

УДК 004.896

DOI 10.52575/2687-0959-2023-55-1-70-83

MSC 93C85

оригинальное исследование

РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ ДЛЯ МОДЕЛИРОВАНИЯ КИНЕМАТИКИ И ДИНАМИКИ МАНИПУЛЯТОРА

Ту Раин 

(Статья представлена членом редакционной коллегии С. М. Ситником)

Белгородский государственный национальный исследовательский университет,

Белгород, 308015, Россия

E-mail: thurein.48@gmail.com

Аннотация. В статье предлагается разработка программного модуля для моделирования кинематики и динамики манипулятора с пятью степенями подвижности. Для решения прямой задачи кинематики манипулятора использован метод Денавита – Хартенберга. Для решения обратной задачи кинематики и динамики манипулятора использованы аналитические методы – метод Левенберга – Марквардта, метод Ньютона – Эйлера и метод мягких вычислений – адаптивная нейро-нечеткая система вывода. Разработан программный модуль для моделирования кинематики и динамики манипулятора с использованием программного комплекса системы автоматизированного проектирования SolidWorks и программы MatLab. Полученный программный модуль позволяет выполнять моделирование кинематики и динамики манипулятора на основе описываемых методов, визуализацию результатов моделирования, формирование траектории для целевого положения и ориентации рабочего органа манипулятора, имитационное моделирование движения манипулятора по заданной траектории.

Ключевые слова: компьютерное моделирование, моделирование кинематики, метод Левенберга – Марквардта, моделирование динамики, метод Ньютона – Эйлера, манипулятор

Для цитирования: Ту Раин. 2023. Разработка программного модуля для моделирования кинематики и динамики манипулятора. Прикладная математика & Физика, 55(1): 70–83. DOI 10.52575/2687-0959-2023-55-1-70-83

DEVELOPMENT OF PROGRAM MODULE FOR MODELING KINEMATICS AND DYNAMICS OF MANIPULATOR

Thu Rain 

(Article submitted by a member of the editorial board S. M. Sitnik)

Belgorod National Research University,

Belgorod, 308015, Russia

E-mail: thurein.48@gmail.com

Abstract. The article proposes the development of a software module for modeling the kinematics and dynamics of a manipulator with five degrees of freedom. To solve the forward kinematics problem of the manipulator, the Denavit-Hartenberg method was used. To solve the inverse kinematics problem and inverse dynamics problem of the manipulator, analytical methods – the Levenberg – Marquardt method, the Newton – Euler method, and the soft computing method – adaptive neuro-fuzzy inference system were used. And a software module was developed for modeling the kinematics and dynamics of the manipulator using computer-aided design application SolidWorks and the MatLab program. The developed software module is able to simulate the kinematics and dynamics of the manipulator based on the described methods, visualize the simulation results, generate a trajectory for the target position and orientation of the end-effector of the manipulator, simulate the movement of the manipulator along a given trajectory.

Keywords: Computer Modeling, Kinematics Modeling, Levenberg – Marquardt Method, Dynamics Modeling, Newton – Euler Method, Manipulator

For citation: Thu Rain. 2023. Development of program module for modeling kinematics and dynamics of manipulator. Applied Mathematics & Physics, 55(1): 70–83. (in Russian) DOI 10.52575/2687-0959-2023-55-1-70-83

1. Введение. Разработки программного обеспечения для кинематического и динамического моделирования манипуляторов требуются для исследования движения рабочего органа, анализа конструкций, разработки системы автоматического управления и алгоритмов функционирования манипуляторов [16, 9, 7, 4]. Одним из основных преимуществ программных симуляций является то, что они могут

предоставить пользователям практическую обратную связь при разработке реальных систем. Это позволяет разработчику определить правильность и эффективность проекта еще до того, как система будет построена.

Существует несколько алгоритмов программного обеспечения для моделирования манипуляторов. Во многих работах для кинематического анализа различных манипуляторов использовалась программа RoboAnalyzer [11, 15, 7, 14]. Программное обеспечение для автоматизированного проектирования «САПР», такое как SolidWorks, CATIA и Autodesk, также используется для моделирования манипуляторов [3, 12]. В статье [13] рассмотрены методы моделирования манипуляторов с использованием различных программ, таких как GraspIt!, OpenGrasp, MATLAB/Simulink, SynGrasp, V-Realm Builder и ADAMS.

