

М.В. Богдановський, ст. викладач
О.В. Підтиченко, к.т.н.
Д.П. Горбик, здобувач

Державний університет «Житомирська політехніка»

Розробка програмного симулятора дослідження задач кінематики та управління роботизованим маніпулятором

Кінематика є невід'ємною частиною дослідження при проєктуванні конструкцій промислових роботів та розробки системи управління. Предметом навчання при цьому виступають підходи та математичні методи опису просторового положення елементів кінематичного ланцюга, кінцевою метою яких є визначення узагальнених координат за положенням та орієнтацією виконавчого органу. В роботі представлено розробку програмного симулятора для розв'язання задач кінематики на прикладі навчального робота «Braccio robot arm», який наочно демонструє математичний апарат. Для прикладу ділянки кінематичного ланцюга з трьох ланок, що формують неоднозначність розв'язання зворотної задачі кінематики продемонстровано використання чисельного методу згасаючих найменших квадратів (Лівенберга – Маркварда) для пошуку усіх доступних конфігурацій, необхідних для ефективного планування руху робота. Продемонстровано реалізацію програмної частини у середовищі Visual Studio з використанням технології Windows Forms на мові C++/CLI. Описано математичний апарат прямої кінематичної задачі на основі матриць однорідних перетворень (4×4), застосування методу Лівенберга – Маркварда з аналітичним обчисленням матриці Якобі, структуру програмного забезпечення, інтерфейс користувача та результати експериментальної перевірки. В ході виконання роботи доведено практичну стійкість результатів та практичну значимість для подальших задач розробки системи управління роботизованим маніпулятором. Окреслені перспективи перенесення математичного ядра симулятора у середовище Unity 3D в межах концепції цифрового двійника фізичного маніпулятора для відтворення програмного управління.

Ключові слова: промислові роботи; симулятор; STEM освіта; кінематика; проєктування; яacobian; моделювання.

Актуальність роботи. Сучасний розвиток робототехніки та мехатроніки характеризується стрімким впровадженням автоматизованих систем у промисловість, медицину, освіту та інші галузі діяльності людини. Одним із ключових елементів таких систем є роботизовані маніпулятори, які забезпечують виконання складних технологічних операцій з високою точністю та повторюваністю. Ефективне керування маніпуляторами неможливе без глибокого розуміння їх кінематичних властивостей, зокрема взаємозв'язку між узагальненими координатами (кутами суглобів) та положенням виконавчого органу у робочому просторі.

У навчальному процесі вивчення кінематики маніпуляторів традиційно базується на аналітичних методах та теоретичних викладках, що часто ускладнює сприйняття матеріалу студентами, особливо на початкових етапах підготовки. Відсутність наочних інструментів для експериментального дослідження кінематичних залежностей обмежує можливості практичного закріплення знань. У зв'язку з цим актуальною задачею є створення інформаційно-програмних засобів, що наочно демонструють сучасні підходи та математичний апарат розв'язання прикладних задач з можливістю інтерактивної перевірки результатів, як елемент STEM освіти при вивченні робототехніки.

Аналіз останніх досліджень та публікацій. Роботизовані маніпулятори є одним із популярних видів технічних систем у сучасній автоматизації виробництва та дослідницьких процесах, пов'язаних із моделюванням характеристик та управлінням [1, 2]. Вони призначені для виконання різноманітних операцій, таких як переміщення об'єктів, обробка матеріалів, складання виробів, а також виконання допоміжних задач у складних або небезпечних умовах. Основною особливістю маніпуляторів є наявність кінематичного ланцюга, що складається з послідовно з'єднаних ланок і суглобів, які забезпечують переміщення кінцевого виконавчого органу у робочому просторі.

З точки зору кінематики, сучасний маніпулятор промислового робота складається з скінченної кількості ланок 5 класу, кожна з яких описується узагальненою координатою (кутом повороту або лінійним переміщенням) [3]. Для забезпечення заданого положення та орієнтації виконавчого органу в робочій зоні кількість ланок більшості роботизованих маніпуляторів у виробництві варіюється від 4 до 6, що приводить до нелінійного розв'язку зворотної задачі кінематики, що немає єдиного аналітичного рішення [3, 4].

Особливістю зворотної задачі кінематики є наявність множини розв'язків або їх відсутність. Для заданих положення та орієнтації виконавчого органу маніпулятор може мати декілька конфігурацій.

У процесі переміщення маніпулятор може потрапляти у сингулярні положення, в яких втрачається керованість або різко погіршується чисельна стабільність розв'язку [4].

У сучасних інженерних практиках для розв'язання задач зворотної кінематики широко використовуються чисельні методи, що базуються на ітераційних алгоритмах [3, 5]. Одним із найбільш ефективних підходів є використання матриці Якобі, яка описує залежність між зміною узагальнених координат та зміною положення виконавчого органу. На основі цієї матриці формуються алгоритми, що дозволяють поступово наблизитися до заданого положення шляхом корекції кутів суглобів.

Серед чисельних методів особливе місце займає метод Левенберга – Марквардта, який є модифікацією методу найменших квадратів із додатковим демпфуванням [6]. Його перевага полягає у поєднанні властивостей методу градієнтного спуску та методу Гауса–Ньютона, що забезпечує як стійкість при далеких від розв'язку початкових наближеннях, так і швидку збіжність у безпосередній близькості до оптимального рішення. Саме ця властивість робить його придатним для застосування у задачах керування маніпуляторами в реальному часі.

