує диференціальну гру переслідування, розв'язок гри та знаходження оптимальних стратегій для гравців і ціни гри, написання програмного коду, що допомагатиме у пошуку розв'язку для диференціальної гри, написання програмного коду, що моделюватиме динаміку гри та візуалізуватиме результати диференціальних ігор при різних початкових даних.

Методи дослідження: методи опису моделей простого та складного рухів, методи та підходи до розв'язку конфліктно керованих процесів та диференціальних ігор, чисельні методи для пошуку коренів трансцендентних рівнянь. Інструменти розроблення: програмне середовище розробки PyCharm IDE 2023.1.1, мова програмування Python; програмне середовище розробки RStudio 4.3.1, мова програмування R.

Результати роботи: побудована математична модель, що враховує вплив сили тертя, сили тяжіння та сили тягу на переслідувача, а також описує простий рух для втікача; розв'язана диференціальна гра та знайдена ціна гри та оптимальні стратегії для гравців; змодельовано шість ігор при різних початкових станах системи та параметрах; продемонстровані результати та наведений поверхневий аналіз змодельованих ігор.

За методами розробки та інструментальними засобами робота виконувалася сумісно з роботами з побудови, дослідження та розв'язку конфліктно керованих процесів та диференціальних ігор.

ВСТУП

Актуальність проблематики

У сучасному світі, де технологічний прогрес дозволяє досягати суттєвих звершень у будь-якій сфері діяльності людини, виникають нові виклики та загрози, які потребують відповідних заходів безпеки на рівні суспільства та держави. У XXI столітті досі мають місце широкомасштабні та локальні війни. Кожні воєнні конфлікти мають свою специфіку ведення бойових дій, але, якщо ми кажемо за сучасний конфлікт, то він обов'язково буде характеризуватися застосуванням широкого спектра літальних апаратів, що мають різний функціонал та назначення. Одним із найважливіших аспектів забезпечення національної безпеки є системи протиповітряної оборони, які мають на меті захистити країну від повітряних загроз, забезпечити недоторканість її наземних та повітряних кордонів, а також унеможливити виконання будь-яких цілей ворога на території усієї держави.

Актуальність розвитку систем протиповітряної оборони обумовлена не тільки зростанням кількості загроз, які можуть створювати, наприклад, дешеві безпілотники, але й розширенням можливостей літальних апаратів, оскільки вони стають складнішими та сучаснішими. Технологічний прогрес суттєво впливає на розвиток авіації, приводячи до появи нових поколінь літальних апаратів з удосконаленою маневровістю, швидкістю та масовими руйнувальними здібностями. Такі нововведення ставлять під загрозу традиційні системи протиповітряної оборони та вимагають розробки та вдосконалення нових технологій та методів боротьби. А останній рік повномасштабної війни Росії проти України звернув увагу світу на стратегічні зсуви у сфері протиповітряної оборони: дешеві та легкі у виробництві безпілотні літальні апарати (БПЛА), які здатні нести немалу руйнівну силу, суттєво змістили акцент на принципи розвитку та вдосконалення усієї системи протиповітряної оборони, відтягнувши багато уваги на свою сторону. Будь-яка керована ракета протиповітряної оборони, яка складається з просунутих і дорогих систем наведення та керування, коштує у виробництві в декілька разів більше за БПЛА, який здатний заподіяти велику шкоду. Навіть у короткій перспективі країна-оператор таких БПЛА виграє внаслідок економічної переваги та можливостей перевантажити системи протиповітряної

оборони супротивника.

Увесь світ уважно стежить за активним застосуванням дешевих БПЛА у сучасних конфліктах та розробляє ефективні механізми протидії ним, оскільки вони представляють потенційну загрозу для традиційних систем протиповітряної оборони. Одним з таких ефективних механізмів протидії є застосування гарматних зенітних систем. Сучасні гарматні зенітні системи мають дешеві засоби ураження у вигляді снарядів чи куль, високу мобільність та ефективність, але є суттєво обмеженими по дальності роботи.

Не зважаючи на ефективність гарматних зенітних систем, керовані ракети залишаються найбільш універсальним та ефективним засобом перехоплення та ураження повітряних цілей. У зв'язку з цим, головною метою розвитку систем протиповітряної оборони є здешевлення керованих ракет при одночасній максимізації їх ефективності. Технологічний прогрес та інновації у цій галузі спрямовані на поліпшення економічних параметрів ракетних систем та розробку нових методів ведення вогню. Це включає вдосконалення систем наведення та керування, зменшення вартості виробництва компонентів, використання новітніх матеріалів та технологій.

Мета й завдання роботи

Мета даної роботи полягає у вивченні та аналізі систем протиповітряної оборони, конструкції ракет протиповітряної оборони (ППО) та систем керування цими ракетами. Основна увага приділяється математичній задачі диференціальної гри переслідування, яка є класичним підходом для дослідження конфліктних ситуацій, включаючи сценарії перехоплення одного гравця іншим за мінімальний час.

Завдання роботи передбачають наступні етапи. Перш за все, необхідно сформулювати математичну модель, яка з певним наближенням відповідатиме системі протиповітряної оборони, ракетам ППО та їх керуванню. Ця модель повинна адекватно відображати фізичні процеси, що відбуваються в системі.

Далі, треба сформулювати математичну задачу диференціальної гри переслідування, яка буде розв'язуватися у рамках даної роботи. Ця задача має на меті визначити оптимальні стратегії перехоплення цілей для ракет ППО з мінімальним часом переслідування.

Після певних етапів розв'язку задачі потрібно буде розробити програмний код, який допоможе здійснити чисельний розв'язок задачі та провести моделювання конкретних ситуацій переслідування. Це включає написання функцій, обчислювальних методів та створення загального середовища гри для коректної та зрозумілої візуальної реалізації стратегій перехоплення та аналізу результатів.

Останніми завданнями у рамках роботи є візуалізація результатів гри та їх подальший аналіз. Візуалізація допоможе графічно представити динаміку перехоплення та зрозуміти особливості розглянутих стратегій. Аналіз результатів сприятиме отриманню висновків про ефективність різних підходів до переслідування та можливих напрямків для подальших досліджень.

РОЗДІЛ 1
ЗАСОБИ ПЕРЕХОПЛЕННЯ ТА УРАЖЕННЯ ДИНАМІЧНИХ ОБ'ЄКТІВ

1.1 Загальні підходи до перехоплення повітряних цілей

У сучасному світі розвиток військово-технічного потенціалу за останні 75 років призвів до інтенсивного збільшення кількості різних типів повітряних засобів, які здатні виконувати широкий спектр завдань: від розвідувальних та транспортних до ударних. Разом із ними активно відбувався розвиток контрзасобів, що протидіють повітряним цілям. Основним і найбільш універсальним способом протидії є перехоплення повітряної цілі із її подальшим знищенням. Такий підхід забезпечує надійне виконання контрзаходів, направлених на протиборство інтересам умовного ворога.

Увесь процес перехоплення охоплює в собі такі етапи: ідентифікація цілі, вибір траєкторії перехоплення, наведення на ціль та виконання маневру перехоплення, ураження цілі бойовою частиною та осколками від снаряда чи ракети.

Однією з найважливіших етапів у перехоплення повітряних цілей є їх ідентифікації. Цей етап містить в собі визначення типу цілі, її координат, швидкості та напрямку руху. Для досягнення цієї мети використовуються різні сенсори, такі як радари, оптичні системи, інфрачервоні прилади та інші сучасні засоби спостереження. Важливою складовою ідентифікації є відрізнення повітряних цілей від дружніх повітряних засобів, що покращує ефективність та безпеку операцій.

Після ідентифікації цілі необхідно вибрати оптимальну траєкторію перехоплення. Це включає в себе врахування таких факторів, як швидкість і маневровість цілі, доступність і можливості власних засобів перехоплення та ураження, територіальні обмеження та поточні погодні умови. Оптимальна траєкторія дозволяє зменшити час перехоплення та забезпечити успішне наближення до цілі.

Ефективне наведення на ціль та виконання маневру перехоплення є критичними етапами перед фактичним ураженням повітряних цілей. Для досягнення точного наведення використовуються різні системи наведення, такі як системи автоматичного наведення, ракетні комп'ютери та інерціаль-

ні навігаційні системи. Важливим аспектом є забезпечення стійкості наведення та врахування можливих маневрів цілі.

Останнім етапом процесу перехоплення є ураження цілі. Для цього використовуються різні види зброї, залежно від типу цілі та завдання. Найпоширенішими засобами ураження є ракети повітря-повітря із бойовою частиною, що складається з вибухівки та вражаючих елементів. Також використовуються кулемети, гармати та інші типи вогневої зброї. Ефективність ураження залежить від точності наведення та вибору зброї.

1.2 Різновиди засобів перехоплення та ураження повітряних цілей

Основні засоби перехоплення та ураження повітряних цілей - це зенітні комплекси, які включають гарматні та ракетні системи. Гарматні комплекси використовуються для протидії повітряним цілям на відносно невеликих відстанях. Вони оснащені потужними автоматичними гарматами, здатними випускати велику кількість снарядів за одиницю часу, щоб покрити область у просторі для знищення повітряної цілі.

Крім гарматних систем, використовуються також ракетні комплекси. Керовані ракети є найбільш універсальним засобом для перехоплення та ураження повітряних цілей. Вони здатні працювати на великій відстані та перехоплювати цілі, які маневрують у повітрі. Ракетні системи зазвичай оснащені датчиками, які виявляють цілі й передають цю інформацію системі управління. За допомогою керування ракети можуть змінювати свій курс та налаштовувати власну траєкторію, щоб точно захопити та уразити ціль.

Різновиди керованих протиповітряних ракет включають ракети короткої, середньої та великої дальності. Кожен тип ракети має свої особливості і призначення, спрямовані на перехоплення та знищення різних типів повітряних цілей.

Керовані протиповітряні ракети використовують різні системи самонаведення, включаючи: активне-радіолокаційне, інфрачервоне та напівактивне-радіолокаційне наведення. Ці системи дозволяють ракетам визначати положення та швидкість цілі, підтримувати її в полі зору та точно уражати.

В рамках роботи розглядатимуться ракети з активною або напівактивною системою наведення, оскільки саме вони здатні отримувати дані про місцеперебування повітряної цілі та відстань до неї. Ракети з інфрачервоною системою наведення є пасивними та без додаткових датчиків не здатні отримувати дані про відстань до цілі, а отже не можуть застосувати додаткові розрахунки для визначення оптимальної траєкторії для перехоплення. Вони мають обмежені дані про ціль, а саме: отримують тільки напрямок до цілі, оскільки в якості способу локації головка самонаведення пасивно аналізує стан середовища, на яке впливає теплове випромінювання під час роботи силової установки повітряної цілі. Оптимальні траєкторії для таких ракет можуть включати довертання на незначний кут у напрямку руху цілі, що на певний відсоток збільшую ймовірність перехоплення, аніж слідування у точку простору, де знаходиться ціль в кожний момент часу. На практиці ракети з даною системою наведення в більшості випадків є ракетами короткої дальності, оскільки для більш ефективних засобів перехоплення та ураження повинні розраховуватися більш оптимальні траєкторії перехоплення.

1.3 Конструкція ракети протиповітряної оборони

Конструкція ракети протиповітряної оборони (ППО) складається з ряду важливих компонентів, кожен з яких відповідає за виконання конкретних функцій. Оглянемо коротко основні складові конструкції ракети ППО та їх функціонал. Для прикладу наведена конструкція американської зенітної ракети класу "поверхня-повітря"РАС-2, що використовується комплексом MIM-104 Patriot (рисунок 1).

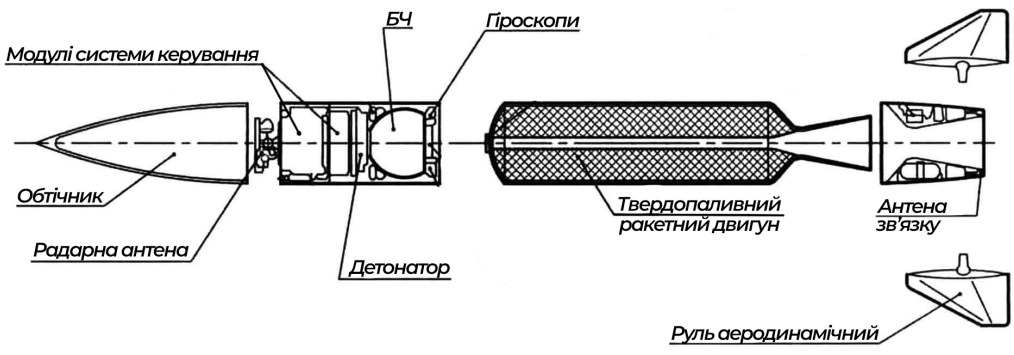

Рисунок 1 - Конструкція ракети РАС-2

Обтічник: Ця частина конструкції ракети забезпечує її аеродинамічні характеристики та дозволяє протягнути ракету через повітря з мінімальним опором. Обтічник гарантує стабільність траєкторії польоту та точність управління.

Радарна антена: Цей елемент використовується для приймання та передачі радіосигналів. Радарна антена є складовою радара, що виявляє цільові об'єкти, визначає їх координати та відстежує їх рух, забезпечуючи точне наведення ракети на ціль.

Модулі системи керування: Система керування ракетою містить комп'ютерні модулі, які обробляють отриману інформацію від радарної антени та інших датчиків. Вони відповідають за розрахунок оптимального шляху польоту, корекцію траєкторії та управління рухом ракети.

Бойова частина: Це основний елемент, який призначений для знищення цілей. Бойова частина може містити вибухову речовину, запальну суміш або інші компоненти, які ефективно здійснюють ураження цілей при зіткненні.

Детонатор: Детонатор використовується для ініціювання вибуху бойової частини ракети у потрібний момент. Він активує вибухову речовину, що спричиняє ураження цілі.

Гіроскопи: Гіроскопи встановлюються для визначення орієнтації ракети у просторі. Вони забезпечують стабілізацію та точність управління ракетою під час польоту.

Твердопаливний ракетний двигун: Цей двигун використовує твердопаливне паливо для створення потужного тягового зусилля, що забезпечує підйом та прискорення ракети.

Антена зв'язку: Антена зв'язку встановлюється для передачі інформації між ракетою та земною станцією. Вона забезпечує комунікацію з оператором та обмін даними про місцеперебування цілей та управління ракетою.

Руль аеродинамічний: Руль аеродинамічний використовується для зміни траєкторії польоту ракети. Він дозволяє коригувати курс та висоту польоту ракети, забезпечуючи точність удару по цілі.

1.4 Система наведення ракети протиповітряної оборони

Система керування ракетою протиповітряної оборони включає в себе різні модулі, що в сумі виконують задачі з визначення оптимальної траєкторії перехоплення, керуванню ракети у просторі та корекції фактичної траєкторії польоту до оптимальної.

Система та методи наведення є ключовою складової у конструкції системи керування ракети. Оптимальні методи наведення повинні забезпечувати найменшу кривизну траєкторії польоту, найпростішу технічну приборну реалізацію системи наведення, а також забезпечення заданої точності на визначених дистанціях.

На практиці виділяють таку класифікацію методів наведення [1]:

а) Двоточкові.

1) Метод прямого наведення.
   - Метод прямого наведення з постійним кутом упередження.
2) Метод погоні.
   - Метод погоні з постійним кутом упередження.
3) Метод паралельного зближення.
4) Метод пропорційного зближення.

б) Триточкові.

1) Метод трьох точок.
2) Методи упередження.
   - Метод з постійним коефіцієнтом упередження.
   - Метод зі змінним коефіцієнтом упередження.