В статьях [6, 8] предлагаются методологии создания веб-графического пользовательского интерфейса для моделирования движения и кинематического исследования промышленных манипуляторов. В них описываются простые и понятные методы моделирования с использованием Web Graphics Library «WebGL» для создания любой трехмерной виртуальной модели манипуляторов на холсте HTML для имитации движения и различных видов анализа. Моделирование кинематического управления манипулятором на основе адаптивной нейро-нечеткой системы вывода «АНСВ» было представлено в работе [1]. В статье [17] разработана динамическая модель манипулятора с 6 степенями свободы с помощью уравнений Ньютона – Эйлера и проанализировано его динамическое поведение. Динамическое моделирование было выполнено с помощью программного обеспечения для динамического моделирования ADAMS. В работе [10] представлен метод анализа сил и моментов манипулятора с тремя степенями свободы, основанный на имитационной модели манипулятора в среде Matlab/Simulink, и проведено сравнение полученных результатов с математическим расчетом.

В данной работе приводится проектирование и реализация программного модуля компьютерного и имитационного моделирования манипулятора в рабочей зоне, реализующие разработанные алгоритмы и методы, позволит применять их в разработке манипуляторов, системах управления манипуляторами. Данный программный модуль компьютерного и имитационного моделирования также позволяет проводить эксперименты, необходимые для исследования предложенных в работе методов и алгоритмов. Разработанные программные модули предоставляют следующие функциональности: визуализация 3D-моделирования; взаимодействие с пользователем через графический интерфейс; расчет обратной задачи кинематики «ОЗК» манипулятора на основе метода Левенберга – Марквардта и АНСВ; расчет обратной задачи динамики «ОЗД» манипулятора на основе метода Ньютона – Эйлера и АНСВ; формирование траектории для целевого положения и ориентации рабочего органа манипулятора; имитационное моделирование движения манипулятора по заданной траектории.

2. Методология. Для программирования математического моделирования манипулятора была выбрана среда «MATLAB» Для реализации 3D-моделирования выбрана САПР-программа «SolidWorks» с целью достижения фактического размера, степени свободы и других факторов, касающихся анимации и моделирования манипулятора, будет разложен манипулятор с пятью степенями свободы, как показано на рисунке 1.

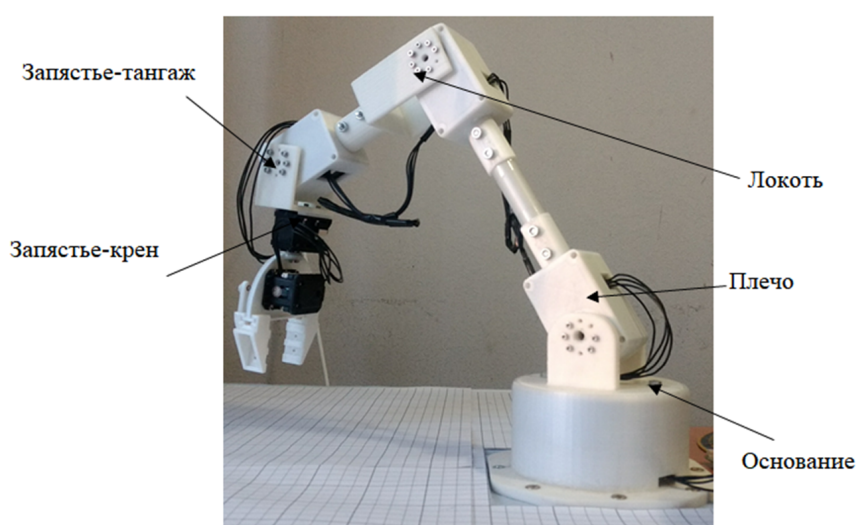


Рис. 1. Манипулятор с пятью степенями свободы

Fig. 1. Manipulator with five degrees of freedom

Процесс 3D-моделирования манипулятора разделен на два основных вида деятельности, а именно графический дизайн и дизайн поведения. Сначала была построена виртуальная геометрическая модель, использующая реальные размеры экспериментального манипулятора.