Окрім математичних аспектів, важливою складовою предметної області є програмна реалізація моделей маніпуляторів [7]. Для навчальних і дослідницьких цілей доцільно використовувати середовища, що дозволяють поєднати обчислення з візуалізацією, що дозволяє передбачати ситуації, зокрема безпеку руху [7, 8]. Так, наприклад, застосування платформи Windows Forms, забезпечує достатній рівень інтерактивності та дозволяє швидко реалізувати графічний інтерфейс користувача. Візуалізація руху маніпулятора у площині дозволяє наочно спостерігати результати обчислень і аналізувати поведінку алгоритмів [8]. Сучасні тенденції розвитку робототехнічних систем передбачають перехід до тривимірного моделювання та використання цифрових двійників [9, 10]. У цьому контексті важливим є поєднання CAD-систем, для створення параметричних моделей та рушіїв для реалізації інтерактивної симуляції. Такий підхід дозволяє не лише відобразити геометрію маніпулятора, але й реалізувати його поведінку у динамічному середовищі, включаючи взаємодію з іншими об'єктами та системами [11].

Важливим STEM компонентом викладання є демонстрація результатів роботи віртуальної моделі на фізичному стенді. Наявність реального роботизованого пристрою, керованого мікроконтролером, відкриває можливість створення системи, у якій обчислення виконуються у програмному середовищі, а результати передаються на апаратний рівень для фізичного виконання. Така інтеграція є важливим кроком до побудови кіберфізичних систем, у яких цифрова та фізична компоненти працюють узгоджено [9].

Аналіз існуючих підходів показує, що поєднання чисельних методів розв'язання задач зворотної кінематики з інтерактивними симуляторами є ефективним інструментом для навчання та дослідження [3–6]. Отримані результати можуть бути використані як основа для подальшого розвитку більш складних систем, зокрема багатоланкових просторових маніпуляторів та колективної роботи декількох роботів у спільному робочому середовищі.

Постановка завдання. Метою даної роботи є розробка та верифікація програмного симулятора кінематики роботизованого маніпулятора, що ілюструє послідовність математичних перетворень та чисельний метод розв'язання оберненої задачі кінематики. Наочно представити використання матриць однорідних перетворень систем координат 4×4 та продемонструвати використання методу Левенберга – Марквардта з аналітичним обчисленням матриці Якобі. Для візуалізації траєкторій використати лінійну інтерполяцією між вузловими конфігураціями, а також підготовка математичну бібліотеку до перенесення у середовище 3D в рамках концепції цифрового двійника фізичного маніпулятора.

Викладення основного матеріалу. Пряма кінематика є базовою задачею в робототехніці, яка полягає у визначенні положення виконавчого органу маніпулятора за відомими значеннями узагальнених координат – кутів обертання його суглобів. На прикладі триланкового маніпулятора ця задача має чіткий аналітичний розв'язок і є відправною точкою для подальшого розуміння більш складних процесів, зокрема зворотної кінематики.

Розглянемо маніпулятор, що складається з трьох послідовно з'єднаних ланок довжинами L_1, L_2, L_3 , які обертаються на кути $\theta_1, \theta_2, \theta_3$ відповідно. Всі обертання відбуваються у площині XY, що дозволяє розглядати задачу у двовимірному просторі.

Кінцеве положення маніпулятора визначається як сума векторів усіх ланок з урахуванням їхніх кутів повороту. Для цього вводяться накопичені кути:

$$\begin{aligned}\varphi_1 &= \theta_1, \\ \varphi_2 &= \theta_1 + \theta_2, \\ \varphi_3 &= \theta_1 + \theta_2 + \theta_3.\end{aligned}\tag{1}$$

Тоді координати виконавчого органу визначаються наступними рівняннями:

$$\begin{aligned}x &= L_1 \cos(\varphi_1) + L_2 \cos(\varphi_2) + L_3 \cos(\varphi_3), \\ y &= L_1 \sin(\varphi_1) + L_2 \sin(\varphi_2) + L_3 \sin(\varphi_3).\end{aligned}\tag{2}$$

Формули (1) та (2) безпосередньо відображають геометричну суть задачі: кожна наступна ланка додає свій внесок до загального положення, орієнтуючись відносно попередньої.

Для більш глибокого розуміння структури перетворень доцільно використовувати матричний підхід. Кожна ланка описується як композиція обертання та перенесення. Обертання навколо осі Z у площині XY задається матрицею:

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

Перенесення вздовж осі X на довжину ланки описується матрицею:

$$T_x(L) = \begin{bmatrix} 1 & 0 & 0 & L \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

Повне перетворення для кожної ланки формується як добуток відповідних матриць. Вирази однорідних перетворень для трьох послідовно з'єднаних ланок наступні:

$$\begin{aligned} T_1 &= R_z(\theta_1), \\ T_2 &= T_1 \cdot T_x(L_1) \cdot R_z(\theta_2), \\ T_3 &= T_2 \cdot T_x(L_2) \cdot R_z(\theta_3). \end{aligned} \quad (5)$$

Кінцева точка визначається як результат застосування матриці T_3 до локальної точки кінця третьої ланки:

$$p_{EE} = T_3 \cdot \begin{bmatrix} L_3 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (6)$$

Даний підхід дозволяє не лише отримати координати виконавчого органу, але й описати орієнтацію кожної ланки у просторі, що є важливим для подальшого переходу до тривимірних моделей.