Короткий опис частини методів:

Метод прямого наведення. Цей метод використовується для наведення на ціль, спираючись на інформацію з двох точок. Шлях прямого наведення може бути визначений заздалегідь або визначатись у процесі наведення.

Метод прямого наведення з постійним кутом упередження. У цьому підході використовується постійний кут упередження для наведення на ціль.

Метод погоні. Цей метод передбачає наведення на ціль шляхом переслідування її руху. Він також базується на використанні двох точок.

Метод погоні з постійним кутом упередження. У цьому випадку також використовується постійний кут упередження для ефективної погоні за ціллю.

Метод паралельного зближення. Цей метод полягає в наближенні до цілі паралельно до напрямку її руху.

Метод пропорційного зближення. У цьому методі наведення досягається за допомогою пропорційного зміщення на основі відстані до цілі.

Метод трьох точок. Цей метод використовує три точки (ціль, ракета, центр управління на землі) для точного наведення на ціль. Центр управління дає сигнал на випромінювач на землі, які направлений в бік цілі. Ракета повинна розміститися між двома цими точками й рухатися в просторі не виходячи з відрізка між цими двома точками.

РОЗДІЛ 2
ДИФЕРЕНЦІАЛЬНІ ІГРИ

2.1 Диференціальні ігри як різновид конфліктно-керованих процесів

Конфліктно-керований процес (або конфліктно-керована система) в теорії ігор належать до ситуації, коли кілька учасників (або гравців) взаємодіють, намагаючись оптимізувати свої власні цілі, часто в умовах конфлікту інтересів.

Конкретніше, конфліктно-керовані процеси моделюють ситуації, де гравці приймають рішення, які впливають на загальний стан системи, і при цьому вони можуть бути в умовах прямого конфлікту (наприклад, коли один гравець намагається мінімізувати величину, яку інший намагається максимізувати) або індиректного конфлікту (коли рішення одного гравця впливають на можливості іншого)[2].

Ці системи можуть бути стохастичними, де рішення гравців впливають на випадкові процеси, або детермінованими, де вони впливають на визначені детерміновані процеси. Ці процеси часто моделюються за допомогою диференціальних або різницевих рівнянь, що відображають динаміку системи в часі.

Диференціальна гра - це важлива категорія ігор в теорії ігор, яка розглядає ситуації, у яких час розглядається як неперервна змінна, а гравці впливають на гру шляхом вибору функцій, які описують їхні стратегії. Головна особливість диференціальних ігор полягає в тому, що стратегії гравців впливають на динаміку гри, а динаміка гри своєю чергою визначає стратегії гравців. Вивчення диференціальних ігор вимагає використання методів диференціального рівняння та оптимального керування для моделювання та аналізу динаміки гри.

Гра переслідування - це класичний приклад диференціальної гри, де одні гравці виступають у ролі переслідувачів, а інші - у ролі втікачів. В даній роботі розглядатиметься гра, де у ролі втікача є повітряна ціль, а у ролі переслідувача - керована ракета протиповітряної оборони. У такій грі кожна сторона намагається максимізувати свою вигоду, приймаючи рішення про оптимальну стратегію в умовах невизначеності та залежності від рішень ін-

ших гравців. Основна особливість гри переслідування як диференціальної гри полягає в тому, що стратегії гравців залежать від їх поточного стану та стану опонента. Наприклад, переслідувач може змінювати свою швидкість або напрямок руху, спираючись на дії втікача, який, у свою чергу, намагається підібрати оптимальну траєкторію, щоб уникнути захоплення.

Під захопленням у грі переслідування зазвичай мають на увазі момент, коли відстань між переслідувачем та втікачем досягає заданого значення. Час, за який відбулося захоплення, зазвичай є платою за гру. Саме тому втікач намагається уникнути або відтермінувати момент захоплення, натомість дії переслідувача полягають у тому, щоб захоплення відбулося за мінімальний час.

2.2 Загальні підходи до розв'язку диференціальних ігор

Розв'язком диференціальної гри будемо вважати знаходження хоча б однієї такої функції:

а) Ціна гри $V(x)$, яка визначена на просторі гри $\mathcal{E}$

б) Оптимальні стратегії $\bar{\phi}(x)$, $\bar{\psi}(x)$ для переслідувача та втікача відповідно

в) Оптимальні траєкторії на просторі гри $\mathcal{E}$

Загальні підходи до розв'язку диференціальних ігор полягають у використанні принципу переходу в кожний момент часу. Цей принцип трактується таким чином: "Якщо у грі відбувся перехід з одного положення в інше і якщо в іншому положенні значення функції ціни гри V відоме, то в першому положенні воно визначатиметься наступною умовою: гравці мають оптимізувати прирощення функції V, досягнувши її мінімаксу, за час переходу". [3]

Основне рівняння диференціальної гри, визначається наступним чином:

$$\min_{\phi}\max_{\psi}[\sum_{j} V_j f_j(x,\phi,\psi) + G(x,\phi,\psi)] = 0, \tag{1}$$

де V_j - позначення для $\frac{\partial V}{\partial x_j}$, $j = \overline{1,n}$, де n - кількість фазових координат (розмірність простору гри $\mathcal{E}$).

РОЗДІЛ 3
МАТЕМАТИЧНА МОДЕЛЬ ГРИ ПЕРЕСЛІДУВАННЯ

3.1 Опис гри переслідування

3.1.1 Описання простору гри

У просторі $\mathcal{E} = (x_p, y_p, u, v, x_e, y_e)$ відбувається диференціальна гра з двома гравцями, де через P позначимо переслідувача (з англ. - Pursuer), а через E позначимо втікача (з англ. - Evader).

Місцеположення гравців задається двома координатами (x, y) на площині, де OX - вісь паралельна умовному горизонту, а OY - вісь перпендикулярна горизонту та направлена до умовного центра Землі. За віссю OY також будемо визначати напрямок дії сили тяжіння $\vec{F}_{grav}$.

Переслідувач рухається внаслідок дії сталої сили тяги $\vec{F}_{thrust}$ на його масу m_p. На переслідувача також діють сила тертя $\vec{F}_{drag}$, яка для спрощення буде пропорційна його швидкості руху та протилежна за знаком, а також сила тяжіння $\vec{F}_{grav}$, що діє тільки вздовж осі OY.

Рух втікача описується простою моделлю руху, яка визначається сталою швидкістю ω. Оскільки в даній грі нам цікаво розглянути і знайти оптимальне керування тільки для переслідувача, то ми можемо не ускладнювати модель руху втікача, адже нам не цікаво, як саме втікач буде застосовувати своє керування для того, щоб, наприклад, протидіяти силі тяжіння. Для знаходження оптимального керування переслідувача нам важливі тільки дані про місцеперебування втікача (x_e, y_e) та його швидкість ω, тобто фактично як він пересувається на площині зі сталою швидкістю.

Проводячи паралель між диференціальною грою і прикладною задачею, в ролі переслідувача ми будемо розглядати ракету протиповітряної оборони, що має твердопаливний ракетний двигун з незмінною масою і сталою тягою (для спрощення) та не має крил, що створюють підіймальну силу. В ролі втікача ми розглядатимемо один з трьох повітряних цілей, які матимуть свою певну поведінку:

а) Крилата ракета. Має сталу швидкість та летить заданою заздалегідь траєкторією. Ця траєкторія може бути пряма або криволінійна, але

вона ніяк не реагує на дії переслідувача, а просто виконує заздалегідь визначений план польоту

б) Літак. Теж має сталу швидкість (для спрощення моделі), але має здатність маневрувати та враховувати дії переслідувача у кожний момент часу.

Загалом розв'язок диференціальної гри дасть оптимальні керування для двох гравців. Якщо обидва гравці будуть діяти згідно з цим керуванням, то гра закінчиться в рівноважній точці, яка визначається мінімаксом ціни $V(x)$. Якщо один з гравців відійде від оптимального керування, то другий матиме змогу покращити для себе результати гри. Будемо визначати поведінку переслідувача виключно як оптимальну, тобто такою, що використовуватиме знайдене оптимальне керування, оскільки задача переслідувача (умовної ракети протиповітряної оборони) у тому, щоб якомога швидше наздогнати (перехопити) втікача (повітряну ціль). А втікач натомість не завжди братиме "активну"участь у грі переслідування, адже з точки зору прикладної ситуації (наприклад, втікач - це крилата ракета) він не матиме жодної можливості дізнатися про наявність переслідувача (ракети протиповітряної оборони).

3.1.2 Математична модель руху переслідувача

Модель руху переслідувача задається такою системою рівнянь:

$$\begin{cases} \dot{x_p} = u \\ \dot{y_p} = v \\ \dot{u} = \frac{F}{m_p}\sin(\theta_p) - \frac{ku}{m_p} \\ \dot{v} = \frac{F}{m_p}\cos(\theta_p) - \frac{kv}{m_p} - g, \end{cases} \tag{2}$$

де x_p, y_p - координата переслідувача на площині; u, v - складові вектора швидкості переслідувача вздовж OX та OY відповідно; θ_p - кут між віссю OY та напрямком руху переслідувача; F - стала сила тяги переслідувача, mp - його маса; k - коефіцієнт тертя; g - прискорення вільного падіння.

На рис. 2 зображений переслідувач у вигляді ракети та вектори сил, що діють на центр мас переслідувача [4]. Також іншим кольором зображе-

ний вектор швидкості переслідувача $\vec{U}$, що визначається наступним чином:

$$\vec{U} = \vec{u} + \vec{v}$$

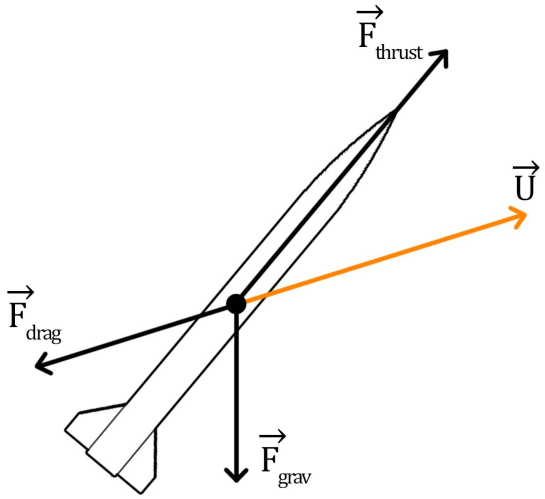

Рисунок 2 - Діаграма сил та швидкості, що діють на переслідувача

Керування переслідувача складається зі змінної θ_p, яка визначає напрямок його польоту. Основною метою розв'язання задачі диференціальної гри про переслідування є знаходження оптимального керування θ_p, при якому відбувається захоплення втікача за мінімальний час T, де T - момент закінчення гри.

Загалом ця модель описує складний інертний рух переслідувача у полі тяжіння та з впливом сили тертя. Мета ускладнення полягає у тому, щоб більш детально дослідити знайдене оптимальне керування, яке повинне, крім того, щоб максимально ефективно протидіяти втікачеві, ще "враховувати"та "протидіяти"природним силам, які діють на переслідувача [5].

3.1.3 Математична модель руху втікача

Модель руху втікача задається такою системою рівнянь:

$$\begin{cases} \dot{x_e} = w \sin(\theta_e) \\ \dot{y_e} = w \cos(\theta_e) \end{cases} \tag{3}$$

де x_e, y_e - координата втікача на площині; w - стала швидкість втікача; θ_e - кут між віссю OY та напрямком руху втікача.

Керування втікача складається зі змінної θ_e, яка також визначає напрямок його польоту. Основною метою розв'язання задачі диференціальної гри про переслідування є знаходження оптимального керування θ_e, при якому втікач має змогу або запобігти захоплення, або максимально відтягнути час T.

3.1.4 Основне рівняння гри

Основне рівняння диференціальних ігор в загальному вигляді написано у (1). У заданій математичній моделі платою за гру є час захоплення, тому $G(x, \phi, \psi) = 1$. Основне рівняння гри в даній моделі виглядатиме наступним чином:

$$\min_{\theta_p}\max_{\theta_e}[V_{x_p}f_{x_p}(x,\theta_p,\theta_e) + V_{y_p}f_{y_p}(x,\theta_p,\theta_e) + V_u f_u(x,\theta_p,\theta_e) + V_v f_v(x,\theta_p,\theta_e)+ \\ +V_{x_e}f_{x_e}(x,\theta_p,\theta_e) + V_{y_e}f_{y_e}(x,\theta_p,\theta_e) + 1] = 0 \tag{3.1}$$

$$\min_{\theta_p}\max_{\theta_e}[uV_{x_p} + vV_{y_p} + V_u(\frac{F}{m_p}\sin(\theta_p) - \frac{ku}{m_p}) + V_v(\frac{F}{m_p}\cos(\theta_p) - \frac{kv}{m_p} - g)+ \\ +V_{x_e}(w\sin(\theta_e)) + V_{y_e}(w\cos(\theta_e)) + 1] = 0 \tag{3.2}$$

Розкриваємо оператори $\min\limits_{\theta_p}$ та $\max\limits_{\theta_e}$:

$$uV_{x_p} + vV_{y_p} - \frac{k}{m_p}(uV_u + vV_v) - V_v g + \frac{F}{m_p}\min_{\theta_p}[V_u\sin(\theta_p) + V_v\cos(\theta_p)]+ \\ +w\max_{\theta_e}[V_{x_e}sin(\theta_e) + V_{y_e}\cos(\theta_e)] + 1 = 0 \tag{3.3}$$

Покладемо, що: $\rho_p = \sqrt{V_u^2 + V_v^2}$ та $\rho_e = \sqrt{V_{x_e}^2 + V_{y_e}^2}$. Тоді за лемою про кругові вектограми в [3](2.8.1) отримаємо, що при:

$$\begin{array}{ll} \sin(\bar{\theta_p}) = -\frac{V_u}{\rho_p} & \cos(\bar{\theta_p}) = -\frac{V_v}{\rho_p} \\ \sin(\bar{\theta_e}) = \frac{V_{xe}}{\rho_e} & \cos(\bar{\theta_e}) = \frac{V_{ye}}{\rho_e} \end{array} \tag{3.4}$$

Кути $\bar{\theta_p}$ та $\bar{\theta_e}$ мінімізують та максимізують відповідні вирази під операторами $\min\limits_{\theta_p}$ та $\max\limits_{\theta_e}$. Тоді остаточно отримаємо основне рівняння для заданої математичної моделі:

$$uV_{x_p} + vV_{y_p} - \frac{k}{m_p}(uV_u + vV_v) - V_v g + \frac{F}{m_p}\rho_p + w\rho_e + 1 = 0 \tag{3.5}$$

3.2 Формулювання та розв'язок задачі переслідування

3.2.1 Формулювання задачі

Постановка задачі виглядає наступним чином:

Маємо диференціальну гру у просторі $\mathcal{E} = (x_p, y_p, u, v, x_e, y_e)$ з плином часу t. Дії гравців P та E описуються системами рівнянь (2) і (3) відповідно. Мета гравця P полягає у переслідуванні та захопленні E. Ціна гри визначається часом до закінчення гри $\tau(x)$ для кожного моменту стану $x \in \mathcal{E}$. Основне рівняння гри задане у рівнянні (3.5). Захоплення відбувається коли $|PE| \leq l$, де l - радіус захоплення.

Знайти оптимальні стратегії $\theta_p = \bar{\theta_p}$ та $\theta_e = \bar{\theta_e}$ гравців P та E відповідно; знайти ціну гри $\tau(x)$; знайти та побудувати оптимальні траєкторії, які відповідають знайденим оптимальним стратегіям гравців.