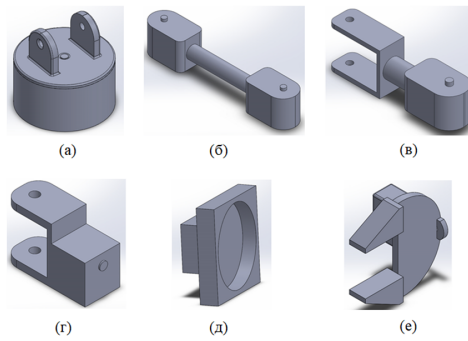


Рис. 2. 3D-детали манипулятора с пятью степенями свободы
Fig. 2. 3D parts of manipulator with five degrees of freedom

На рисунке 2 показаны разработанные 3D-детали звеньев манипулятора с пятью степенями свободы.

После разработки 3D-деталей отдельные детали были собраны в один файл «сборка». После ввода деталей в сборку были введены шарниры вращения манипулятора. Также при виртуальной сборке манипулятора, помимо обычных ограничений, 3D-детали помещаются в вертикальное положение. Это положение было принято для облегчения следующих шагов; собранное положение рассматривается как исходное, и все изменения углов стыков рассчитываются в соответствии с этим положением. Полученная 3D-модель, выбор системы координат и кинематические параметры манипулятора представлены на рисунке 3.

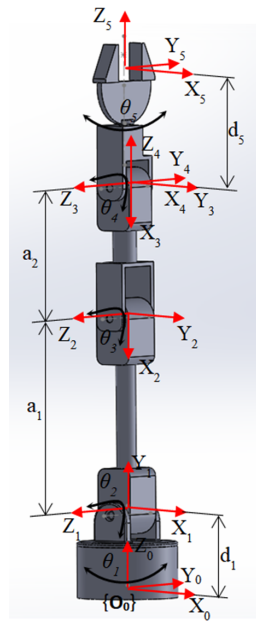


Рис. 3. 3D-модель сборки манипулятора с пятью степенями свободы
Fig. 3. 3D assembly model of manipulator with five degrees of freedom

Значения соответствующих параметров указаны в таблице 1 и геометрических параметров в таблице 2.

Таблица 1. Параметры Денавита – Хартенберга (Д-Х) манипулятора с пятью степенями свободы

Table 1. Denavit-Hartenberg parameters of manipulator with five degrees of freedom

Звено	a_{i-1} (м)	α_{i-1} (рад)	d (м)	θ (рад)
1	0	$\pi/2$	d_1	θ_1
2	0.246	0	0	θ_2
3	0.163	0	0	θ_3
4	0	$-\pi/2$	0	θ_4
5	0	0	0.165	θ_5

Затем модель интерпретируется в формат URDF (Унифицированный формат описания робототехники) для проведения моделирования и анализа. На рисунке 4 показан процесс перевода САПР, который, по

Таблица 2. Геометрические параметры манипулятора
Table 2. Geometric parameters of the manipulator

Звено	Высота (м)	Радиус (м)	Масса (кг)
1	0.071	0.06	0.102
2	0.295	0.035	0.569
3	0.205	0.035	0.310
4	0.091	0.035	0.123
5	0.08	0.05	0.142

сути, состоит из двух этапов. Первоначально сборка САПР, разработанная в SolidWorks, экспортируется в формат URDF. Затем файл URDF импортируется в MATLAB для создания имитационной модели, которая будет открыта в среде моделирования.

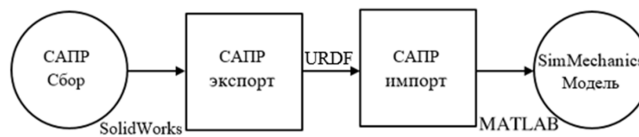


Рис. 4. Процесс перевода САПР
Fig. 4. The process of CAD assembly file translation

3. Разработка компонентов программного интерфейса «GUI». Для упрощения экспериментальной работы и получения лучшего представления о поведении системы в среде MATLAB был разработан графический пользовательский интерфейс (GUI) с использованием инструмента Среды разработки графического пользовательского интерфейса MATLAB (GUIDE). Для выбранного набора параметров графический интерфейс выполняет моделирование, отображающее временные характеристики углов сочленения, крутящих моментов в сочленениях и положения рабочего органа манипулятора. На рисунке 5 изображено основное окно программы, в левой части которого находится группа меню, в центральной части находятся панели вводов и выводов и в правой части находится панель визуализации манипулятора. Используя переключатели в группе меню, пользователь выбирает необходимый тип задачи:

1. Прямая задача кинематики,
2. Обратная задача кинематики,
3. Траектория и динамика.