Таким чином, пряма кінематика забезпечує однозначне відображення простору узагальнених координат у простір декартових координат. Вона є обчислювально простою та стабільною, що дозволяє використовувати її як основу для перевірки правильності роботи алгоритмів зворотної кінематики.

Постановка зворотної задачі кінематики. На відміну від прямої кінематики, задача зворотної кінематики полягає у визначенні таких значень кутів $\theta_1, \theta_2, \theta_3$, при яких кінцевий ефектор маніпулятора досягає заданої точки (x, y) у робочому просторі. Формально це означає необхідність розв'язати систему нелінійних рівнянь:

$$\begin{aligned} x_{target} &= L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3), \\ y_{target} &= L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3). \end{aligned} \quad (7)$$

Система рівнянь (7) є нелінійною відносно змінних $\theta_1, \theta_2, \theta_3$, що унеможливує отримання універсального аналітичного розв'язку для довільної конфігурації. Крім того, для однієї і тієї ж точки можуть існувати декілька різних рішень, що відповідають різним конфігураціям маніпулятора.

З геометричної точки зору задача зворотної кінематики може бути інтерпретована як задача знаходження такого положення ланок, при якому їх сумарний вектор збігається із заданим вектором до цілі. Однак через наявність кількох ступенів свободи задача стає надлишковою, що створює додаткові труднощі при її розв'язанні.

Важливим аспектом є обмеження області досяжності. Маніпулятор може досягати лише тих точок, що знаходяться у межах кола радіусом:

$$R = L_1 + L_2 + L_3. \quad (8)$$

Якщо задана точка лежить поза цим колом, задача не має розв'язку. Крім того, навіть у межах досяжної області можуть існувати точки, що відповідають сингулярним конфігураціям, у яких матриця Якобі втрачає ранг, що призводить до нестабільності чисельних методів.

Ще однією особливістю є залежність розв'язку від початкового наближення. Ітераційні методи, що використовуються для розв'язання задачі, можуть сходитися до різних локальних мінімумів залежно від початкових значень кутів. Це означає, що одна і та ж задача може мати різні результати при різних початкових умовах.

Таким чином, зворотна задача кінематики є суттєво складнішою за пряму і вимагає застосування спеціалізованих чисельних методів. Її коректна постановка включає врахування обмежень, перевірку досяжності цілі та забезпечення стабільності алгоритму.

Матричні перетворення та Якобіан. Для ефективного розв'язання зворотної задачі кінематики необхідно встановити зв'язок між змінами кутів суглобів і змінами положення виконавчого органу. Цей зв'язок описується за допомогою матриці Якобі, яка є ключовим елементом чисельних методів розв'язання.

Матриця Якобі визначається як матриця частинних похідних координат виконавчого органу за узагальненими координатами:

$$J = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} \end{bmatrix}. \quad (9)$$

Обчислимо ці похідні для трьох ланок маніпулятора:

$$\begin{aligned} \frac{\partial x}{\partial \theta_1} &= -L_1 \sin(\varphi_1) - L_2 \sin(\varphi_2) - L_3 \sin(\varphi_3), \\ \frac{\partial x}{\partial \theta_2} &= -L_2 \sin(\varphi_2) - L_3 \sin(\varphi_3), \\ \frac{\partial x}{\partial \theta_3} &= -L_3 \sin(\varphi_3), \\ \frac{\partial y}{\partial \theta_1} &= L_1 \cos(\varphi_1) + L_2 \cos(\varphi_2) + L_3 \cos(\varphi_3), \\ \frac{\partial y}{\partial \theta_2} &= L_2 \cos(\varphi_2) + L_3 \cos(\varphi_3), \\ \frac{\partial y}{\partial \theta_3} &= L_3 \cos(\varphi_3). \end{aligned} \quad (10)$$

Матриця Якобі описує локальну лінійну апроксимацію залежності між змінами кутів і змінами координат:

$$\Delta x \approx J \cdot \Delta \theta, \quad (11)$$

де $\Delta x = [\Delta x, \Delta y]^T$, $\Delta \theta = [\Delta \theta_1, \Delta \theta_2, \Delta \theta_3]^T$.

Вираз (11) дозволяє перейти від задачі у просторі координат до задачі у просторі кутів. У випадку зворотної кінематики необхідно знайти таке $\Delta \theta$, яке мінімізує помилку між поточним і цільовим положенням. Оскільки матриця Якобі є прямокутною (2×3), задача розв'язується у сенсі найменших квадратів. Це означає, що шукається такий вектор змін кутів, який мінімізує норму помилки:

$$\|J \cdot \Delta \theta - \Delta x\|^2 \rightarrow \min. \quad (12)$$

Розв'язання даної задачі приводить до нормальних рівнянь виду:

$$J^T J \cdot \Delta \theta = J^T \cdot \Delta x. \quad (13)$$

Вираз (13) використовується у методі Левенберга – Марквардта з додатковим демпфуванням, що забезпечує стійкість розв'язку.

Таким чином, матриця Якобі пов'язує геометричну модель маніпулятора з чисельними методами оптимізації. Вона дозволяє перейти від нелінійної задачі до послідовності лінійних наближень, що робить можливим ефективне обчислення зворотної кінематики.