3.2.2 Регресивний підхід до розв'язку

Для знаходження розв'язку використаємо принцип переходу для кінцевого моменту гри $t = T(x)$, коли вже відбулося захоплення. Таким чином ми розвернемо гру у зворотний бік в контексті плину часу та будемо застосовувати принцип переходу у зворотному напрямку.

Розглянемо кінцевий момент гри, коли $t = T(x)$ та $|PE| \leq 0$. Позначимо змінні стану гри в кінцевий момент через параметри s_i, де $i = \overline{1,5}$:

$$x_p = s_1, \tag{4.1}$$

$$y_p = s_2, \tag{4.2}$$

$$u = s_3, \tag{4.3}$$

$$v = s_4, \tag{4.4}$$

$$x_e = s1 + l\sin(s_5), \tag{4.5}$$

$$y_e = s2 + l\cos(s_5). \tag{4.6}$$

Допустима область захоплення визначатиметься наступною нерівністю:

$$w - s_5\sin(s_5) - s_4\cos(s_5) < 0 \tag{5}$$

На практиці вона означає, що кінцева швидкість переслідувача повинна бути більшою за швидкість втікача.

Знайдемо рівняння характеристик у регресивній формі, де $\overset{\circ}{x_p}$ означає $\frac{\partial x_p}{\partial \tau}$, а τ визначається як $\tau = T(x) - t$:

$$\overset{\circ}{x_p} = -u, \tag{6.1}$$

$$\overset{\circ}{y_p} = -v, \tag{6.2}$$

$$\overset{\circ}{u} = \frac{F}{m_p}\frac{V_u}{\rho_p} + k\frac{u}{m_p}, \tag{6.3}$$

$$\overset{\circ}{v} = \frac{F}{m_p}\frac{V_v}{\rho_p} + k\frac{v}{m_p} + g, \tag{6.4}$$

$$\overset{\circ}{x_e} = -w\frac{V_{xe}}{\rho_e}, \tag{6.5}$$

$$\overset{\circ}{y_e} = -w\frac{V_{ye}}{\rho_e}, \tag{6.6}$$

$$\overset{\circ}{V_{x_p}} = 0, \tag{6.7}$$

$$\overset{\circ}{V_{y_p}} = 0, \tag{6.8}$$

$$\overset{\circ}{V_u} = V_{x_p} - \frac{k}{m_p}V_u, \tag{6.9}$$

$$\overset{\circ}{V_v} = V_{y_p} - \frac{k}{m_p}V_v, \tag{6.10}$$

$$\overset{\circ}{V_{x_e}} = 0, \tag{6.11}$$

$$\overset{\circ}{V_{y_e}} = 0. \tag{6.12}$$

Наступним кроком розв'язку буде інтегрування рівнянь характеристик для знаходження невідомих змінних $s_1(x), s_2(x), s_3(x), s_4(x), s_5(x), \tau(x)$.

Але перед цим важливо визначити V_i на термінальній множині, тим самим доповнив множину початкових значень для подальшого інтегрування. Оскільки V_i - це похідна від ціни гри по i-й змінній, то при на термінальній множині їх значення зануляються:

$$V_{s_1} = 0 = V_{x_p} + V_{x_e}, \tag{7.1}$$

$$V_{s_2} = 0 = V_{y_p} + V_{y_e}, \tag{7.2}$$

$$V_{s_3} = 0 = V_u, \tag{7.3}$$

$$V_{s_4} = 0 = V_v, \tag{7.4}$$

$$V_{s_5} = 0 = V_{x_e}\cos(s_5) - V_{y_e}\sin(s_5). \tag{7.5}$$

З рівнянь (7.1),(7.2) та (7.5) при певному λ знайдемо:

$$\begin{aligned} -V_{x_p} = V_{x_e} = \lambda\sin(s_5), \\ -V_{y_p} = V_{y_e} = \lambda\cos(s_5). \end{aligned} \tag{8}$$

На цей момент у нас достатньо початкових умов та рівнянь, щоб проінтегрувати рівняння характеристик. Почнемо з інтегрування рівнянь (6.9) та (6.10) та знайдемо $\overset{\circ}{V_u}$ та $\overset{\circ}{V_v}$:

$$V_u = -\lambda\frac{m_p}{k}\sin(s_5)(1 - e^{-\frac{k\tau}{m_p}})$$
$$V_v = -\lambda\frac{m_p}{k}\cos(s_5)(1 - e^{-\frac{k\tau}{m_p}})$$

На цьому моменті ми можемо повернутися до рівнянь в (3.4) та знайти оптимальні керування для гравців. Оскільки для гравця P:

$$\sin(\bar{\theta_p}) = -\frac{V_u}{\rho_p} = -\frac{-\lambda\frac{m_p}{k}\sin(s_5)(1 - e^{-\frac{k\tau}{m_p}})}{\sqrt{V_u^2 + V_v^2}} = -\frac{-\lambda\frac{m_p}{k}\sin(s_5)(1 - e^{-\frac{k\tau}{m_p}})}{|\lambda\frac{m_p}{k}(1 - e^{-\frac{k\tau}{m_p}})|} = \sin(s_5),$$

$$\cos(\bar{\theta_p}) = -\frac{V_v}{\rho_p} = -\frac{-\lambda\frac{m_p}{k}\cos(s_5)(1 - e^{-\frac{k\tau}{m_p}})}{\sqrt{V_u^2 + V_v^2}} = -\frac{-\lambda\frac{m_p}{k}\cos(s_5)(1 - e^{-\frac{k\tau}{m_p}})}{|\lambda\frac{m_p}{k}(1 - e^{-\frac{k\tau}{m_p}})|} = \cos(s_5),$$

та для гравця E:

$$\sin(\bar{\theta}_e) = \frac{V_{x_e}}{\rho_e} = \frac{\lambda \sin(s_5)}{\sqrt{V_{x_e}^2 + V_{y_e}^2}} = \frac{\lambda \sin(s_5)}{|\lambda|} = \sin(s_5),$$

$$\cos(\bar{\theta}_e) = \frac{V_{y_e}}{\rho_e} = \frac{\lambda \cos(s_5)}{\sqrt{V_{x_e}^2 + V_{y_e}^2}} = \frac{\lambda \cos(s_5)}{|\lambda|} = \cos(s_5),$$

то можна зробити висновок, що оптимальні керування гравців $\bar{\theta}_p = \bar{\theta}_e = s_5(x)$ дорівнюють одне одному. У прикладному сенсі це означає, що оптимальною стратегією для втікача є рух у напрямку від переслідувача і для переслідувача відповідно навпаки.

Наступним кроком у розв'язку поставленої задачі буде пошук функції ціни гри $\tau(x)$, через яке ми вже зможемо знайти функцію оптимального керування $s_5(x)$ для гравців.

3.2.3 Знаходження ціни гри

Ціна гри переслідування в даній математичній моделі визначається функцією $\tau(x_0)$, що дорівнює часу до моменту захоплення втікача переслідувачем при оптимальній поведінці обох гравців, тобто часу, за який система перейде зі стану $x_0 = (x_{p0}, y_{p0}, u_0, v_0, x_{e0}, y_{e0})$ у стан $x_s = (s_1, s_2, s_3, s_4, s_1 + l\sin(s_5), s_2 + l\cos(s_5))$ на термінальній множині.

Для знаходження $\tau(x)$ нам треба знайти характеристичні функції у регресивній формі. Для чого спочатку проінтегруємо рівняння (6.3) та (6.4):

$$\frac{\partial u}{\partial \tau} = \frac{F}{m_p}\frac{V_u}{\rho_p} + k\frac{u}{m_p},$$

$$\ln(ku - F\sin s_5) = \frac{k\tau}{m_p} + C_1,$$

де з початкової умови (4.3) знайдемо константу інтегрування:

$$C_1 = \ln(ks_3 - F\sin(s_5)),$$

тоді остаточно отримаємо:

$$ku - F\sin(s_5) = e^{\frac{k\tau}{m_p}}(ks_3 - F\sin(s_5)),$$

$$u(\tau) = s_3 e^{\frac{k\tau}{m_p}} - F\sin(s_5)\frac{e^{\frac{k\tau}{m_p}} - 1}{k}. \tag{9.1}$$

Знайдемо $v(\tau)$ з урахуванням наявності у рівнянні (6.4) ще одного доданку g:

$$\frac{\partial v}{\partial \tau} = \frac{F}{m_p}\frac{V_v}{\rho_p} + k\frac{v}{m_p} + g,$$

$$\ln(kv - F\cos(s_5) + gm_p) = \frac{k\tau}{m_p} + C_2,$$

де з початкової умови (4.4) знайдемо константу інтегрування:

$$C_2 = \ln(ks_4 - F\cos(s_5) + gm_p),$$

тоді остаточно отримаємо:

$$kv - F\cos(s_5) + gm_p = e^{\frac{k\tau}{m_p}}(ks_4 - F\cos(s_5) + gm_p),$$

$$v(\tau) = s_4 e^{\frac{k\tau}{m_p}} - F\cos(s_5)\frac{e^{\frac{k\tau}{m_p}} - 1}{k} + gm_p\frac{e^{\frac{k\tau}{m_p}} - 1}{k}. \tag{9.2}$$

Знайшовши рівняння (9.1) та (9.2) можемо перейти до інтегрування рівнянь (6.1) та (6.2), щоб знайти функції $x_p(\tau)$ та $y_p(\tau)$:

$$\frac{\partial x_p}{\partial \tau} = -u = -s_3 e^{\frac{k\tau}{m_p}} + F\sin(s_5)\frac{e^{\frac{k\tau}{m_p}} - 1}{k},$$

$$x_p = -s_3\frac{m_p}{k}e^{\frac{k\tau}{m_p}} + \frac{F\sin(s_5)}{k}\left(\frac{m_p}{k}e^{\frac{k\tau}{m_p}} - \tau\right) + C_3,$$

де з початкової умови (4.1) знайдемо константу інтегрування:

$$C_3 = s_1 + s_3\frac{m_p}{k} - \frac{F\sin(s_5)m_p^2}{k^2},$$

тоді остаточно отримаємо:

$$x_p = -s_3\frac{m_p}{k}e^{\frac{k\tau}{m_p}} + \frac{F\sin(s_5)}{k}\left(\frac{m_p}{k}e^{\frac{k\tau}{m_p}} - \tau\right) + s_1 + s_3\frac{m_p}{k} - \frac{F\sin(s_5)m_p^2}{k^2},$$

$$x_p(\tau) = s_1 - s_3\frac{m_p}{k}(e^{\frac{k\tau}{m_p}} - 1) + \frac{F\sin(s_5)m_p}{k^2}\Big(e^{\frac{k\tau}{m_p}} - \frac{k\tau}{m_p} - 1\Big). \tag{9.3}$$

Знайдемо схожим чином $y_p(\tau)$ з урахуванням наявності у рівнянні (9.2) ще одного доданку $gm_p\frac{e^{\frac{k\tau}{m_p}}-1}{k}$:

$$\frac{\partial y_p}{\partial \tau} = -v = -s_4e^{\frac{k\tau}{m_p}} + F\cos(s_5)\frac{e^{\frac{k\tau}{m_p}} - 1}{k} - gm_p\frac{e^{\frac{k\tau}{m_p}} - 1}{k},$$

$$y_p = -s_4\frac{m_p}{k}e^{\frac{k\tau}{m_p}} + \Big(\frac{F\cos(s_5)}{k} - \frac{gm_p}{k}\Big)\Big(\frac{m_p}{k}e^{\frac{k\tau}{m_p}} - \tau\Big) + C_4,$$

де з початкової умови (4.2) знайдемо константу інтегрування:

$$C_4 = s_2 + s_4\frac{m_p}{k} - \frac{F\cos(s_5)m_p - gm_p^2}{k^2},$$

тоді остаточно отримаємо:

$$y_p = -s_4\frac{m_p}{k}e^{\frac{k\tau}{m_p}} + \Big(\frac{F\cos(s_5)}{k} - \frac{gm_p}{k}\Big)\Big(\frac{m_p}{k}e^{\frac{k\tau}{m_p}} - \tau\Big) + \\ + s_2 + s_4\frac{m_p}{k} - \frac{F\cos(s_5)m_p - gm_p^2}{k^2},$$

$$y_p(\tau) = s_2 - s_4\frac{m_p}{k}(e^{\frac{k\tau}{m_p}} - 1) + \frac{F\cos(s_5)m_p}{k^2}\Big(e^{\frac{k\tau}{m_p}} - \frac{k\tau}{m_p} - 1\Big) - \\ - \frac{gm_p^2}{k^2}\Big(e^{\frac{k\tau}{m_p}} - \frac{k\tau}{m_p} - 1\Big). \tag{9.4}$$

Проінтегруємо рівняння (6.5) та (6.6):

$$\frac{\partial x_e}{\partial \tau} = -w\frac{V_{x_e}}{\rho_e} = -w\sin(s_5),$$

$$x_e = -w\sin(s_5)\tau + C_5,$$

де з початкової умови (4.5) знайдемо константу інтегрування:

$$C_5 = s_1 + l\sin(s_5),$$

тоді остаточно отримаємо:

$$x_e(\tau) = s_1 + \sin(s_5)(l - w\tau). \tag{9.5}$$

Повністю аналогічним чином знайдемо $y_e(\tau)$:

$$y_e(\tau) = s_2 + \cos(s_5)(l - w\tau). \tag{9.6}$$

Отже в результаті операцій інтегрування ми отримали системи рівнянь (9.1)-(9.6):

$$\begin{cases} u(\tau) = s_3 e^{\frac{k\tau}{m_p}} - F\sin(s_5)\frac{e^{\frac{k\tau}{m_p}}-1}{k}, \\ v(\tau) = s_4 e^{\frac{k\tau}{m_p}} - F\cos(s_5)\frac{e^{\frac{k\tau}{m_p}}-1}{k} + gm_p\frac{e^{\frac{k\tau}{m_p}}-1}{k}, \\ x_p(\tau) = s_1 - s_3\frac{m_p}{k}(e^{\frac{k\tau}{m_p}} - 1) + \frac{F\sin(s_5)m_p}{k^2}\Big(e^{\frac{k\tau}{m_p}} - \frac{k\tau}{m_p} - 1\Big), \\ y_p(\tau) = s_2 - s_4\frac{m_p}{k}(e^{\frac{k\tau}{m_p}} - 1) + \frac{F\cos(s_5)m_p}{k^2}\Big(e^{\frac{k\tau}{m_p}} - \frac{k\tau}{m_p} - 1\Big) - \frac{gm_p^2}{k^2}\Big(e^{\frac{k\tau}{m_p}} - \frac{k\tau}{m_p} - 1\Big), \\ x_e(\tau) = s_1 + \sin(s_5)(l - w\tau), \\ y_e(\tau) = s_2 + \cos(s_5)(l - w\tau). \end{cases}$$

Для знаходження розв'язків поставленої задачі нам треба розв'язати цю систему рівнянь відносно шести невідомих $s_1, s_2, s_3, s_4, s_5, \tau$. Для цього спочатку використаємо теорему про побудову розв'язку, побудуємо функцію $Q(\tau)$ та знайдемо шукану ціну гри $\tau(x)$. Буде отримана така функція $Q(\tau)$:

$$Q(\tau) = l - w\tau + \frac{Fm_p - gm_p^2}{k^2}\left(\frac{k\tau}{m_p} + e^{-\frac{k\tau}{m_p}} - 1\right). \tag{10.1}$$

За допомогою нескладних перетворень отримаємо допоміжні рівняння, які надалі допоможуть виключити $s_5(x)$:

$$x_e - x_p - u\frac{m_p}{k}\Big(1 - e^{-\frac{k\tau}{m_p}}\Big) = \left(Q + \frac{gm_p^2}{k^2}\Big(e^{-\frac{k\tau}{m_p}} + \frac{k\tau}{m_p} - 1\Big)\right)\sin(s_5), \tag{10.2}$$

$$\begin{aligned} y_e - y_p - v\frac{m_p}{k}\Big(1 - e^{-\frac{k\tau}{m_p}}\Big) + \frac{gm_p^2}{k^2}\Big(e^{-\frac{k\tau}{m_p}} + \frac{k\tau}{m_p} - 1\Big) = \\ = \left(Q + \frac{gm_p^2}{k^2}\Big(e^{-\frac{k\tau}{m_p}} + \frac{k\tau}{m_p} - 1\Big)\right)\cos(s_5). \end{aligned} \tag{10.3}$$

Підносимо рівняння (10.2) та (10.3) до квадрата та додаємо їх, щоб позбутися $s_5(x)$. Для спрощення запису позначимо за $A = \frac{gm_p^2}{k^2}\big(e^{-\frac{k\tau}{m_p}} + \frac{k\tau}{m_p} - 1\big)$, а за $E_1 = \frac{m_p}{k}\big(1 - e^{-\frac{k\tau}{m_p}}\big)$. Тоді отримаємо:

$$r^2 - 2rUE_1 + U^2E_1^2 + A^2 + 2A(y_e - y_p - vE_1) = \big(Q + A\big)^2, \qquad (10.4)$$

де $r = (x_e - x_p, y_e - y_p)$ - вектор між втікачем та переслідувачем; $U = (u, v)$ - вектор швидкості переслідувача.