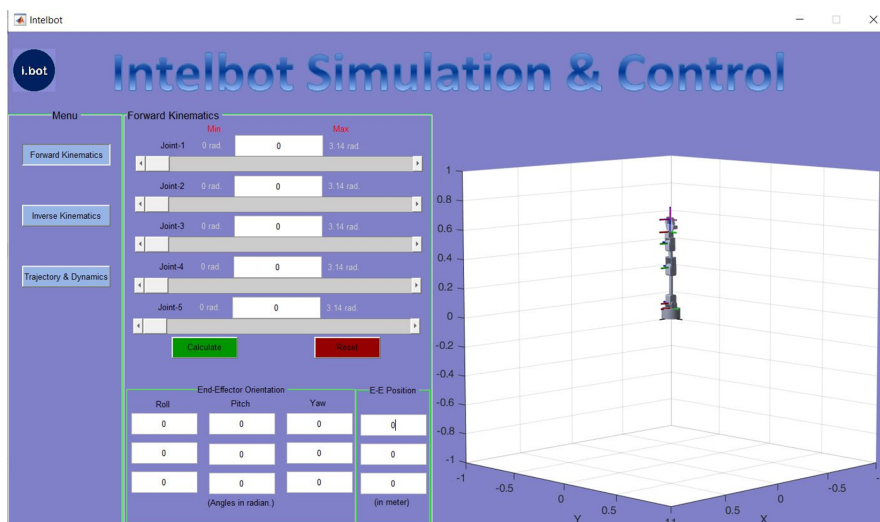


Рис. 5. Основное окно программы
Fig. 5. The main window of the program

4. Разработка программного компонента решения прямой задачи кинематики манипулятора. Определение положения и ориентации рабочего органа при заданных значений вектора обобщенных координат манипулятора называется прямой задачей кинематики. Прямую задачу кинематики манипулятора формулируют следующим образом: по заданному вектору обобщенных координат

$\theta = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)^T$ найти положение и ориентацию рабочего органа $s = f(\theta)$. Положение и ориентацию рабочего органа будем искать в форме матрицы однородного преобразования:

$$T_i = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Согласно правилам Денавита – Хартенберга, матрица однородного преобразования T_i , которая имеет вид

$$T_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\alpha_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\alpha_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

задает переход от системы координат (i -ого) звена к системе координат (i -ого) звена. Выполняя необходимые вычисления, получим ориентацию рабочего органа:

$$\begin{aligned} n_x &= s_1 s_5 - c_5 [c_4 (c_1 s_2 s_3 - c_1 c_2 c_3) + s_4 (c_1 c_2 s_3 + c_1 c_3 s_2)], \\ n_y &= -c_1 s_5 - c_5 [c_4 (s_1 s_2 s_3 - c_2 c_3 s_1) + s_4 (c_2 s_1 s_3 + c_3 s_1 s_2)], \\ n_z &= c_5 (c_4 s_2 s_3 + s_4 c_2 s_3), \\ o_x &= c_5 s_1 + [c_4 (c_1 s_2 s_3 - c_1 c_2 c_3) + s_4 (c_1 c_2 s_3 + c_1 c_3 s_2)], \\ o_y &= s_5 [c_4 (s_1 s_2 s_3 - c_2 c_3 s_1) + s_4 (c_2 s_1 s_3 + c_3 s_1 s_2)] - c_1 c_5, \\ o_z &= -s_5 [c_4 (c_2 s_3 + c_3 s_2) + s_4 (c_2 c_3 - s_2 s_3)], \\ a_x &= c_4 (c_1 c_2 s_3 + c_1 c_3 s_2) - s_4 (c_1 s_2 s_3 - c_1 c_2 c_3), \\ a_y &= c_4 (c_2 s_1 s_3 + c_3 s_1 s_2) - s_4 (s_1 s_2 s_3 - c_2 c_3 s_1), \\ a_z &= s_4 (c_2 s_3 + c_3 s_2) - c_4 (c_2 c_3 - s_2 s_3). \end{aligned}$$

А положение рабочего органа:

$$\begin{aligned} p_x &= d_5 [c_4 (c_1 c_2 s_3 + c_1 c_3 s_2) - s_4 (c_1 s_2 s_3 - c_1 c_2 c_3)] + a_2 c_1 c_2 + a_3 c_1 c_2 c_3 - a_3 c_1 s_2 s_3, \\ p_y &= d_5 [c_4 (c_2 s_1 s_3 + c_3 s_1 s_2) - s_4 (s_1 s_2 s_3 - c_2 c_3 s_1)] + a_2 c_2 s_1 + a_3 c_2 c_3 s_1 - a_3 s_1 s_2 s_3, \\ p_z &= d_1 + a_2 s_2 - d_5 [c_4 (c_2 c_3 - s_2 s_3) - s_4 (c_2 s_3 + c_3 s_2)] + a_3 c_2 s_3 + a_3 c_3 s_2. \end{aligned}$$

На рисунке 6 приведена блок-схема алгоритма программного компонента прямой задачи кинематики.