Метод Левенберга – Марквардта. Задача зворотної кінематики, як було зазначено раніше, зводиться до розв'язання нелінійної системи рівнянь відносно кутів суглобів маніпулятора. У загальному випадку така система не має аналітичного розв'язку або є надто складною для практичного використання. Саме тому у робототехніці широко застосовуються чисельні ітераційні методи, які дозволяють поступово наближатися до шуканого розв'язку. Одним із найбільш ефективних методів для таких задач є метод Левенберга – Марквардта (Levenberg – Marquardt, LM), який поєднує у собі властивості методу градієнтного спуску та методу Гауса – Ньютона. Його основна ідея полягає у мінімізації функції помилки між поточним положенням кінцевого ефектора та заданою цільовою точкою.

Формулювання задачі оптимізації. Нехай задано цільову точку:

$$x_{target} = \begin{bmatrix} x_{target} \\ y_{target} \end{bmatrix}, \quad (14)$$

та поточне положення виконавчого органу:

$$x(\theta) = \begin{bmatrix} x(\theta_1, \theta_2, \theta_3) \\ y(\theta_1, \theta_2, \theta_3) \end{bmatrix}. \quad (15)$$

Тоді вектор помилки визначається як:

$$e = x_{target} - x(\theta). \quad (16)$$

Задача зводиться до мінімізації квадратичної норми цієї помилки:

$$E(\theta) = \|e\|^2 = e_x^2 + e_y^2. \quad (17)$$

Отже, зворотна кінематика інтерпретується як задача оптимізації, у якій необхідно знайти такі значення θ , що мінімізують функцію E .

Лінеаризація задачі та використання Якобіана. Оскільки функція $x(\theta)$ є нелінійною, для її розв'язання використовується лінеаризація у околі поточного наближення:

$$x(\theta + \Delta \theta) \approx x(\theta) + J \cdot \Delta \theta, \quad (18)$$

де J – матриця Якобі (9).

Підставляючи це у вираз для помилки, отримуємо:

$$e \approx e_0 - J \cdot \Delta \theta, \quad (19)$$

де e_0 – поточна помилка обчислення.

Задача зводиться до мінімізації наступного виразу:

$$\|e_0 - J \cdot \Delta \theta\|^2 \rightarrow \min. \quad (20)$$

Задача (20) може бути розв'язана методом найменших квадратів через нормовану форму рівняння:

$$J^T J \cdot \Delta\theta = J^T \cdot e_0. \quad (21)$$

Метод Левенберга – Марквардта. На додачу до метода Гауса–Ньютона, що безпосередньо використовує нормальні рівняння для обчислення кроку наближення, вводиться демпфуючий параметр λ , який стабілізує розв'язок:

$$(J^T J + \lambda I) \cdot \Delta\theta = J^T \cdot e, \quad (22)$$

де I – одинична матриця.

Параметр λ регулює компроміс між швидкістю збіжності та стабільністю роботи алгоритму. При великих значеннях λ метод поводить як метод градієнтного спуску, при малих значеннях λ – як метод Гауса–Ньютона. Алгоритм методу Левенберга – Марквардта реалізується у вигляді ітераційного процесу за наступними кроками:

1. Задати початкові значення кутів θ_0 ;
2. Обчислити поточне положення x та помилку e ;
3. Побудувати матрицю Якобі J ;
4. Сформувати розширену матрицю $A = J^T J + \lambda I$;
5. Обчислити вектор: $b = J^T e$;
6. Розв'язати систему: $A \cdot \Delta\theta = b$;
7. Оновити кути: $\theta_{new} = \theta + \Delta\theta$;
8. Оцінити нову помилку: якщо вона зменшилась – прийняти крок і зменшити λ , якщо збільшилась –

відхилити крок і збільшити λ ;

9. Повторювати процес до досягнення критерію зупинки – помилки розузгодження.

Для забезпечення роботи алгоритму вводяться декілька умов зупинки: за помилкою $\|e\| < \varepsilon_{err}$, за кроком $\|\Delta\theta\| < \varepsilon_{step}$ за градієнтом збіжності $\|J^T e\| < \varepsilon_{grad}$. Наявність декількох критеріїв дозволяє уникнути як передчасної зупинки, так і зациклення алгоритму.

У контексті даної роботи метод Левенберга – Марквардта має ряд практичних особливостей. Кути обмежуються фізичними межами суглобів: $\theta_1 \in [20^\circ, 160^\circ]$, θ_2 та $\theta_3 \in [-100^\circ, 100^\circ]$. Початкове наближення береться з поточної конфігурації маніпулятора, що забезпечує плавність руху. Перед запуском алгоритму виконується перевірка умови досяжності:

$$\sqrt{x^2 + y^2} \leq L_1 + L_2 + L_3. \quad (23)$$

Кожну ітерацію використовується адаптивна зміна параметра λ , що підвищує стабільність у складних конфігураціях.

З навчальної точки зору метод Левенберга–Марквардта можна інтерпретувати як процес поступового «підтягування» маніпулятора до цільової точки. На кожній ітерації алгоритм визначає, у який бік і на яку величину необхідно змінити кути, щоб зменшити помилку. При цьому матриця Якобі визначає напрямок руху, параметр λ контролює «обережність» кроку, система нормальних рівнянь визначає оптимальний баланс між усіма суглобами. Алгоритм, що його реалізує, не знаходить гарантовано точний розв'язок одразу, а поступово наближається до нього, забезпечуючи стабільність пошуку рішення навіть при складних обмеженнях.