Рівняння (10.4) визначає взаємозалежність між змінними стану x_p, y_p, u, v, x_e, y_e та ціною гри $\tau(x)$. З цього рівняння вже можна знайти функцію $\tau(x)$ для кожного фазового стану системи.

Оскільки $\tau(x)$ входить в рівняння (10.4) нетривіально, то пошук коренів в аналітичному вигляді буде занадто складним або неможливим. Тому для знаходження коренів рівняння (10.4) застосуємо чисельні методи, реалізовані за допомогою мови програмування Python у програмному середовищі PyCharm. Код програми наведений у додатку А.

При побудові Q-функції важливо зазначити, що вона повинна приймати додатні значення $Q(\tau) > 0$ при будь-якому невід'ємному τ на множині заданих параметрів (l, w, F, m_p, g, k). Саме тому перед моделюванням конкретних ігор функція $Q(\tau)$ буде окремо досліджуватись на її знак.

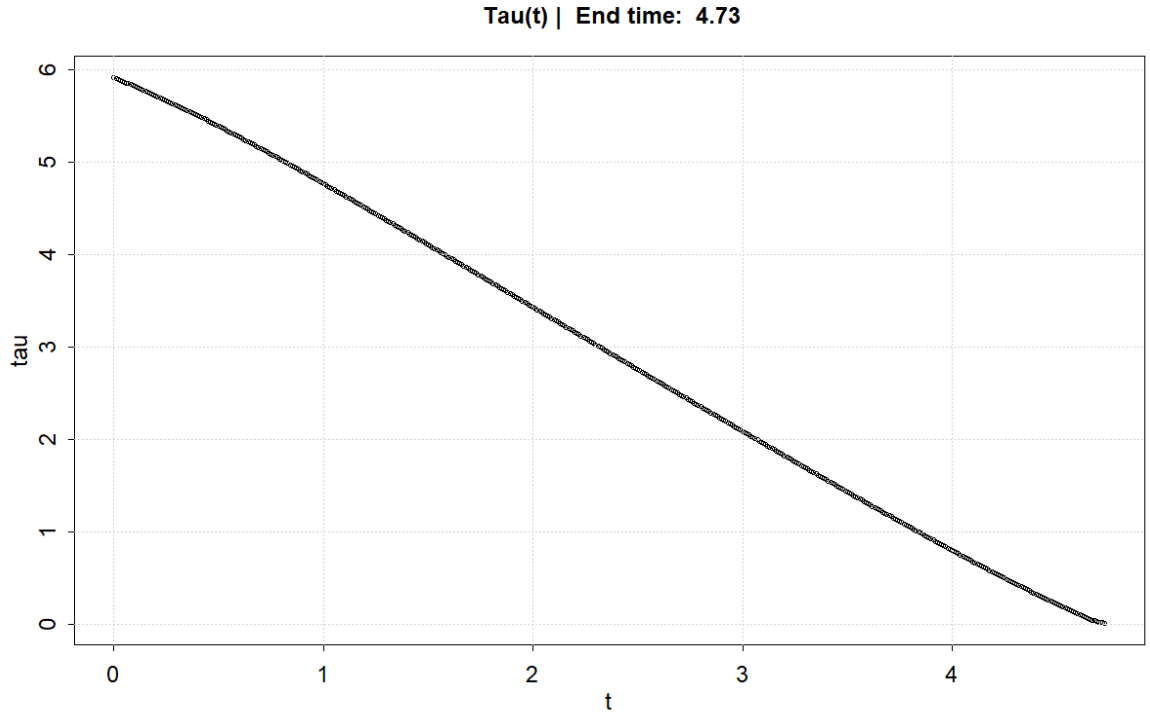

Рисунок 3 - Графік $\tau(t)$

Розглянемо конкретний приклад диференціальної гри з набором параметрів: $(l, w, F, m_p, g, k) = (10.0, 40.0, 200.0, 1.0, 9.81, 1.0)$. У цьому випадку

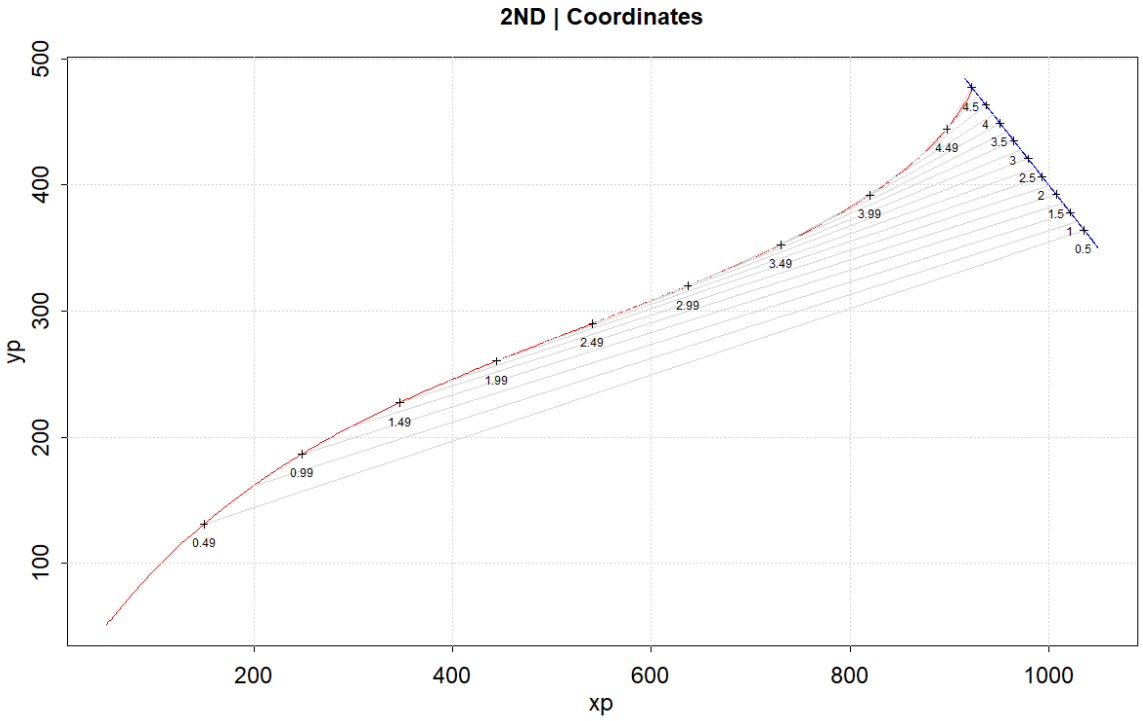

Рисунок 4 - Графік $y_{p,e}(x_{p,e})$

впускатимемо дослідження поведінки функції $Q(\tau)$, адже це буде зроблено пізніше більш детально. За допомогою вбудованих бібліотек була написана функція, яка підраховує найменший корінь рівняння (10.4) за допомогою гібридного чисельного методу [6]. У результаті був побудований графік функції $\tau(t)$ для заданої гри, який відображає міру часу, яка залишилася до захоплення переслідувачем втікача (рисунок 3).

На рисунку 4 зображені траєкторії переслідувача та втікача для цієї ж самої конфігурації гри: а) червоний колір - траєкторія переслідувача; б) синій колір - траєкторія втікача. Рух втікача відбувається на прямій без маневрування.

3.2.4 Знаходження оптимального керування та стратегії

Як було вже визначено, оптимальне керування $\bar{\theta}_p$ та $\bar{\theta}_e$ для гравців визначається параметром $s_5(\tau)$, який своєю чергою знаходиться через ціну гри. Таким чином ми вже можемо підійти до моменту знаходження оптимальних керувань та визначенню стратегій для гравців.

Повернемося до рівнянь (10.2) та (10.3). Залишимо використані в минулому пункті скорочення для спрощення викладок. Поділимо одне рівня-

ння на інше, врахувавши появу нових потенційних обмежень:

$$\frac{x_e - x_p - u\frac{m_p}{k}\left(1 - e^{-\frac{k\tau}{m_p}}\right)}{y_e - y_p - v\frac{m_p}{k}\left(1 - e^{-\frac{k\tau}{m_p}}\right) + \frac{gm_p^2}{k^2}\left(e^{-\frac{k\tau}{m_p}} + \frac{k\tau}{m_p} - 1\right)} = \frac{(Q+A)\sin(s_5)}{(Q+A)\cos(s_5)} = \tan(s_5), \tag{11.1}$$

$$s_5 = \arctan\left(\frac{x_e - x_p - u\frac{m_p}{k}\left(1 - e^{-\frac{k\tau}{m_p}}\right)}{y_e - y_p - v\frac{m_p}{k}\left(1 - e^{-\frac{k\tau}{m_p}}\right) + \frac{gm_p^2}{k^2}\left(e^{-\frac{k\tau}{m_p}} + \frac{k\tau}{m_p} - 1\right)}\right). \tag{11.2}$$

Отримали функцію $s_5(\tau)$, задану у явному вигляді, яка визначає оптимальні керування гравців в кожний момент гри $\tau > 0$ при заданих параметрах (l, w, F, m_p, g, k) та у конкретному фазовому стані системи $(x_p, y_p, u, v, x_e, y_e)$.

На рисунку 5 зображений графік функції $\theta_p(t)$. Величина кута θ_p відраховується від вертикальної осі OY та на графіку позначена в градусах. Жовті та сірі криві на графіку відображають кути між вертикальною віссю і вектором швидкості переслідувача та напрямком до центру мас втікача відповідно.

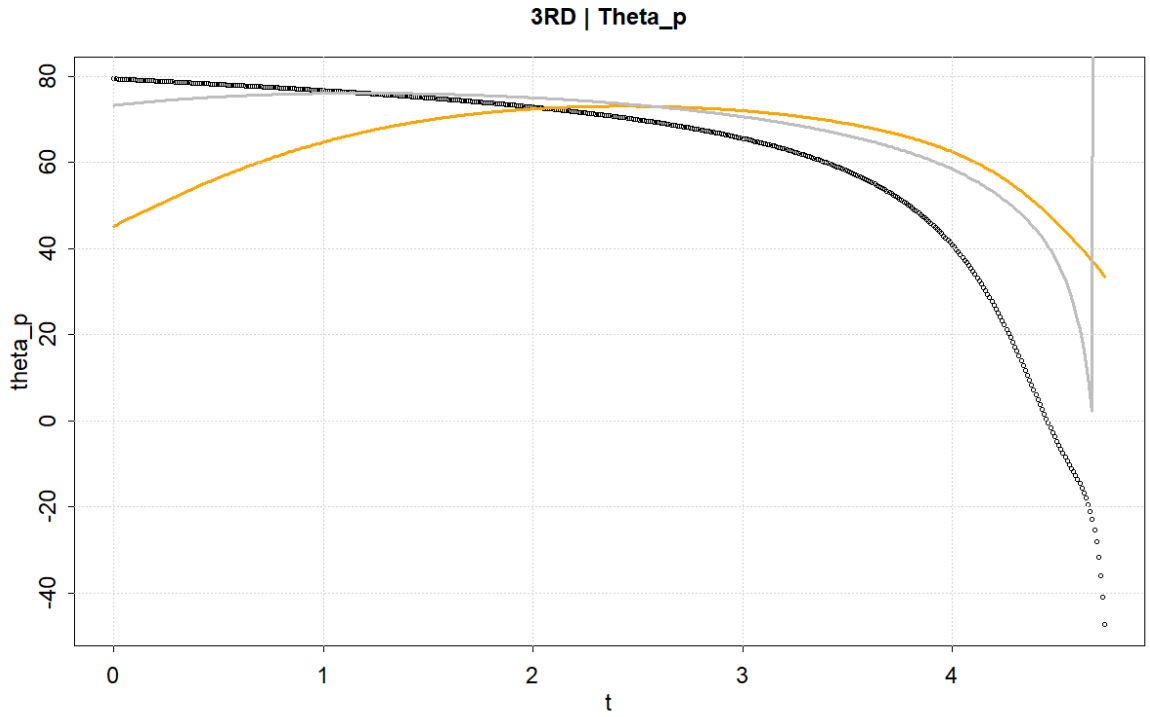

Рисунок 5 - Графік $\theta_p(t), \alpha_{\vec{U}}(t), \alpha_{toE}(t)$:
чорний, жовтий та сірий кольори відповідно

Як описувалося раніше, оптимальні стратегії досягаються при дотриманні оптимальних керувань $\bar{\theta}_p = \bar{\theta}_e = s_5(\tau)$ та на практиці полягають у втечі по прямій від переслідувача втікачем. Варто врахувати, що обидва гравці мають різні моделі руху, що може охоплювати в собі різницю у масі, швидкостях, а й відповідно у мірі маневровності. Тому одна з успішних стратегій для втікача може бути активне маневрування аби не дати переслідувачеві з обмеженою маневровністю підібратися на достатньо близьку відстань, щоб виконати захоплення. Такі ігрові ситуації при різних початкових станах і параметрах розглянемо в наступному розділі.