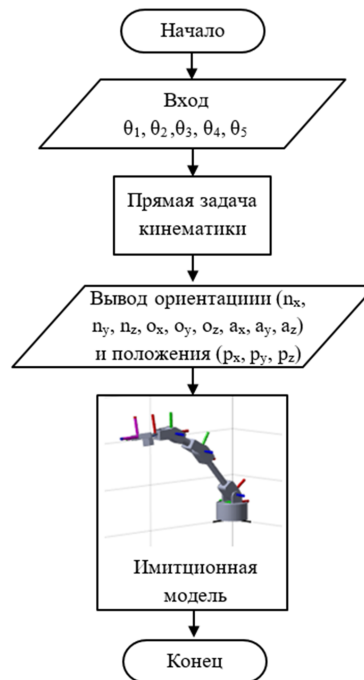


Рис. 6. Блок-схема алгоритма программы прямой задачи кинематики

Fig. 6. Block diagram of the program algorithm for the forward kinematics

5. Разработка программного компонента решения обратной задачи кинематики манипулятора. Метод Левенберга – Марквардта. Определение вектора обобщенных координат, который позволяет

манипулятору достичь желаемых положения и ориентаций рабочего органа, называется обратной задачей кинематики. В данной работе для численного решения обратной задачи кинематики используется метод Левенберга – Марквардта. Кинематические параметры манипулятора представлены набором ограничений, которые накладываются на вектор обобщенных координат. Ограничение по положению рабочего органа можно записать как:

$$p_i(\theta) = p_i^d,$$

где $p \in \mathbb{R}^3$ – текущее положение рабочего органа; $p^d \in \mathbb{R}^3$ – целевое положение в пространстве. Для ограничения ориентации

$$R_i(\theta) = R_i^d,$$

где $R_i \in SO(3)$ – ориентация рабочего органа; $R_i^d \in SO(3)$ – целевая ориентация в пространстве. В обоих случаях вектор невязок $e_i(\theta)$ может быть определен как

$$e_i(\theta) = \begin{cases} p_i^d - p_i(\theta) \\ \alpha(R_i^d R_i(\theta)^T) \end{cases},$$

где $\alpha(R) \in \mathbb{R}^3$ для произвольного $R \in SO(3)$ – эквивалентный вектор угла-оси.

Принимая число всех ограничений $3m$, определим вектор невязок $e(\theta) \in \mathbb{R}^{3m}$ как:

$$e(\theta) = [e_1^T(\theta) \quad e_1^T(\theta) \quad \dots \quad e_m^T(\theta)]^T.$$

Решение обратной задачи кинематики сводится к решению следующего нелинейного уравнения:

$$e(\theta) = 0.$$

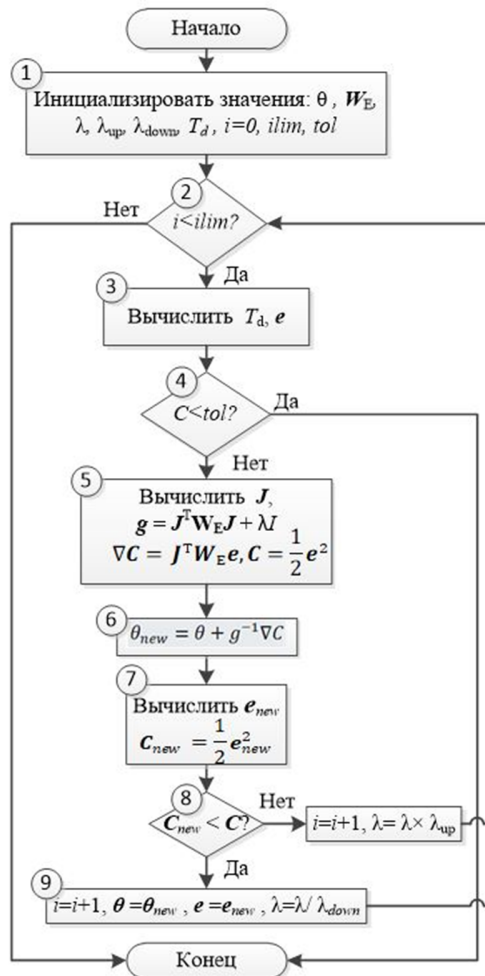


Рис. 7. Блок-схема алгоритма Левенберга – Марквардта
Fig. 7. Block diagram of the Levenberg – Marquardt algorithm