Програмна реалізація симулятора виконана як навчально-дослідницька система, призначена для наочного представлення математичних обчислень розв'язання задач кінематики, графічної візуалізації кінематичної структури роботизованого маніпулятора та інтерфейсу користувача для взаємодії з параметрами моделі.

Для реалізації симулятора було обрано середовище Visual Studio та технологію Windows Forms із використанням мови C++/CLI. Windows Forms дозволяє створювати прикладні програми з графічним інтерфейсом, що має множину готових об'єктів та візуальних компонентів форми, таких як текстові поля, елементи керування та графічну область побудови, необхідні для відтворення навчального STEM контенту. Також C++/CLI дає змогу поєднати строго типізовану мову з зручним конструюванням графічного інтерфейсу у Visual Studio, що є доцільним для задач, у яких математичні розрахунки тісно інтегруються з GUI. Реалізація об'єктно-орієнтовану архітектуру програми та низькорівневих математичних структур без необхідності переходу до сторонніх бібліотек забезпечує прозорість усіх етапів створення проекту: від опису векторів і матриць до побудови графічного зображення ланок маніпулятора та реалізації ітераційного алгоритму зворотної кінематики.

Архітектурно симулятор побудований за модульним принципом. Математичні обчислення, геометричне представлення ланок та елементи інтерфейсу користувача розділені між окремими сутностями, кожна з яких виконує власну функцію. Такий підхід спрощує супровід програми, покращує читабельність коду та забезпечує можливість подальшого розширення без радикальної зміни всієї системи.

Математичне ядро програми складається з декількох базових класів. Клас Vector4 використовується для представлення точок у однорідних координатах. Його застосування дозволяє уніфікувати роботу з просторовими перетвореннями та підготувати модель до переходу від двовимірного до тривимірного опису. Клас Matrix4x4 реалізує базові матричні операції, необхідні для побудови матриць повороту, перенесення та їх послідовного множення для реалізації прямої задачі кінематики на основі композиції

перетворень. Клас Link відповідає за представлення окремої ланки: він містить локальну геометрію об'єкта та матрицю трансформації, що визначає поточне положення ланки в глобальній системі координат.

Окреме місце у структурі програми займає головна форма Form1, яка виконує роль центру координації між інтерфейсом та математичним ядром. У межах цієї форми розміщені елементи введення параметрів, панель графічного відображення, вкладка математичного файлу звітності, інструменти роботи з траєкторією та безпосередньо виклики алгоритмів прямої і зворотної кінематики. Таким чином, форма не лише представляє користувацький інтерфейс, але й реалізує прикладну логіку програми.

Логіка обчислень організована таким чином, що програма підтримує два основних режими роботи. У режимі прямої задачі кінематики користувач задає кути повороту суглобів і довжини ланок, після чого програма обчислює положення кінцевого виконавчого органу та візуалізує конфігурацію маніпулятора. У режимі зворотної задачі кінематики користувач задає бажані координати виконавчого органу, а програма запускає ітераційний алгоритм, який підбирає кути суглобів так, щоб досягти цільового положення з прийнятною точністю. Перемикання між цими режимами реалізовано через окремих елемент керування, що дозволяє використовувати одну й ту саму графічну модель у двох принципово різних сценаріях.

Інтерфейс користувача виконує не лише прикладну, але й навчальну функцію. У головному вікні розміщено графічну панель, на якій відображається поточне положення маніпулятора. Праворуч від панелі зосереджені елементи введення: текстові поля для кутів суглобів, текстові поля для довжин ланок, кнопка застосування параметрів, перемикач режиму FK/IK, поля координат виконавчого органу, а також повзунки для швидкого ручного керування кутами. Адаптація інтерфейсу до режиму зворотної задачі кінематики полягає у гнучкому використанні полів для та відображення результатів розрахунків чи управління в залежності від задачі кінематики до розв'язання.

Симулятор містить окрему вкладку «Математика», у якій відображаються формули, матриці перетворень, числові значення кутів, координати проміжних суглобів та виконавчого органу. Дана вкладка реалізує дидактичну функцію, оскільки дозволяє простежити зв'язок між математичним апаратом і тим, що користувач спостерігає на екрані симуляції. Інтерфейс програми реалізує можливість інтерактивного навчання, у якому користувач бачить не лише результат, але й логіку його отримання.

Якщо пряма задача кінематики є алгоритмічно детермінованою, то у випадку зворотної задачі кінематики, після введення цільових координат виконується перевірка досяжності цільової точки виконавчого органу. Якщо відстань до цільової точки перевищує сумарну довжину ланок, програма повідомляє користувача про неможливість досягнення заданого положення. Якщо ж точка знаходиться в межах досяжності, запускається алгоритм Левенберга – Марквардта, який використовує поточні кути як початкове наближення. Інтерфейс містить налаштування обмеження кінематичних параметри роботизованого маніпулятора для перевірки та пошуку фізично досяжних конфігурацій та зберігання їх множини для подальшого планування руху, інтерполюючи узагальнені координати.