РОЗДІЛ 4
АНАЛІЗ РЕЗУЛЬТАТІВ ГРИ

4.1 Моделювання різних початкових моментів гри

Розглянемо спочатку декілька звичайних комбінацій початкових моментів гри та наборів параметрів:

Гра 1. Переслідувач P діє оптимально, втікач E діє неоптимально: рухається синусоїдою у напрямку переслідувача. Параметри та стан системи в момент $t = 0$:

$$x_{p0} = 50 \qquad l = 10.0$$
$$y_{p0} = 50 \qquad w = 50.0$$
$$u_0 = 0 \qquad F = 200.0$$
$$v_0 = 0 \qquad m_p = 1.0$$
$$x_{e0} = 1050 \qquad k = 1.0$$
$$y_{e0} = 350 \qquad g = 5.0$$

Перш ніж будемо аналізувати результати гри треба дослідити поведінку функції $Q(\tau)$, яка задається рівнянням (10.1), оскільки розв'язність задачі залежить від знаку $Q(\tau)$. Побудуємо графік функції за цим рівнянням і будемо підбирати так параметри, щоб $Q(\tau) > 0$ при усіх невід'ємних τ:

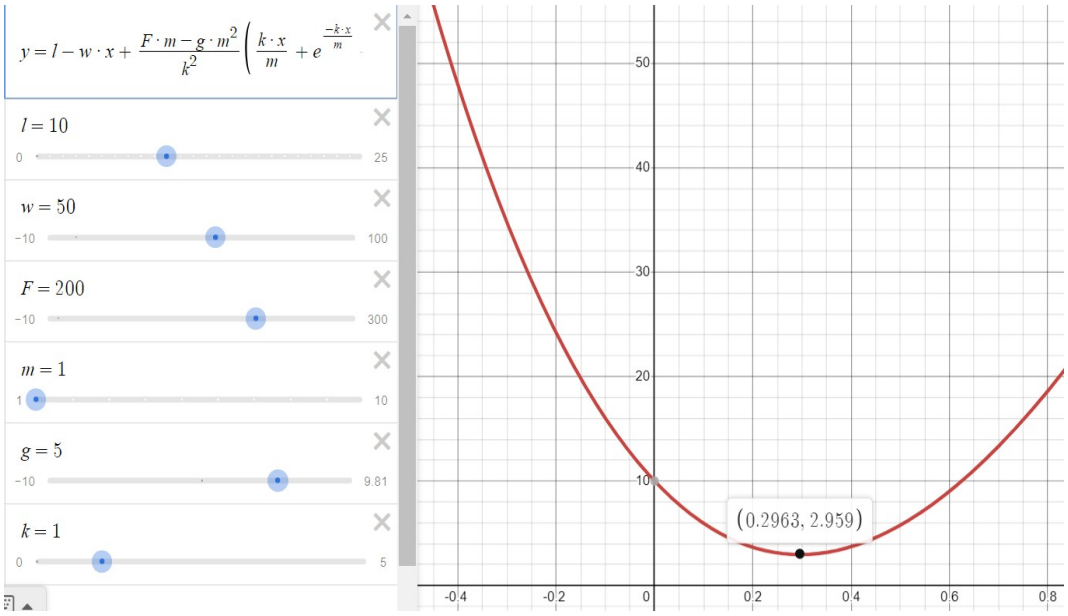

Рисунок 6 - Графік $Q(\tau)$ при множині параметрів з **Гри 1**.
Ресурс: desmos.com[7]

На рисунку 6 зображена крива, що задається функцією $Q(\tau)$. На рівні поверхневого аналізу ми можемо зрозуміти, що графік функції має єдиний екстремум, який є мінімумом функції й приймає додатне значення. Отже, для конкретної **Гри 1** із заданими параметрами виконується нерівність $Q(\tau) > 0$, що забезпечує наявність оптимальних розв'язків для керувань в даній грі. Можемо провести моделювання та подальший аналіз.

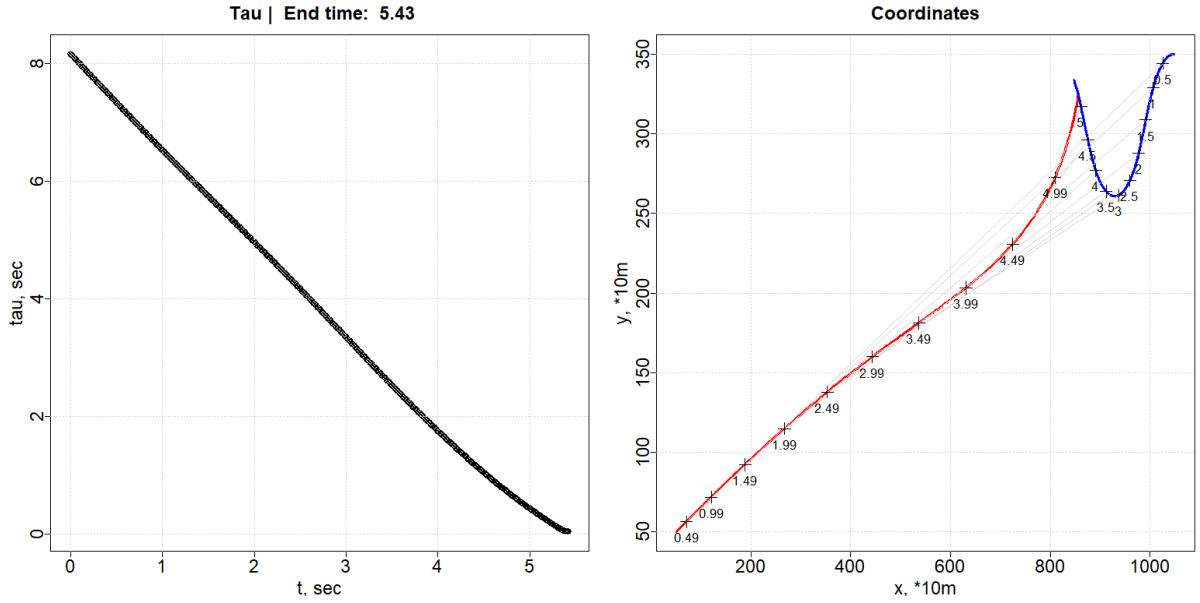

Рисунок 7 - Графіки $\tau(t)$ та $y_{p,e}(x_{p,e})$ для **Гри 1**

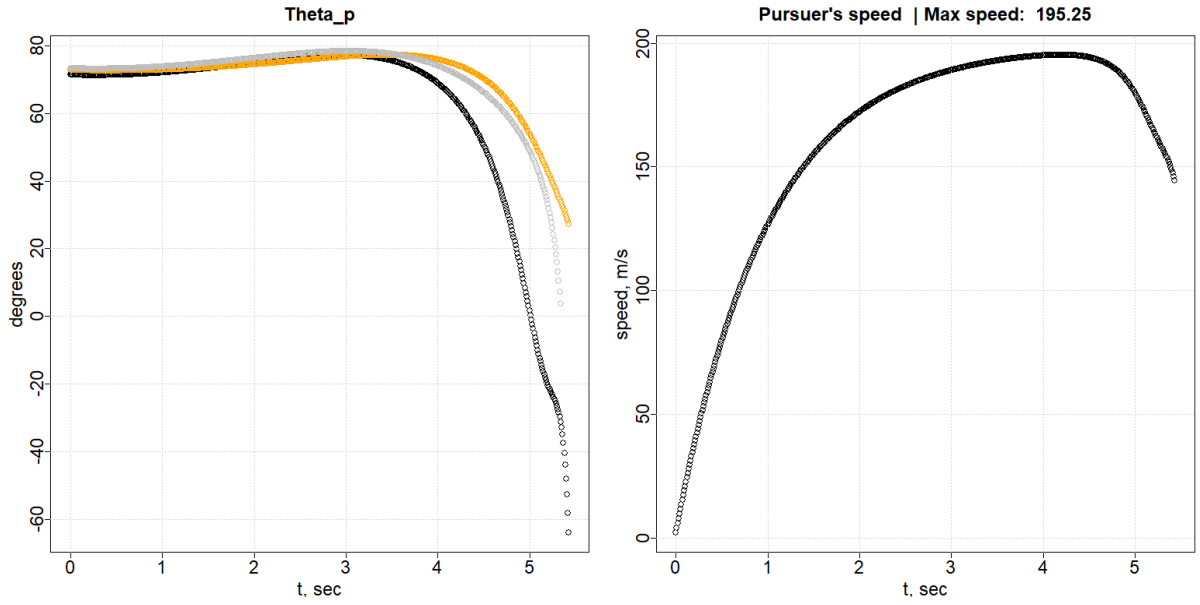

Рисунок 8 - Графіки $\theta_p(t)$ та $U(t)$ для **Гри 1**

На рисунках 7-8 зображені 4 графіки, які зображають перебіг **Гри 1** при заданих параметрах. У цій грі переслідувач перехоплює втікача за $\tau = 6.07$ с.

Варто звернути увагу на графік $\theta_p(t)$ кута керування, а саме на те, як чорна крива на графіку, що відображає кут керування, відрізняється від жовтої кривої, що зображує фактичний напрямок руху переслідувача. Це відбувається з двох причин: переслідувач має інертність, а отже йому потрібні додаткові зусилля аби змінити свій фактичний напрямок руху; переслідувач в рамках оптимального керування бере певне упередження, аби більш ефективно досягти цілі та перехопити втікача.

Гра 2. Переслідувач P діє оптимально, втікач E діє неоптимально: рухається синусоїдою у напрямку переслідувача. Переслідувач починає з ненульовою швидкістю, направленою в інший бік від втікача. Параметри та стан системи в момент $t = 0$:

$$x_{p0} = 50 \quad l = 10.0$$
$$y_{p0} = 50 \quad w = 65.0$$
$$u_0 = 120 \quad F = 240.0$$
$$v_0 = 160 \quad m_p = 1.0$$
$$x_{e0} = 1050 \quad k = 1.0$$
$$y_{e0} = 200 \quad g = 5.0$$

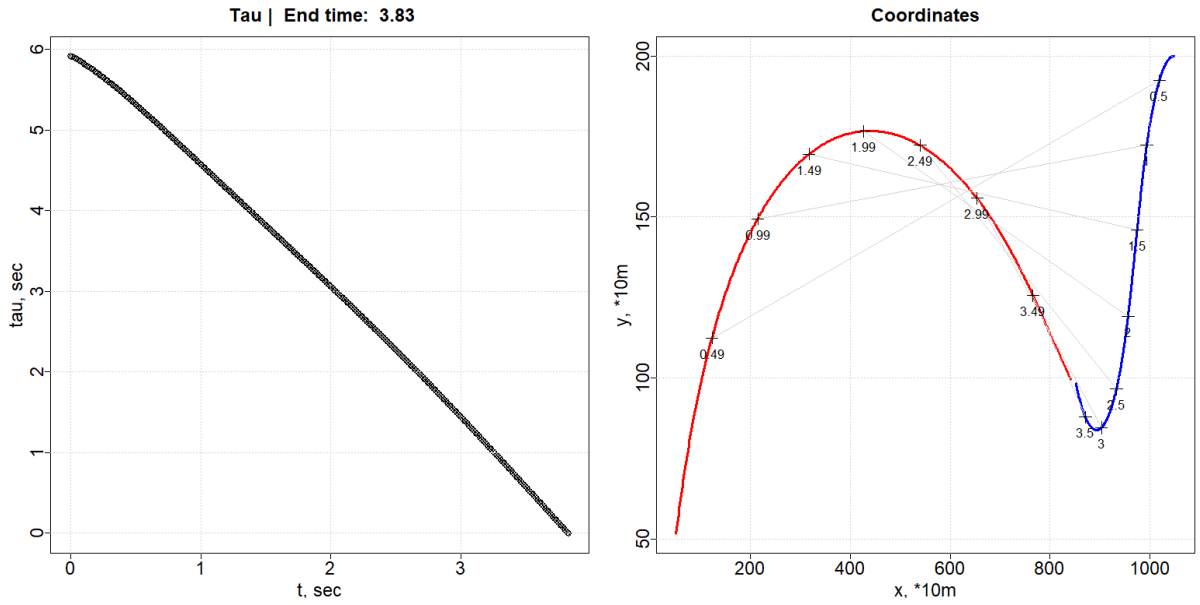

Рисунок 9 - Графіки $\tau(t)$ та $y_{p,e}(x_{p,e})$ для **Гри 2**

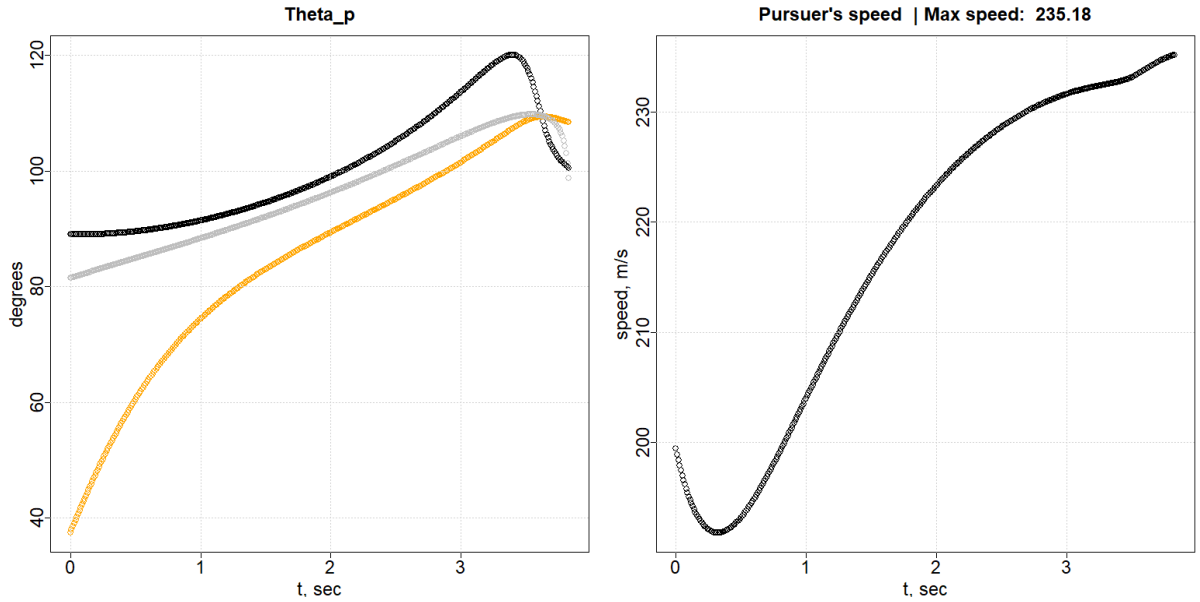

Рисунок 10 - Графіки $\theta_p(t)$ та $U(t)$ для **Гри 2**

По графіках можна побачити, що перші 1.5 с переслідувач намагався загальмувати, щоб оптимальним чином розвернутися, оскільки на початку гри він мав ненульовий вектор швидкості $U = (120, 160)$. Після розвороту через 3 с від початку гри втікач розвернувся в мінімальній точці синусоїди, по траєкторії якої він рухався, та натрапив на переслідувача. В результаті гри відбулося захоплення на моменті $t = 3.83$.

Розглянемо гру, де обидва гравці діють оптимально, тобто їх керування рівні між собою та дорівнюють $\bar{\theta}_p = \bar{\theta}_e - s_5(\tau)$ і в якості результату отримаємо мінімакс ціни гри.