Алгоритм метода Левенберга – Марквардта представлен на рисунке 7 в виде блок-схемы и включает следующие этапы:

Шаг 1. Инициализация значений, которые будут использоваться для настройки коэффициента демпфирования: параметров θ , весовой матрицы W_E , параметра λ , а также λ_{up} и λ_{down} .

Шаг 2. Проверка счетчика итераций: если счетчик итераций не достиг максимального количества шагов, то переход к шагу 3, если достиг, то переход к концу алгоритма.

Шаг 3. Расчет матрицы преобразования T_i и оценка вектора невязок e .

Шаг 4. Проверка: если $C < tol$, то переход к концу; если $C > tol$, то переход к шагу 5.

Шаг 5. Вычисление матрицы Якобиана J , $g_k = JW_E J + \lambda I$, функции стоимости, $\nabla C = J^T W_E e$ и $C = \frac{1}{2} e^2$.

Шаг 6. Вычисление нового значения $\theta_{new} = \theta + g^{-1} \nabla C$.

Шаг 7. Оценка нового вектора невязок e_{new} в точке, заданной θ_{new} и вычисление функции оценки в новой точке $C_{new} = \frac{1}{2} e_{new}^2$.

Шаг 8. Проверка: если $C_{new} > C$, то отклонить шаг, сохранить старый параметр предположения θ и старые невязки e , и настроить $\lambda = \lambda \times \lambda_{up}$ и переход к шагу 2.

Шаг 9. Проверка: если $C_{new} < C$, то принимать шаг $\theta = \theta_{new}$, и установить $e = e_{new}$ и $\lambda = \lambda / \lambda_{down}$ и переход к шагу 2.

6. Адаптивная нейро-нечеткая система вывода «АНСВ» для рушения ОЗК. В этой работе для прогнозирования углов относительно поворота и крутящего момента сочленений звеньев манипулятора, необходимых для выполнения заданной траектории в рабочей зоне, используется АНСВ как другой способ. АНСВ состоит из нейронов, соединенных направленными связями, в которых каждый нейрон выполняет определенную функцию на своих входящих сигналах для генерации выходного сигнала одного нейрона. Для прогнозирования углов относительно поворота сочленений звеньев манипулятора используются 5 АНСВ. В каждой АНСВ значения координаты (p_x, p_y, p_z) , задающие положение и значения (γ, β, α) , задающие ориентацию схвата манипулятора работают как входные значения, а значения углов относительно поворота звеньев $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$ работают как выходные значения. На рисунке 8 представлены структуры АНСВ с шестью входами и одним выходом для прогнозирования ОЗК, а сводка параметров и факторов, учитываемых при разработке модели АНСВ, представлена в таблице 3.