Результати апробації. Розглянемо на прикладі частини кінематичного ланцюга навчального робота «Wgaccio robot arm» розв'язання зворотної задачі кінематики, її збіжність та поведінку системи у різних конфігураціях. Прийmemo між шарнірні відстані наступними:

$$L_1 = 20, L_2 = 20, L_3 = 20.$$

Початкове положення кінцевого ефектора, визначене у режимі прямої кінематики, становить:

$$X_{start} = 30,79, Y_{start} = 47,02.$$

Цільове положення, задане користувачем у режимі зворотної кінематики:

$$X_{target} = -10, Y_{target} = 35.$$

Перед запуском алгоритму, в програмі було виконано перевірку досяжності цільової точки:

$$R = \sqrt{(-10)^2 + 35^2} \approx 36,40.$$

Максимальний радіус досяжності складає:

$$R_{max} = L_1 + L_2 + L_3 = 60.$$

Оскільки $R < R_{max}$, точка є досяжною, що дозволяє застосувати алгоритм зворотної кінематики.

На першій ітерації алгоритм використовує поточні значення кутів як початкове наближення. Обчислюється положення виконавчого органу та визначається початкова помилка:

$$e_x = X_{target} - X_{current},$$

$$e_y = Y_{target} - Y_{current}.$$

Підставляючи значення, отримуємо:

$$e_x = -10 - 30,79 = -40,79,$$

$$e_y = 35 - 47,02 = -12,02.$$

Норма помилки складає:

$$\|e\| = \sqrt{(-40,79)^2 + (-12,02)^2} \approx 42,53.$$

Отримане значення помилки є значним, що свідчить про віддаленість початкового стану від цільового положення. Далі формується матриця Якобі для поточних значень кутів, що визначає чутливість зміни координат виконавчого органу до змін кутів суглобів. На основі матриці (9) обчислюється система

нормальних рівнянь (22). Розв'язуючи цю систему, отримуємо вектор корекції кутів $\Delta\theta$ та їх оновлення θ_{new} . На кожній наступній ітерації процес повторюється: обчислюється нове положення, визначається помилка, формується Якоб'ян і вирішується система рівнянь. Поступово помилка зменшується, а конфігурація маніпулятора наближається до цільової. У ході експерименту було встановлено, що алгоритм демонструє стабільну збіжність для заданої точки. Зменшення помилки відбувається монотонно або з незначними коливаннями, що пов'язано з адаптивною зміною параметра демпфування λ . Кінцевий результат характеризується малою нормою помилки $\|e\| < 0,1$ що відповідає заданому порогу точності.

Демонстрація вкладки «Симулятор» розв'язання прямої задачі кінематики за наведеним прикладом зображена на рисунку 1.

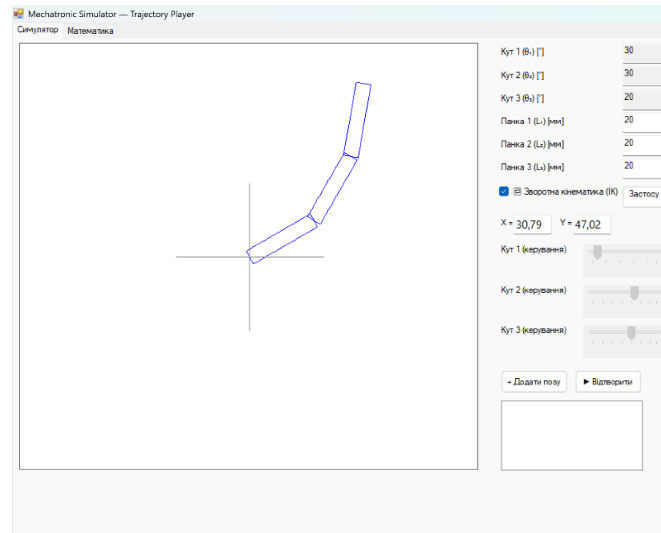


Рис. 1. Розв'язання прямої задачі кінематики

Демонстрація вкладки «Симулятор» розв'язання зворотної задачі кінематики за наведеним прикладом зображена на рисунку 2.

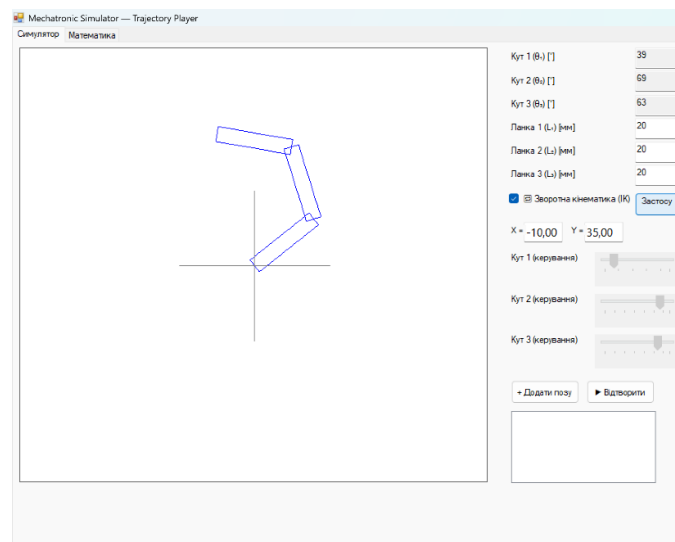


Рис. 2. Розв'язання зворотної задачі кінематики

Як видно, в результаті розв'язання оберненої задачі, вектор узагальнених координат набув визначення $\Delta\theta = [39,69,63]^T$.

Для пошуку інших кінцевих конфігурацій користувач експериментує з початковими конфігураціями, зберігаючи результати в програмі командою «Додати позу».