Гра 3. Переслідувач P та втікач E діють оптимально. Переслідувач починає з ненульовою швидкістю, направленою в протилежний бік від втікача. Параметри та стан системи в момент $t = 0$:

$x_{p0} = 200 \quad l = 10.0$

$y_{p0} = 100 \quad w = 75.0$

$u_0 = -200 \quad F = 200.0$

$v_0 = -100 \quad m_p = 1.0$

$x_{e0} = 200 \quad k = 1.0$

$y_{e0} = 200 \quad g = 5.0$

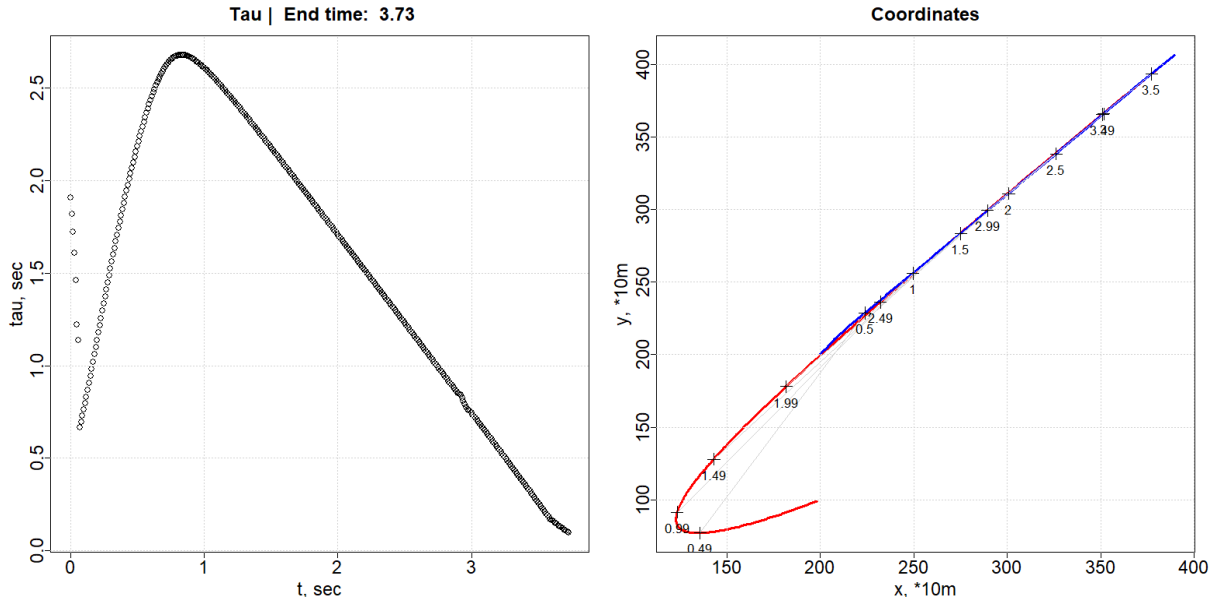

Рисунок 11 - Графіки $\tau(t)$ та $y_{p,e}(x_{p,e})$ для **Гри 3**

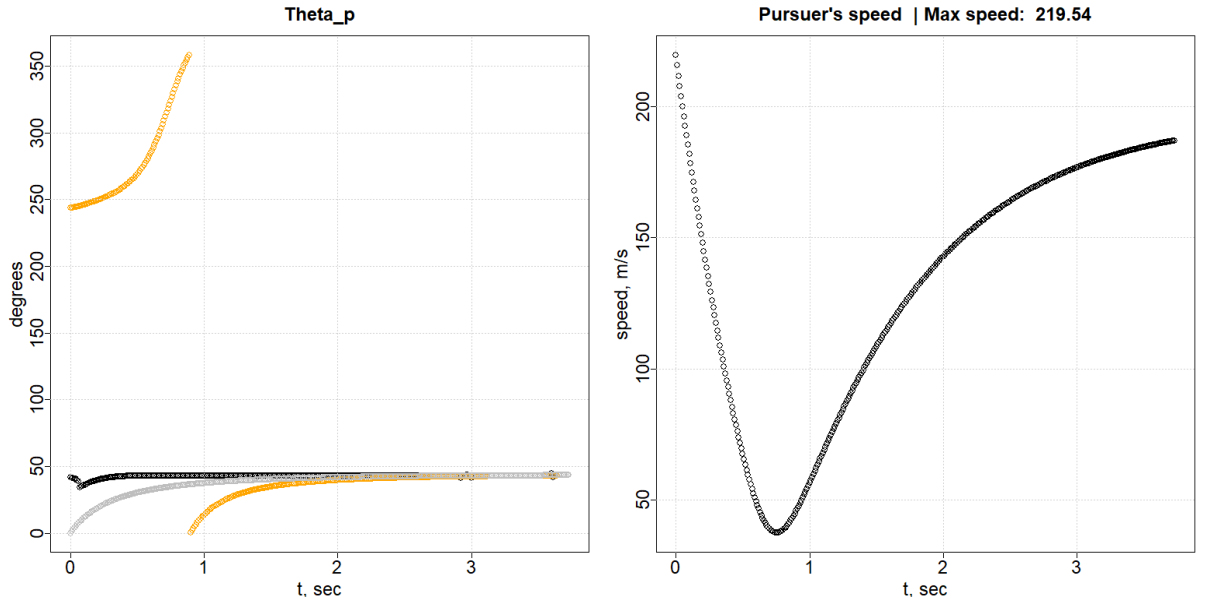

Рисунок 12 - Графіки $\theta_p(t)$ та $U(t)$ для **Гри 3**

На рисунку 11 аналізуючи графік координат, можна побачити "маневр розвороту який описується в [3]. Гра починається в точках $(x_p, y_p) = (200.0, 100.0)$ та $(x_e, y_e) = (200.0, 200.0)$, але з швидкостями направленими у різні боки. В такій ситуації оптимальне керування переслідувача змусить його швидко загальмувати, щоб знизити швидкість до такої, яка буде забезпечувати належний рівень маневровності. Це можна спостерігати на рисун-

ку 12 (графік швидкості $U(t)$). Після такого маневру переслідувач виходить на пряму рівно позаду втікача і, маючи більшу швидкість, прискорюється до моменту захоплення. В результаті гри відбулося захоплення на моменті $t = 3.73$.

Гра 4. Переслідувач P діє оптимально, втікач E діє неоптимально: протягом першого проміжку часу $t \in [0, 2)$ рухається вздовж прямої, у момент $t = 2.5$ повертає на певний кут і продовжує рух по прямій. Параметри та стан системи в момент $t = 0$:

$$x_{p0} = 50 \qquad l = 15.0$$
$$y_{p0} = 50 \qquad w = 90.0$$
$$u_0 = 0 \qquad F = 300.0$$
$$v_0 = 0 \qquad m_p = 1.0$$
$$x_{e0} = 1050 \qquad k = 1.0$$
$$y_{e0} = 350 \qquad g = 5.0$$

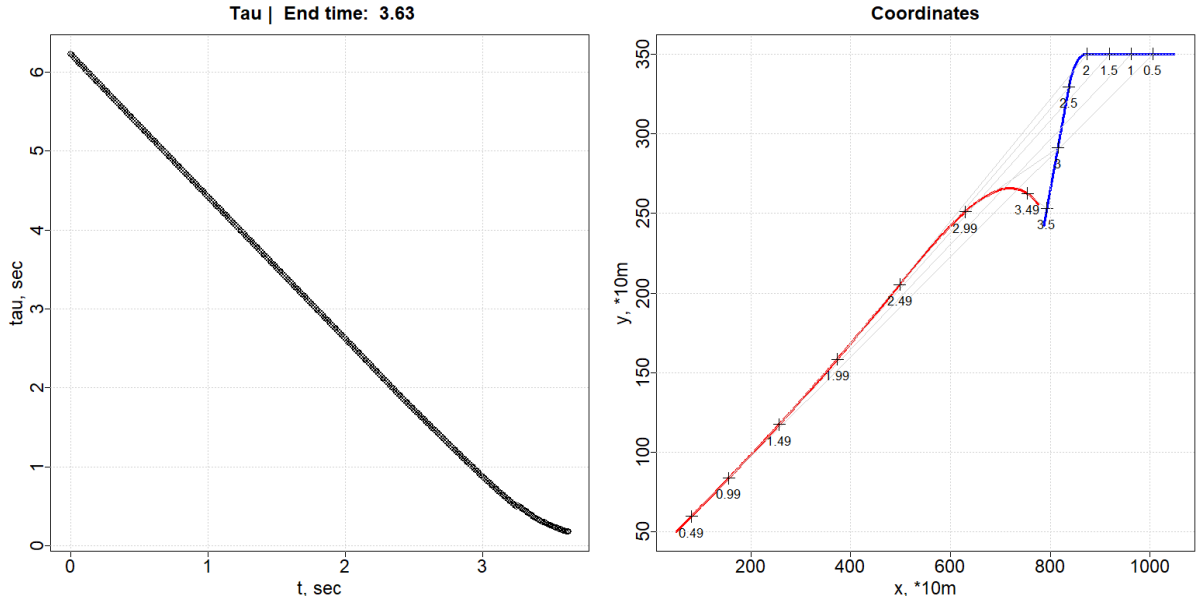

Рисунок 13 - Графіки $\tau(t)$ та $y_{p,e}(x_{p,e})$ для **Гри 4**

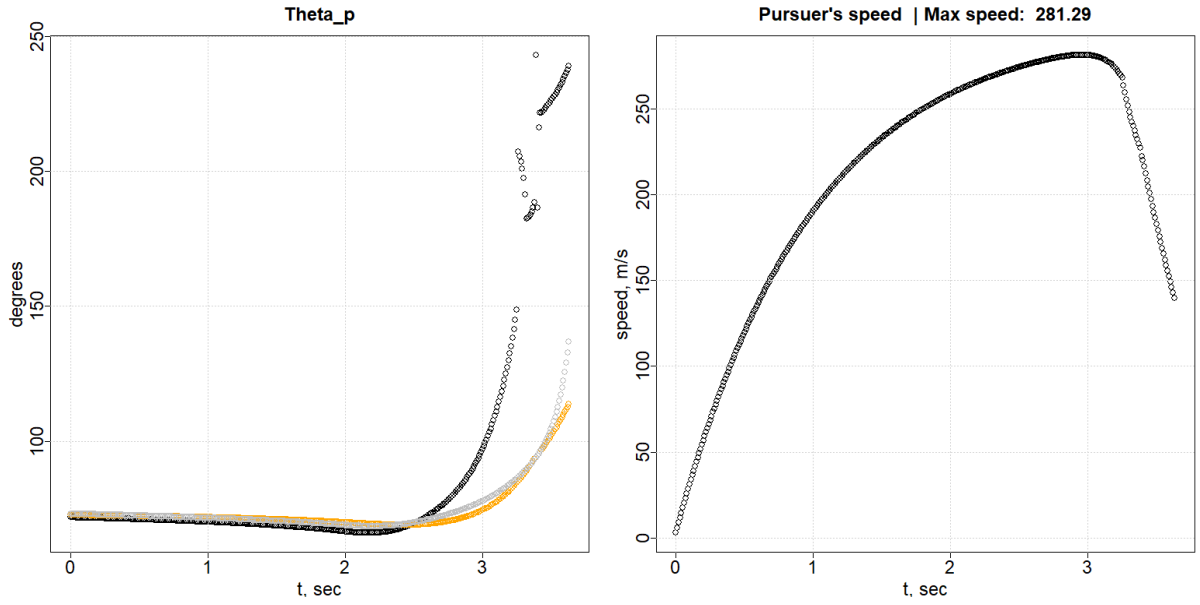

Рисунок 14 - Графіки $\theta_p(t)$ та $U(t)$ для **Гри 4**

На рисунку 13 на графіку координат можна побачити, що переслідувач до моменту $t = 2.0$ прямував на упередження втікачеві, але як тільки втікач почав змінювати свій напрямок, переслідувач почав змінювати своє керування для зміни напрямку руху, але ще певний час рухався за інерцією, що можна побачити на рисунку 14 (графік швидкості $U(t)$). В результаті гри відбулося захоплення на моменті $t = 3.63$.

Гра 5. Переслідувач P діє оптимально, втікач E діє неоптимально: протягом першого проміжку часу $t \in [0, 1)$ рухається вздовж прямої, за проміжок $t \in [1, 4]$ робить півколо і змінює свій напрям на протилежний. Параметри та стан системи в момент $t = 0$:

$$x_{p0} = 50 \quad l = 10.0$$
$$y_{p0} = 50 \quad w = 60.0$$
$$u_0 = 0 \quad F = 250.0$$
$$v_0 = 0 \quad m_p = 1.0$$
$$x_{e0} = 800 \quad k = 1.0$$
$$y_{e0} = 350 \quad g = 5.0$$

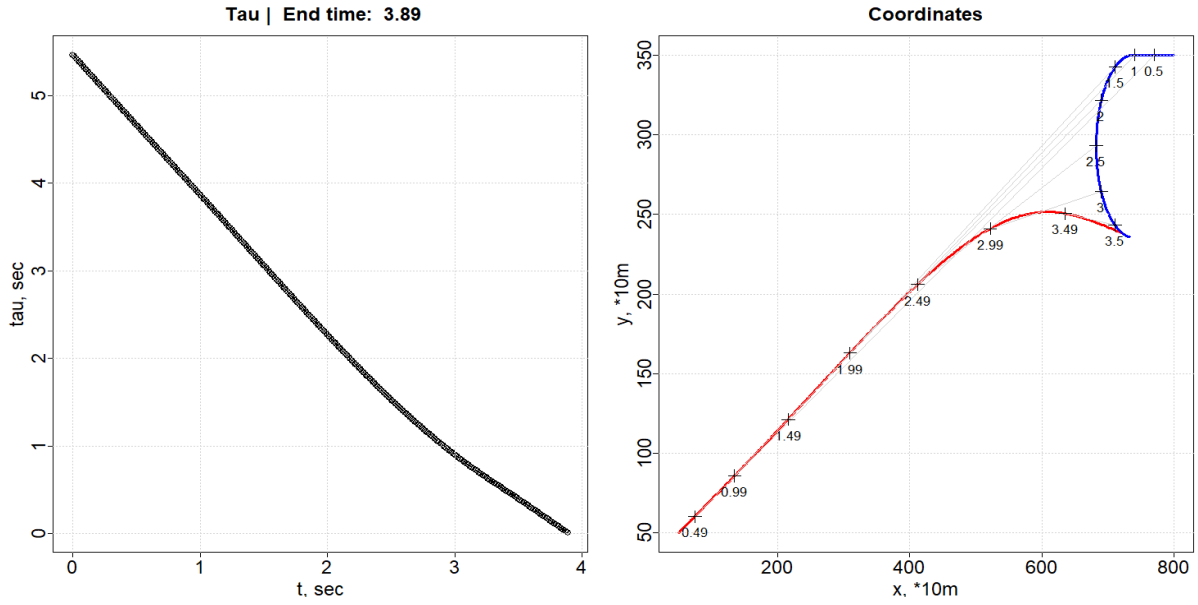

Рисунок 15 - Графіки $\tau(t)$ та $y_{p,e}(x_{p,e})$ для **Гри 5**

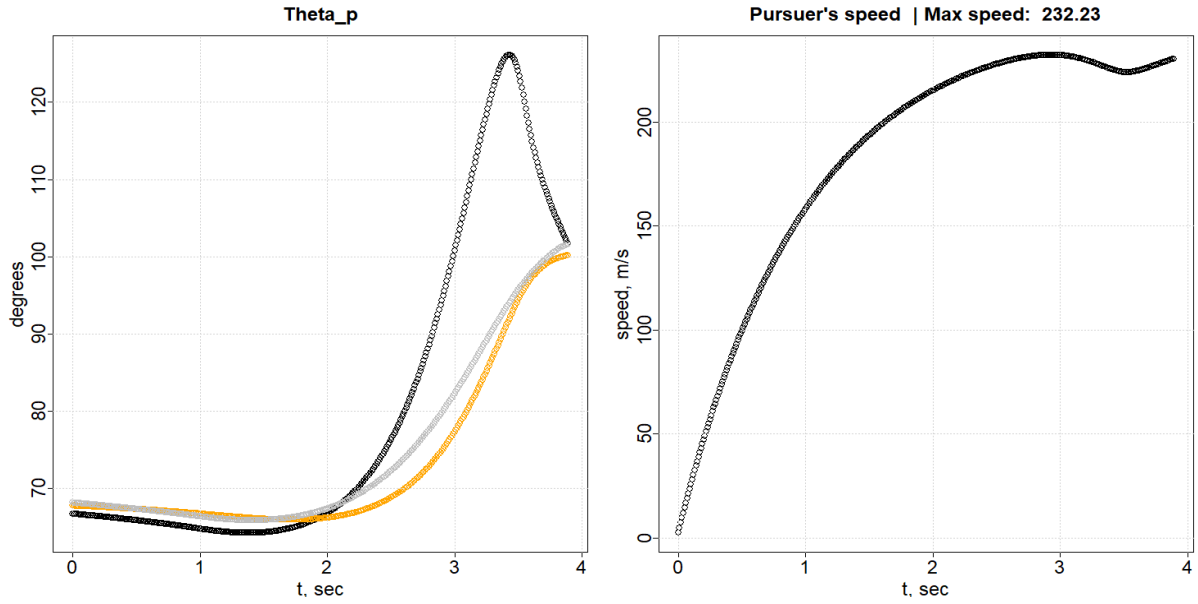

Рисунок 16 - Графіки $\theta_p(t)$ та $U(t)$ для **Гри 5**

Переслідувач увесь час гри рухається з певним упередженням до втікача і встигає його захопити ще до того, як втікач встигає завершити свій розворот. В результаті гри відбулося захоплення на моменті $t = 3.89$.