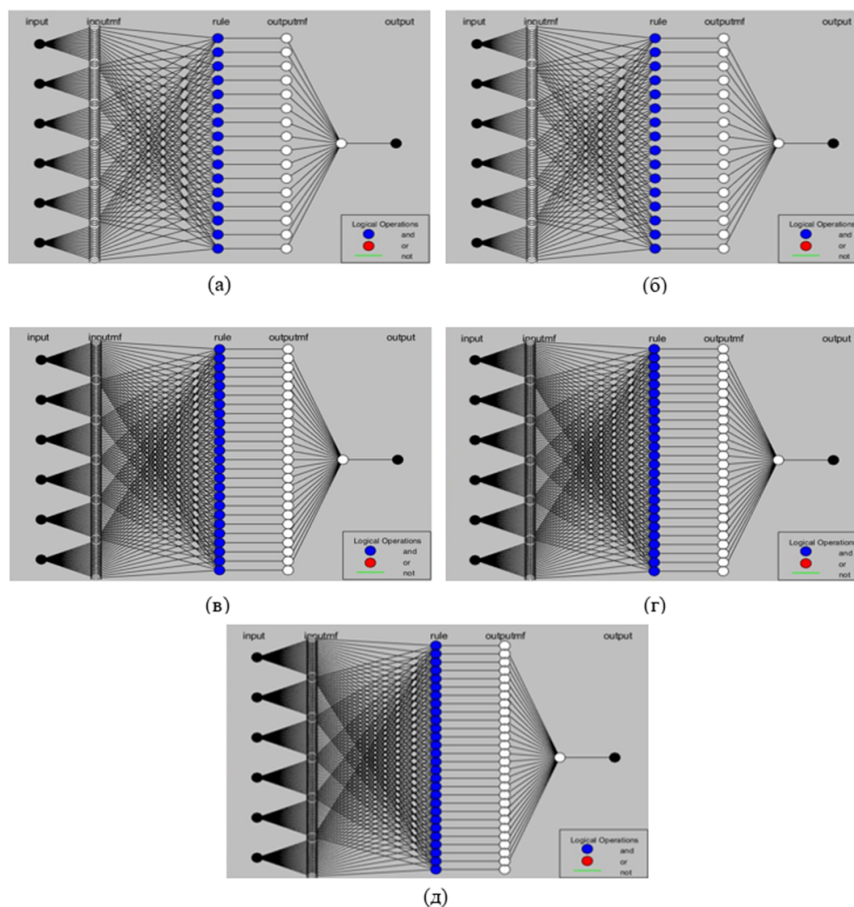


Рис. 8. Структуры АНСВ (а) звена-1, (б) звена-2, (в) звена-3, (г) звена-4 и (д) звена-5 для решения ОЗК
Fig. 8. Structures of ANFIS (a) link-1, (b) link-2, (c) link-3, (d) link-4 and (e) link-5 for solving inverse kinematics problem

На рисунке 9 приведена блок-схема алгоритма программного компонента обратной задачи кинематики.

Таблица 3. Параметры и факторы моделей АНСВ для ОЗК
 Table 3. Parameters and Factors of ANFIS Models for inverse kinematics problem

Звено	АНСВ-1	АНСВ-2	АНСВ-3	АНСВ-4	АНСВ-5
Количество входов	6	6	6	6	6
Количество выходов	1	1	1	1	1
Тип функций принадлежности	Gaussian	Gaussian	Gaussian	Gaussian	Gaussian
Количество функций принадлежности	23	21	25	26	31
Количество узлов	331	303	359	373	443