Демонстрація вкладки «Математика», що демонструє усі ключові розрахунки розв'язання задач, зображена на рисунку 3.

```

Mechatronic Simulator - Trajectory Player
Симулятор  Математика

=== АНАЛІЗ КІНЕМАТИКИ ===

Кутти:  $\theta_1=38,5^\circ$ ,  $\theta_2=68,8^\circ$ ,  $\theta_3=62,8^\circ$ 
Довжини:  $L_1=20,0$  мм,  $L_2=20,0$  мм,  $L_3=20,0$  мм

--- Символічна модель ---

--- Символічні формули прямої кінематики ---

 $x = L_1 \cdot \cos(\theta_1) + L_2 \cdot \cos(\theta_1 + \theta_2) + L_3 \cdot \cos(\theta_1 + \theta_2 + \theta_3)$ 
 $y = L_1 \cdot \sin(\theta_1) + L_2 \cdot \sin(\theta_1 + \theta_2) + L_3 \cdot \sin(\theta_1 + \theta_2 + \theta_3)$ 

--- Складені кутти ---
 $\phi_1 = \theta_1$ 
 $\phi_2 = \theta_1 + \theta_2$ 
 $\phi_3 = \theta_1 + \theta_2 + \theta_3$ 

--- Координати проміжних ланок ---
 $X_1 = L_1 \cdot \cos(\theta_1)$ 
 $Y_1 = L_1 \cdot \sin(\theta_1)$ 
 $X_2 = L_1 \cdot \cos(\theta_1) + L_2 \cdot \cos(\theta_1 + \theta_2)$ 
 $Y_2 = L_1 \cdot \sin(\theta_1) + L_2 \cdot \sin(\theta_1 + \theta_2)$ 
 $X_3 = L_1 \cdot \cos(\theta_1) + L_2 \cdot \cos(\theta_1 + \theta_2) + L_3 \cdot \cos(\theta_1 + \theta_2 + \theta_3)$ 
 $Y_3 = L_1 \cdot \sin(\theta_1) + L_2 \cdot \sin(\theta_1 + \theta_2) + L_3 \cdot \sin(\theta_1 + \theta_2 + \theta_3)$ 

--- Матриці обертання та перенесення ---
 $R_{i>(n_i)} =$ 

```

Рис. 3. Математичний аналіз розв'язання задач кінематики (фрагмент)

У процесі апробації було також досліджено поведінку алгоритму у складних положеннях маніпулятора. До таких положень відносяться конфігурації, близькі до повного розгинання ланок, положення, у яких маніпулятор змушений переходити через різкі зміни напрямку руху та ситуації, коли початкове наближення знаходиться далеко від цільової точки. При цьому алгоритм може демонструвати наступні особливості: сповільнення збіжності при наближенні до сингулярних конфігурацій, де матриця Якобі стає погано обумовленою, що ускладнює обчислення корекції кутів; збільшення кількості ітерацій при досягненні заданої точності; можливість локальної нестабільності у деяких випадках, коли алгоритм може тимчасово збільшувати помилку, після чого знову переходить до збіжності завдяки зміні параметра λ . Вплив початкового наближення, якщо початкове положення сильно відрізняється від цільового, коли алгоритм може обрати іншу конфігурацію маніпулятора або потребувати більше часу для збіжності.

Незважаючи на ці фактори алгоритм демонструє стабільну роботу та здатність знаходити коректні розв'язки у межах досяжної області. Таким чином, результати моделювання демонструють, що реалізований алгоритм зворотної кінематики на основі методу Левенберга – Марквардта є ефективним і придатним для використання у навчальних та прикладних задачах. Отримані результати підтверджують коректність математичної моделі, та правильність її програмної реалізації.

Висновки та перспективи подальших досліджень. Розроблений програмний симулятор розв'язання задач кінематики є дидактичним інструментом навчання, що дозволяє наочно представити математичний зміст вирішення задач та спирається на чисельний метод розв'язання практично важливої, зворотної задачі кінематики. Візуалізація даних при побудові траєкторії руху виконавчого органу дозволяє оцінити доступні конфігурації та вибрати із множини такі, які відповідає планованому експерименту для подальшої реалізації. Перспективою подальших досліджень є розробка повноцінної, тривимірної моделі «Braccio robot arm» як цифрового двійника з використанням рушію Unity, та створення комп'ютерного управління фізичним макетом робота в режимі реального часу.

References:

1. Siciliano, B., Sciavicco, L., Villani, L. and Oriolo, G. (2009), *Robotics: Modelling, Planning and Control*, Springer, 632 p., [Online], available at: https://people.disim.univaq.it/~costanzo.manes/EDU_stuff/Robotics_Modelling,%20Planning%20and%20Control_Sciavicco_extract.pdf
2. Spong, M.W., Hutchinson, S. and Vidyasagar, M. (2020), *Robot Modeling and Control*, 2nd ed., John Wiley & Sons, 538 p.
3. Lu, J., Zou, T. and Jiang, X. (2022), «A Neural Network Based Approach to Inverse Kinematics Problem for General Six-Axis Robots», *Sensors*, Vol. 22, No. 22, [Online], available at: <https://www.mdpi.com/1424-8220/22/22/8909>
4. Trullo, A. et al. (2023), «A Systematic Review of Inverse Kinematics Methods for Fixed-Base Serial Manipulators: Analytical, Numerical, and Machine Learning Methods», *International Journal of Robotics and Control Systems*, Vol. 3, No. 4, [Online], available at: https://www.researchgate.net/publication/394449259_A_Systematic_Review_of_Inverse_Kinematics_Methods_for_Fixed-Base_Serial_Manipulators_Analytical_Numerical_and_Machine_Learning_Methods
5. Xie, S., Sun, L., Wang, Z. and Chen, G. (2022), «A speedup method for solving the inverse kinematics problem of robotic manipulators», *International Journal of Advanced Robotic Systems*, Vol. 19, No. 3, [Online], available at: https://www.researchgate.net/publication/361420818_A_speedup_method_for_solving_the_inverse_kinematics_problem_of_robotic_manipulators