Гра 6. Переслідувач P діє оптимально, втікач E діє наближено до оптимального: $\theta_e = \theta_p - 1.5 * \sin(t)$, тобто гравець E рухається від переслідувача, використовуючи його керування як основу свого, але при цьому

незначним чином маневрує. Переслідувач починає з ненульовою швидкістю, направленою вгору. Також змінені параметри, що впливають на силу тертя та силу тяжіння, а також збільшена маса переслідувача. Параметри та стан системи в момент $t = 0$:

$$x_{p0} = 50 \quad l = 10.0$$
$$y_{p0} = 50 \quad w = 80.0$$
$$u_0 = 0 \quad F = 500.0$$
$$v_0 = 200 \quad m_p = 2.0$$
$$x_{e0} = 500 \quad k = 2.0$$
$$y_{e0} = 250 \quad g = 10.0$$

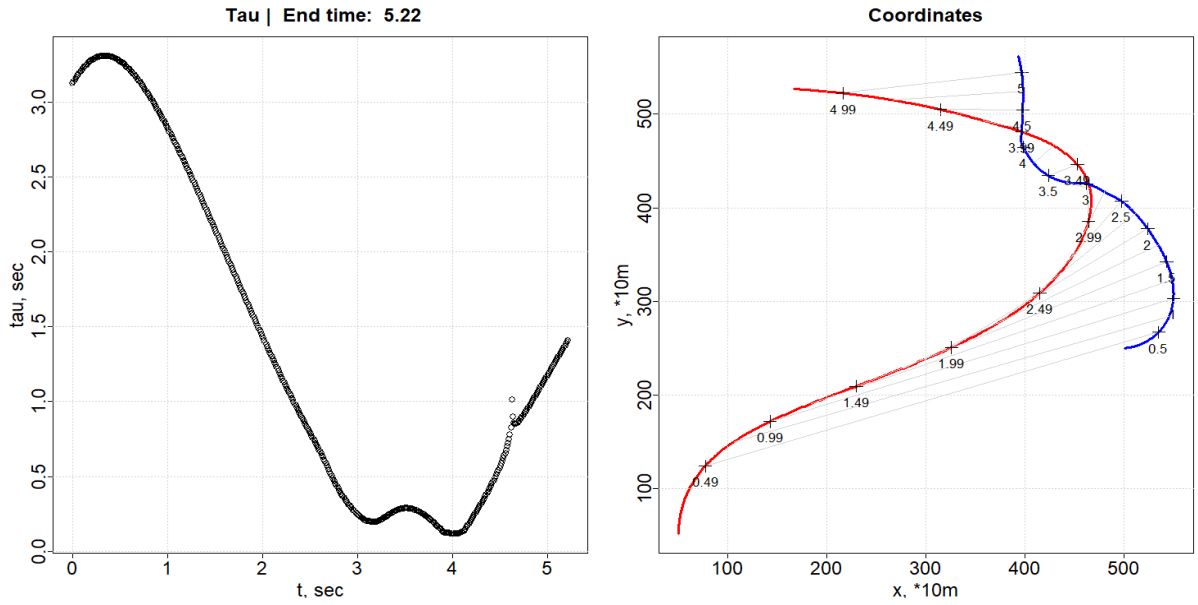

Рисунок 17 - Графіки $\tau(t)$ та $y_{p,e}(x_{p,e})$ для **Гри 6**

В результаті гри перехоплення не відбувається. В момент часу $t = 3$ втікач успішно маневрує, що робить неможливим для переслідувача захоплення у найближчий момент, оскільки переслідувач є інертним і йому не достатньо маневровності через велику швидкість в цей момент. На рисунку 17 на правому графіку координат можна побачити цей момент в точці $(x, y) = (450, 420)$, біля якої відмічений час $t = 3$. На лівому графіку можна побачити, що $\tau(t)$, що характеризує час, що залишився до моменту захоплення, починає зростати, пройшовши дві екстремальні точки в моменти часу $t = 3.1$ та $t = 4$, коли переслідувач був максимально близький до втікача. На рисунку 18 (графік $\theta_p(t)$ можна побачити злам керування. Це відбувається через те, що функція $Q(\tau)$ стає від'ємною (див. Рисунок 19),

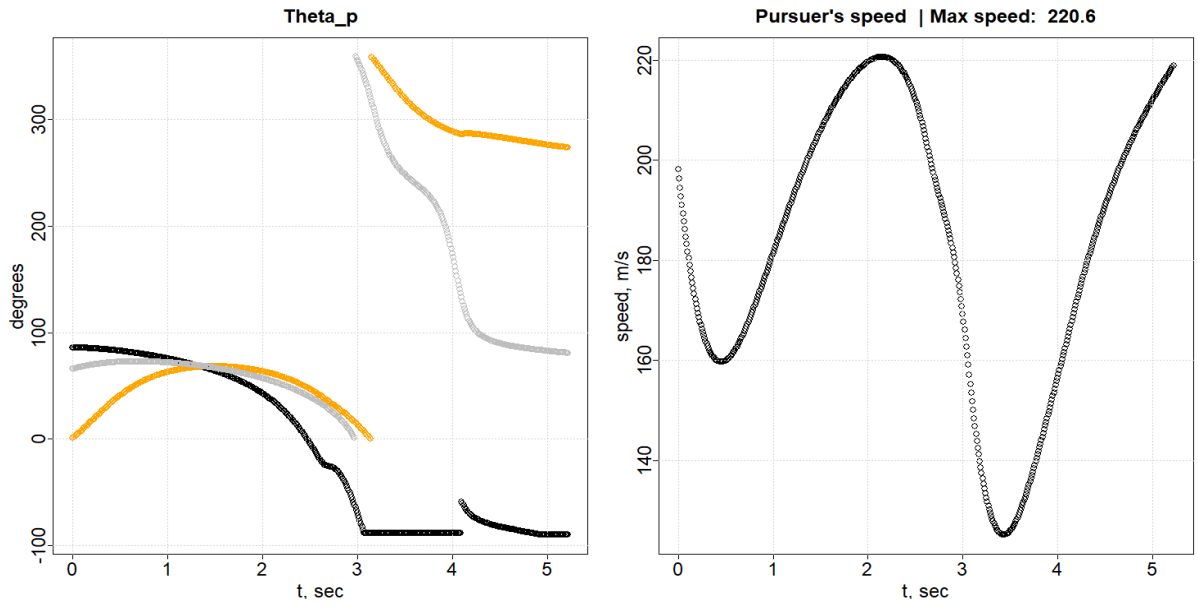

Рисунок 18 - Графіки $\theta_p(t)$ та $U(t)$ для **Гри 6**

оскільки стартові параметри не забезпечують додатного знаку для $Q(\tau)$ при всіх невід'ємних значеннях τ.

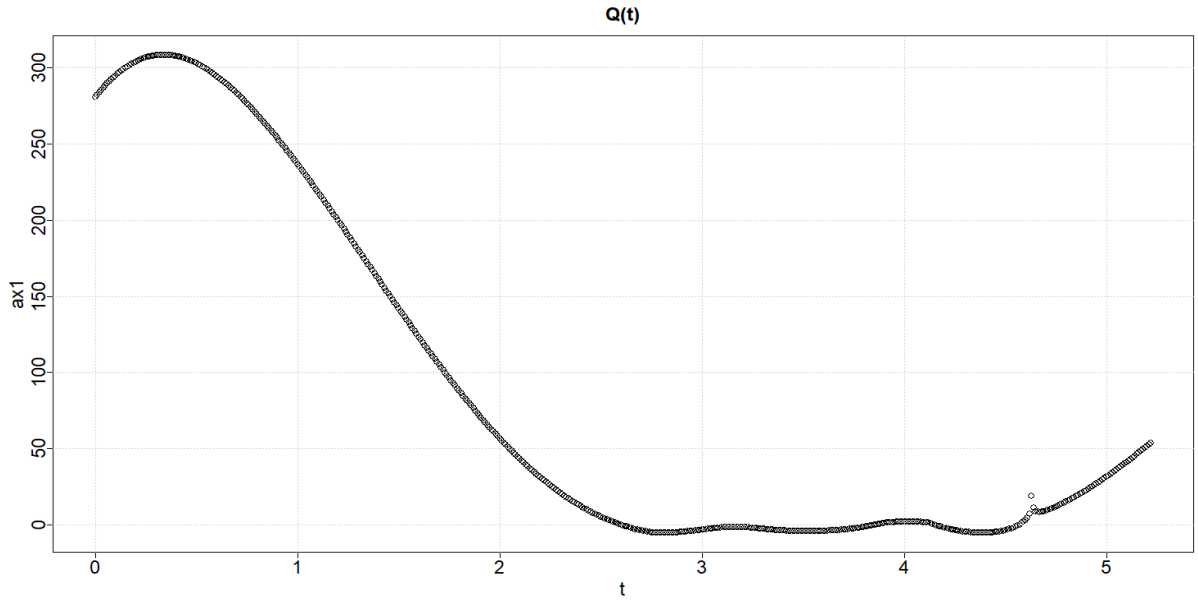

Рисунок 19 - Графік $Q(\tau)$ для **Гри 6**

4.2 Аналіз результатів проведених ігор

В рамках роботи було проведено моделювання шести ігор переслідування при різних початкових станах системи та параметрах гри. Аналізуючи

результати гри можна зробити висновок, що знайдене керування для переслідувача, яке відповідає заданій моделі руху, тримається близько до оптимального в рамках математичної моделі гри, що наближено описує рух у площині двох об'єктів з урахуванням впливу на одного з них (переслідувача) сил тяжіння та тертя. Також за результатами гри можна спостерігати, що керування переслідувача, окрім того, що будує упередження для більш оптимального перехоплення, ще "враховує"дію сил тяжіння й тертя на переслідувача.

ВИСНОВКИ

У цій роботі були поставлені мета та завдання, спрямовані на дослідження систем протиповітряної оборони, ракет протиповітряної оборони (ППО) та систем керування цими ракетами з використанням математичної задачі диференціальної гри переслідування. Головна мета роботи полягала у розв'язанні побудові та аналізі математичної моделі задачі, візуалізації результатів та аналізі отриманих даних.

В рамках виконання роботи була сформульована математична модель (розділ 3), що враховує фізичні процеси, що відбуваються під час перебігу процесу перехоплення втікача переслідувачем. Була сформульована математична задача диференціальної гри переслідування, яка дозволяє знайти оптимальні стратегії перехоплення цілей для ракет ППО з мінімальним часом переслідування.

Для розв'язання задачі був розроблений програмний код (додаток А), що допомагав здійснити чисельний розв'язок та моделювання ситуацій переслідування. Цей код включав сукупність функцій та чисельних методів, що використовувалися на певних етапах під час розв'язку задача у розділі 3. Також в рамках роботи була написана програма для візуалізації динаміки гри й аналізу результатів.

Після закінчення розв'язку задачі та побудови моделі для оптимального перехоплення та ураження було змодельовано шість ігор при різних початкових даних та був проведений їх аналіз(розділ 4).

Підсумовуючи, можна сказати, що застосування підходів диференціальних ігор для пошуку розв'язку задачі перехоплення ракетою ППО певної повітряної цілі можна вважати доцільним, але розбивка прикладної задачі на більш конкретні ситуації (різні типи повітряних цілей, різна дистанція чи здатність до маневрування) може бути ефективніша для пошуку оптимальних стратегій у конкретній конфліктній ситуації з двома гравцями, адже розширення вектора станів диференціальної гри призводило б суттєвого ускладнення процесу пошуку розв'язків задачі й подальшого прорахунку конкретних розв'язаних моделей. Саме тому ускладнена модель руху для втікача опускалася в даній роботі. В рамках заданої теми роботи для побудови та аналізу моделі оптимального перехоплення та ураження динамічних об'єктів було достатньо розглянути складніший рух тільки для

об'єкта, який перехоплює.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Література

[1] Зімін Г. В., Бутурлін Ф. Т., Бурмистров С. К., Ніздрань Я. І., Демідов В. П., Мальгін А. С., Неупокоєв Ф. К., Орлов О. Е., Довідник офіцера протиповітряної оборони, - Ленінград: Військове видавництво, 1981. - 431 сторінки.

[2] А. А. Чикрій, Конфліктно керовані процеси, - Київ, 1992.

[3] Р. Айзекс, диференціальні ігри, - Москва, 1967.

[4] М. Г. Котик, Динаміка зльоту і посадки літаків, - Москва, 1984.

[5] Н. Н. Красовський, Ігрові задачі про зустріч рухів, - Москва, 1970.

[6] Документація бібліотеки SciPy. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.fsolve.html#scipy.optimize`

[7] Графічний калькулятор на вебресурсі. `https://www.desmos.com/calculator`

ДОДАТОК А

Код програми на Python

Програмний код, який використовувався для розв'язку задачі у розділі 3. У ньому реалізовані функції для розрахунку проміжних значень, а також використані гібридні чисельні методи [6] для пошуку нетривіальних коренів в рамках розв'язку задачі. Також цей програмний код візуалізує перебіг гри переслідування у динаміці. Написаний мовою програмування Python. Реалізований у програмному середовищі PyCharm.

```
import numpy as np
from scipy.optimize import fsolve, root, newton_krylov,
broyden2, newton, brentq, bisect
import pygame
import math
import csv

# Params Pursuer
x_start_p = 50.0  # *10
y_start_p = 50.0  # *10
u = 0.0
v = 200.0
F = 10000.0  #
mp = 2.0 #
k = 2.0  #               : k = 0.5*C_d*A*ro*V  =
0.5*0.6*0.0491*1.2255*700
g = 10.0  # / 2
theta_p = math.radians(math.pi / 2)
prev_theta_p = math.radians(0)
v_angle = 0

# Params Evader
x_start_e = 500.0  # *10
y_start_e = 250.0 # *10
```

```
w = 200.0  # /
theta_e = math.radians(0)

# Game constants
p_m = 10  #
t = 0.0  #
tau = 0.0
prev_tau = 5
t_multiplier = 10  #
x_width = 1100.0  # *10
y_height = 600.0  # *10
l = 5.0 #
end_game_event = False
angle_to_e = 0

# Draw constants
color1 = (255, 0, 0)
color2 = (0, 0, 255)
line_color_p = (0, 0, 0)
line_color_e = (0, 0, 255)
line_color_p_v = (255, 128, 0)
line_width = 1

#Auxiliary variables
ktmp, E1, A, Q, exp_term = np.float64(0.0), np.float64(0.0),
np.float64(0.0), np.float64(0.0), np.float64(0.0)
r_vector = np.float64(0.0)
u_vector = np.float64(0.0)

#Some variables for output in .csv file
ax1, ax2, ax3, ax_mean = 0.0, 0.0, 0.0, 0.0

def equation_tau(tau):
    global E1, A, Q
```

```
    ktmp = k * tau / mp
    exp_term = np.exp(-ktmp)
    E1 = mp / k * (1 - exp_term)
    A = g * mp * mp * (exp_term + ktmp - 1) / (k * k)
    Q = l - w * tau + (F * mp - g * mp * mp) * (exp_term + ktmp
    - 1) / (k * k)
    eq = r_vector*(r_vector - 2*u_vector*E1) + (u_vector*E1)**2
    + A*(A - 2*(ye-y-v*E1)) - (Q+A)**2
    return eq

def find_tau(x, y, xe, ye, u, v, w):
    global ax1, ax2, ax3
    global r_vector, u_vector
    global prev_tau
    global ktmp, E1, A, Q, exp_term

    r_vector = math.sqrt((xe - x) * (xe - x) + (ye - y) * (ye -
    y))
    u_vector = math.sqrt(u * u + v * v)

    result = fsolve(equation_tau, prev_tau)[0]
    prev_tau = result

    print("result tau: ", result)

    ax1 = Q[0]

    return result

def find_theta_p():
    global x, y, xe, ye, u, v, w
    global tau
    global ax_mean, ax2, ax3
    global ktmp, E1, A, Q, exp_term
```

```
    x, y = cx(pursuer[0]), cy(pursuer[1])
    xe, ye = cx(evader[0]), cy(evader[1])

    tau = find_tau(x, y, xe, ye, u, v, w)

    m = (xe - x - u*mp*E1/k)[0]
    n = (ye - y - v*mp*E1/k + A)[0]

    ax2 = m
    ax3 = n

    #print("m: ", round(m,3), "| n: ", round(n,3), "| m/n: ",
    round(m/n,3))

    s5 = math.atan(m / n)
    return s5

def equation(s5):
    m = (xe - x - u*mp*E1/k)[0]
    n = (ye - y - v*mp*E1/k + A)[0]

    return math.tan(s5) - m/n

def equation1(s5):

    return math.tan(s5) - (cx(evader[0]) - cx(pursuer[0])
    u*tau)/(cy(evader[1]) - cy(pursuer[1])-v*tau)

def equation2(s5):

    return math.tan(1/s5) - (cy(evader[1]) - cy(pursuer[1])
    v*tau)/(cx(evader[0]) - cx(pursuer[0])-u*tau)
```

```
def calculate_angle(A, B):
    angle = math.atan2((B[0] - A[0]), (B[1] - A[1]))
    if angle < 0:
        angle += 2 * math.pi

    return angle

def cx(x):
    return x

def cy(y):
    return y_height - y

def distance(x1, y1, x2, y2):
    x_diff = x2 - x1
    y_diff = y2 - y1
    return math.sqrt(x_diff ** 2 + y_diff ** 2)

def est_time_p(x, y):
    return distance(x, y, cx(pursuer[0]), cy(pursuer[1])) /
    math.sqrt(v*v+u*u)

def init_game():
    pygame.init()
    screen = pygame.display.set_mode((x_width, y_height))
    pygame.display.set_caption("           ")

    with open('data.csv', 'w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(
            ['t', 'cx(pursuer[0])', 'cy(pursuer[1])', 'u', 'v',
            'theta_p', 'v_angle', 'angle_to_e', 'tau',
            'cx(evader[0])',
            'cy(evader[1])','ax_mean', 'ax1', 'ax2', 'ax3'])
```

```
    return screen

def set_starting_conditions():
    global t
    global t_multiplier
    global pursuer
    global pursuer_speed
    global evader
    global target_point
    global theta_p
    global prev_theta_p
    t = 0

    pursuer = [cx(x_start_p), cy(y_start_p)]
    evader = [cx(x_start_e), cy(y_start_e)]
    pursuer_speed = [0, 0]
    launcher_angle = calculate_angle((cx(pursuer[0]),
    cy(pursuer[1])),
                                     (cx(evader[0]),
                                     cy(evader[1])))  #
                                     math.radians(45)
    theta_p = launcher_angle
    prev_theta_p = theta_p

def main_loop(screen):
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                if end_game_event:
                    display_success(screen)
                pygame.quit()
                quit()

        screen_fill(screen)
```

```
        draw_points(screen)
        draw_trajectory(screen)
        check_end_game()

        update_coordinates()
        update_screen()
        update_time()

def screen_fill(screen):
    screen.fill((255, 255, 255))

def draw_points(screen):
    pygame.draw.circle(screen, color1, pursuer, 2)
    pygame.draw.circle(screen, color2, evader, 2)

def draw_trajectory(screen):
    pygame.draw.line(screen, line_color_p, (pursuer[0],
    pursuer[1]),
                     (pursuer[0] + 100 * math.sin(theta_p),
                     pursuer[1] - 100 * math.cos(theta_p)),
                     line_width)
    pygame.draw.line(screen, (211, 211, 211), (pursuer[0],
    pursuer[1]),
                     (pursuer[0] + 10000 *
                     math.sin(angle_to_e), pursuer[1] - 10000 *
                     math.cos(angle_to_e)),
                     line_width)
    pygame.draw.line(screen, line_color_e, (evader[0],
    evader[1]),
                     (evader[0] - 50 * math.sin(theta_e
                     math.pi), evader[1] + 50 *
                     math.cos(theta_e-math.pi)),
                     line_width)
    pygame.draw.line(screen, line_color_p_v, (pursuer[0],
```

```
    pursuer[1]),
                    (pursuer[0] + 50 * math.sin(v_angle),
                    pursuer[1] - 50 * math.cos(v_angle)),
                    line_width)

def check_end_game():
    global end_game_event
    if abs(pursuer[0] - evader[0]) <= 1 and abs(pursuer[1] -
    evader[1]) <= 1:
        end_game_event = True
        quit_event = pygame.event.Event(pygame.QUIT)
        pygame.event.post(quit_event)

def calculate_evader():
    global evader, theta_e
    dt = t_multiplier / 1000

    xe = cx(evader[0])
    ye = cy(evader[1])

    #if t < 1:
    #    theta_e = math.radians(270) # - math.sin(t)
    #else:
    #    if t>=1 and t<=4:
    #        theta_e =  math.radians(270 - (t - 1) * (180 / (4 - 1)))
    #    else:
    #        theta_e = math.radians(90)

    theta_e = theta_p - 1.5*math.sin(t)

    xe += w * math.sin(theta_e) * dt
    ye += w * math.cos(theta_e) * dt

    evader[0] = cx(xe)
```

```
    evader[1] = cy(ye)

def calculate_pursuer():
    global pursuer
    global u, v
    global theta_p
    dt = t_multiplier / 1000
    ektmp = np.exp(-k*dt/mp)

    x = cx(pursuer[0])
    y = cy(pursuer[1])
    x += F/k * math.sin(theta_p) * dt - ektmp
    (F/k*math.sin(prev_theta_p)-u)*dt
    y += F/k * math.cos(theta_p) * dt - ektmp
    (F/k*math.cos(prev_theta_p)-v)*dt - g*mp*(1-ektmp/k)*dt

    u += F/k * math.sin(theta_p) * dt - k*u/mp*dt
    v += F/k * math.cos(theta_p) * dt - k*v/mp*dt - g*dt
    pursuer[0] = cx(x)
    pursuer[1] = cy(y)

def update_coordinates():
    global theta_e
    global theta_p
    global pursuer
    global pursuer_speed
    global evader
    global target_point
    global angle_to_e
    global tau, prev_tau
    global prev_theta_p
    global file
    global v_angle
```

```
calculate_evader()

angle_to_e = calculate_angle((cx(pursuer[0]),
cy(pursuer[1])), (cx(evader[0]), cy(evader[1])))

theta_p = find_theta_p()

if abs(v_angle - angle_to_e) >= 3*math.pi / 4:
    prev_tau = 3
if abs(prev_theta_p-theta_p) >= math.pi/4:
    if abs(prev_theta_p - theta_p - math.pi) >= math.pi / 4:
        theta_p = prev_theta_p

prev_theta_p = theta_p

v1_prev = cx(pursuer[0])
v2_prev = cy(pursuer[1])

calculate_pursuer()

v_angle = calculate_angle((v1_prev, v2_prev),
(cx(pursuer[0]), cy(pursuer[1])))

#                    .csv
with open('data.csv', 'a', newline='') as file:
    writer = csv.writer(file)
    writer.writerow([t, cx(pursuer[0]), cy(pursuer[1]), u,
    v, theta_p, v_angle, angle_to_e, tau, cx(evader[0]),
cy(evader[1]), ax_mean, ax1, ax2, ax3])
#print("coord: ", pursuer[0], pursuer[1])
#print("v_angle: ", math.degrees(v_angle))
#print("theta_p: ", math.degrees(theta_p))
#print("tau: ", tau)
print("-------------------")
```

```
def update_screen():
    pygame.display.update()
    pygame.time.delay(t_multiplier)

def update_time():
    global t
    t += t_multiplier / 1000  #
    (1000 )       1000/t_multiplier

def display_success(screen):
    file.close
    text_pos_x = x_width / 2
    text_pos_y = y_height / 2
    font_size = 32
    font = pygame.font.Font(None, font_size)
    text = "Success!"
    color_text = (0, 0, 0)
    text_surface = font.render(text, True, color_text)
    screen.blit(text_surface, (text_pos_x, text_pos_y))
    pygame.display.flip()
    pygame.time.delay(5000)

if __name__ == "__main__":
    screen = init_game()
    set_starting_conditions()
    main_loop(screen)
```

Код програми на R

Наведений програмний код використовувався для аналізу результатів змодельованих ігор та для побудови графіків з розділу 4. Написаний мовою R. Реалізований у програмному середовищі RStudio.

```
data <- read.csv("data.csv")

m_r_v = 2
m_r_h = 2
cex_par = 0.7
par()
par(mfrow=c(1,2))
par(mar = c(m_r_v, m_r_h, m_r_v, m_r_h-1))
par(mgp = c(1.1, 0.15, 0))
par(tck = -0.01)
par(cex.axis = 1)
par(cex.main = 1)

par_t = max(data$t)
t <- data$t[data$t<par_t]
tau_array <- data$tau[data$t<par_t]

xp <- data$cx.pursuer.0..[data$t<par_t]
yp <- data$cy.pursuer.1..[data$t<par_t]

xe <- data$cx.evader.0..[data$t<par_t]
ye <- data$cy.evader.1..[data$t<par_t]

u <- data$u[data$t<par_t]
v <- data$v[data$t<par_t]

theta_p <- data$theta_p[data$t<par_t]*(180/pi)
v_angle <- data$v_angle[data$t<par_t]*(180/pi)
angle_to_e <- data$angle_to_e[data$t<par_t]*(180/pi)

ax1 <- data$ax1[data$t<par_t]
ax2 <- data$ax2[data$t<par_t]
ax3 <- data$ax3[data$t<par_t]
```

```
#Tau plot
end_time_text <- paste("End time: ", round(max(t),2))
plot(t,tau_array,cex=cex_par, ylab="tau, sec",xlab="t, sec",
main = paste(" Tau | ",end_time_text))
grid()

#Coordinates plot
plot(xp,yp,type='l',col="red",ylab="y, *10m",xlab="x, *10m",
lwd = 2, main = paste("Coordinates"),
     ylim=c(min(c(yp,ye)),max(c(yp,ye))),
     xlim=c(min(c(xp,xe)),max(c(xp,xe))))
lines(xe,ye,type='l',col="blue", lwd = 2)
indices <- seq(50, length(xp), by = 25)
segments(xp[indices], yp[indices], xe[indices], ye[indices],
col = "lightgray", lwd = 0.001)
indices <- seq(50, length(xp), by = 50)
points(xp[indices], yp[indices], pch = 3, cex = 1, col =
"black")
text(xp[indices], yp[indices], labels = round(t[indices],2),
pos = 1, cex = 0.7)
points(xe[indices], ye[indices], pch = 3, cex = 1, col =
"black")
text(xe[indices], ye[indices], labels = round(t[indices],1),
pos = 1, cex = 0.7)
grid()

#Theta_p plot
plot(t,theta_p,cex=cex_par, main = paste("Theta_p"),
ylab="degrees",xlab="t, sec", ylim =
c(min(theta_p,v_angle),max(theta_p,v_angle)))
points(t,v_angle, cex=cex_par,col="orange")
points(t,angle_to_e, cex=cex_par,col="gray")
grid()
```

```
#Speed plot
p_speed <- sqrt(u^2+v^2)
main_text_speed <- paste("Pursuer's speed ",paste("| Max speed:
",round(max(p_speed),2)))
plot(t,p_speed,cex=cex_par, main = main_text_speed,
ylab="speed, m/s",xlab="t, sec")
grid()

#Auxiliary plot
#plot(t,u,cex=cex_par, main = paste("5TH | U part of P's
speed"))
#grid()

#Auxiliary plot
#plot(t,v,cex=cex_par, main = paste("6TH | V part of P's
speed"))
#grid()

#Auxiliary plot
plot(t,ax1,cex=cex_par, main = paste("Q(t)"))
grid()

#Auxiliary plot
#ax2_lim = max(abs(min(ax2)),ax2)
#plot(t,ax2,cex=cex_par, main = paste("Ax2 (m from atan arg)"),
ylim = c(-ax2_lim,ax2_lim))
#grid()

#Auxiliary plot
#ax3_lim = max(abs(min(ax3)),ax3)
#plot(t,ax3,cex=cex_par, main = paste("Ax3 (n from atan arg)"),
ylim = c(-ax3_lim,ax3_lim))
#grid()
```

```
#Auxiliary plot
#plot(t,ax2/ax3,cex=cex_par, main = paste("Ax (m/n from atan
arg)"), ylim = c(-ax3_lim,ax3_lim))
#grid()

#Calculating auxiliary meanings

t_par = 400:450
theta_p[t_par]

round(u[t_par],1)
round(v[t_par],1)
round(tau_array[t_par],1)
round(xe[t_par],1)
round(xp[t_par],1)
round(ye[t_par],1)
round(yp[t_par],1)
round(ax2[t_par],3)
round(ax3[t_par],3)
p1 = round(ax2[t_par]/ax3[t_par],3)
180+atan(p1)*180/pi
2*180+atan(8.7/-0.5)*180/pi
which.max(theta_p[400:length(theta_p)])

par(mfrow=c(1,1))


#Finding point T

T_x = xp + ax2
T_y = yp + ax3

#plot(xp,yp,type='l',col="red", main = paste("2ND |
Coordinates"),
```

```
#     ylim=c(min(c(yp,ye)),max(c(yp,ye,ax2,ax3))),
xlim=c(min(c(xp,xe)),max(c(xp,xe))))
#lines(xe,ye,type='l',col="blue")
#points(T_x,T_y, type = 'l', col="orange", cex = cex_par)
#indices <- seq(50, length(xp), by = 25)
#segments(xp[indices], yp[indices], xe[indices], ye[indices],
col = "lightgray", lwd = 0.001)
#indices <- seq(50, length(xp), by = 50)
#points(xp[indices], yp[indices], pch = 3, cex = 0.5, col =
"black")
#text(xp[indices], yp[indices], labels = round(t[indices],2),
pos = 1, cex = 0.5)
#points(xe[indices], ye[indices], pch = 3, cex = 0.5, col =
"black")
#text(xe[indices], ye[indices], labels = round(t[indices],1),
pos = 1, cex = 0.5)
#points(T_x[indices], T_y[indices], pch = 3, cex = 0.5, col =
"black")
#text(T_x[indices], T_y[indices], labels = round(t[indices],2),
pos = 1, cex = 0.5)
#grid()
```