6. Sugihara, T. (2011), «Solvability-Unconcerned Inverse Kinematics by the Levenberg–Marquardt Method», *IEEE Transactions on Robotics*, Vol. 27, No. 5, [Online], available at: https://www.researchgate.net/publication/224240315_Solvability-Unconcerned_Inverse_Kinematics_by_the_Levenberg-Marquardt_Method
7. Mazumder, A. et al. (2023), «Towards Next Generation Digital Twin in Robotics: Trends, Scopes, Challenges, and Future», *Heliyon*, Vol. 9, [Online], available at: https://www.researchgate.net/publication/367509675_Towards_Next_Generation_Digital_Twin_in_Robotics_Trends_Scopes_Challenges_and_Future
8. Douthwaite, J.A. et al. (2021), «A Modular Digital Twinning Framework for Safety Assurance of Collaborative Robotics», *Frontiers in Robotics and AI*, Vol. 8, [Online], available at: <https://eprints.whiterose.ac.uk/id/eprint/182317/1/frobt-08-758099.pdf>
9. Loyola, O. et al. (2024), «Integration of Cyber-Physical System and Digital Twin for Controlling a Robotic Manipulator: An Industry 4.0 Approach», *Advances in Computing. CCC 2023*, Springer, [Online], available at: https://www.researchgate.net/publication/375634281_Integration_of_Cyber_Physical_System_and_Digital_Twin_for_Controlling_a_Robotic_Manipulator_An_Industry_40_Approach
10. Singh, M., Murry, N. and Devine, D. (2024), «Unity and ROS as a Digital and Communication Layer for Digital Twin Application: Case Study of Robotic Arm in a Smart Manufacturing Cell», *Sensors*, Vol. 24, No. 17, [Online], available at: https://www.researchgate.net/publication/383685217_Unity_and_ROS_as_a_Digital_and_Communication_Layer_for_Digital_Twin_Application_Case_Study_of_Robotic_Arm_in_a_Smart_Manufacturing_Cell
11. Liu, X. et al. (2024), «Review on Motion Planning of Robotic Manipulator in Dynamic Environments», *Journal of Sensors*, [Online], available at: https://www.researchgate.net/publication/386043108_Review_on_Motion_Planning_of_Robotic_Manipulator_in_Dynamic_Environments

Богдановський Мартін Віталійович – старший викладач кафедри робототехніки, електроенергетики та автоматизації Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0003-2870-4248>.

Наукові інтереси:

– робототехніка; мехатроніка; моделювання систем управління.

E-mail: aikt_bmv@ztu.edu.ua.

Підтиченко Олександр Владиславович – кандидат технічних наук, доцент кафедри робототехніки, електроенергетики та автоматизації імені проф. Б.Б. Самотокіна Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0003-1748-8068>.

Наукові інтереси:

– мікропроцесорні системи керування; програмування ПЛК; гнучкі виробничі комірки.

E-mail: aikt_pov@ztu.edu.ua.

Горбик Денис Петрович – здобувач Державного університету «Житомирська політехніка».

Наукові інтереси:

– комп'ютерне моделювання та програмування; розробка керуючих програм ЧПК.

E-mail: at39_gdp@students.ztu.edu.ua.

Bogdanovskyi M.V., Pidtychenko O.V., Gorbik D.P.

Development of a software simulator for the study of robotic manipulator kinematics tasks and control

Kinematics is an integral part of research in the design of industrial robot structures and the development of control systems. The subject of study in this case are approaches and mathematical methods for describing the spatial position of the elements of the kinematic chain, the ultimate goal of which is to determine the generalized coordinates according to the position and orientation of the end effector. The paper presents the development of a software simulator for solving kinematics problems on the example of the educational robot «Braccio robot arm», which clearly demonstrates the mathematical apparatus. For example, a three link kinematic chain section that forms the ambiguity of solving the inverse problem of kinematics, the use of the numerical method of fading least squares (Liewenberg – Marquardt) to find all available configurations necessary for effective planning of the robot's movement is demonstrated. The implementation of the software part in the Visual Studio environment using Windows Forms technology in the C++/CLI language is demonstrated. The mathematical apparatus of a direct kinematic problem based on homogeneous transformation matrices (4×4), the application of the Lievenberg – Marquardt method with the analytical calculation of the Jacobi matrix, the structure of the software, the user interface and the results of experimental verification are described. In the course of the work, the practical stability of the results and the practical significance for further tasks of developing a control system for a robotic manipulator have been proven. The prospects for transferring the mathematical core of the simulator to the Unity 3D environment within the framework of the concept of a digital twin of a physical manipulator for reproducing program control are outlined.

Keywords: industrial robots; simulator; stem education; kinematics; design; Jacobian; modeling.

Стаття надійшла до редакції 30.12.2025.