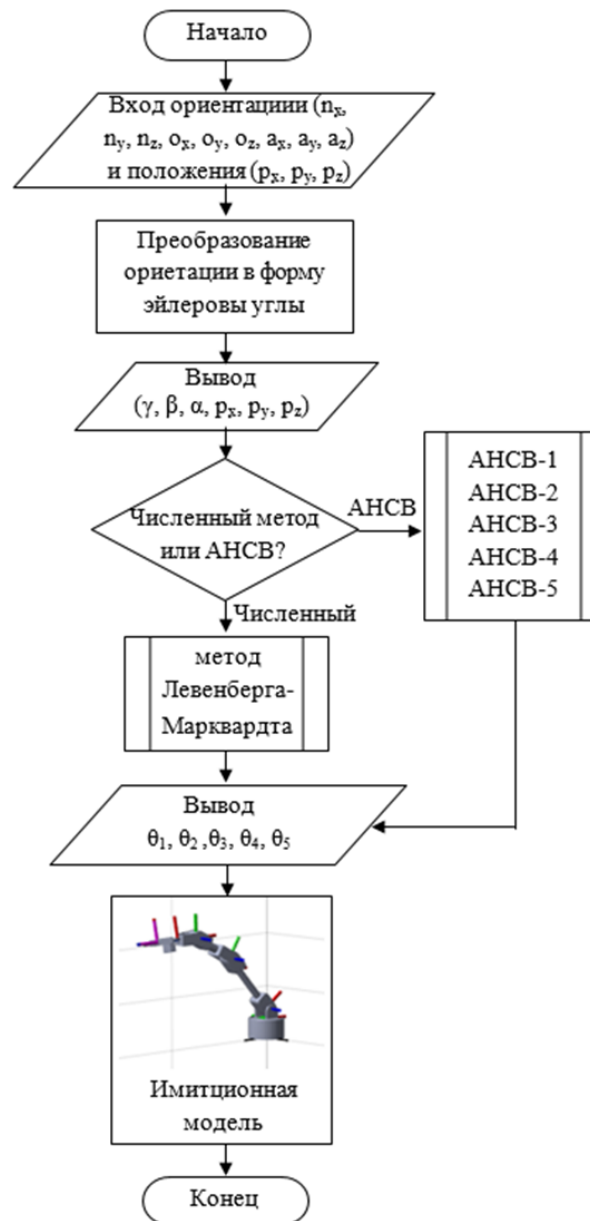


Рис. 9. Блок-схема алгоритма программы обратной задачи кинематики

Fig. 9. Block diagram of the program algorithm for the inverse kinematics

7. Разработка программного компонента решения обратной задачи динамики манипулятора. Метод Ньютона – Эйлера для рушения ОЗД. Определение самих управляющих крутящих моментов, обеспечивающих требуемые законы движения звеньев, является обратной задачей динамики. Алгоритм Ньютона – Эйлера состоит из двух частей – прямой рекурсии и обратной рекурсии. Прямая рекурсия позволяет определить скорости и ускорения звеньев, обобщенные силы и моменты сил, действующих на

все звенья 1 – n . Рекурсивные уравнения имеют следующий вид:

$$\begin{aligned}\omega_{i+1} &= R_i^{i+1} \omega_i + \dot{\theta}_{i+1} \hat{Z}_{i+1} \\ \dot{\omega}_{i+1} &= R_i^{i+1} \dot{\omega}_i + R_i^{i+1} \omega_i \times \dot{\theta}_{i+1} \hat{Z}_{i+1} + \ddot{\theta}_{i+1} \hat{Z}_{i+1} \\ \dot{v}_{i+1} &= R_i^{i+1} (\dot{\omega}_i \times b_{i+1}^{i+1} + \omega_i \times b_{i+1}^i) + \dot{v}_i \\ \dot{v}_{C_{i+1}} &= \dot{\omega}_{i+1} \times b_{i+1}^{C_{i+1}} + \omega_{i+1} \times (\omega_{i+1} \times b_{C_{i+1}}^{i+1}) + \dot{v}_{i+1},\end{aligned}$$

где ω_i – угловая скорость звена i ; $\dot{\omega}_i$ – угловое ускорение звена i ; R_i^{i+1} – матрица перехода от системы координат звена i в систему координат звена $i+1$; $\dot{\theta}_i$ – угловая скорость вращения шарнира i ; $\ddot{\theta}_i$ – угловое ускорение вращательного шарнира i ; \hat{Z}_i – ось вращательного сочленения i ; b_{i+1}^{i+1} – вектор от начала системы координат i до начала системы координат $i+1$; $b_{C_i}^i$ – вектор от начала системы координат i до центра масс звена i ; \dot{v}_i – ускорение начала системы координат звена i ; $\dot{v}_{C_i}^i$ – ускорение центра масс звена i .

Зная линейное и угловое ускорения центра масс для каждого звена, можно использовать уравнения Ньютона – Эйлера для вычисления моментов сил инерции, приложенных в центре масс каждого из звеньев. Таким образом, имеем

$$\begin{aligned}F_{i+1} &= m_{i+1} \dot{v}_{C_i} \\ N_{i+1} &= I_{i+1} \dot{\omega}_{i+1} + \omega_{i+1} \times I_{i+1} \omega_{i+1},\end{aligned}$$

где F_i – общая внешняя сила на звено i ; m_i – общая масса звена i ; N_i – суммарный внешний крутящий момент на звене i ; I_i – тензор инерции звена i относительно его центра масс. Обратная рекурсия требует вычисления сил и моментов сил взаимодействия, а также моментов, развиваемых в приводах, начиная от конечного звена манипулятора (n) и обратно к основанию.

$$\begin{aligned}f_i &= R_{i+1}^i f_{i+1} + F_i, \\ n_i &= N_i + R_{i+1}^i n_{i+1} + b_{C_i}^i \times F_i + b_{i+1}^i \times R_{i+1}^i f_{i+1}, \\ \tau_i &= n_i^T \hat{Z}_i,\end{aligned}$$

где f_i – сила, действующая на звено (i) со стороны звена ($i+1$); n_i – момент, действующий на звено (i) со стороны звена ($i+1$); τ_i – входной крутящий момент в шарнире (i). На рисунке 10 представлена блок-схема алгоритма для метода Ньютона – Эйлера, который используется для решения обратной задачи динамики манипулятора [2].

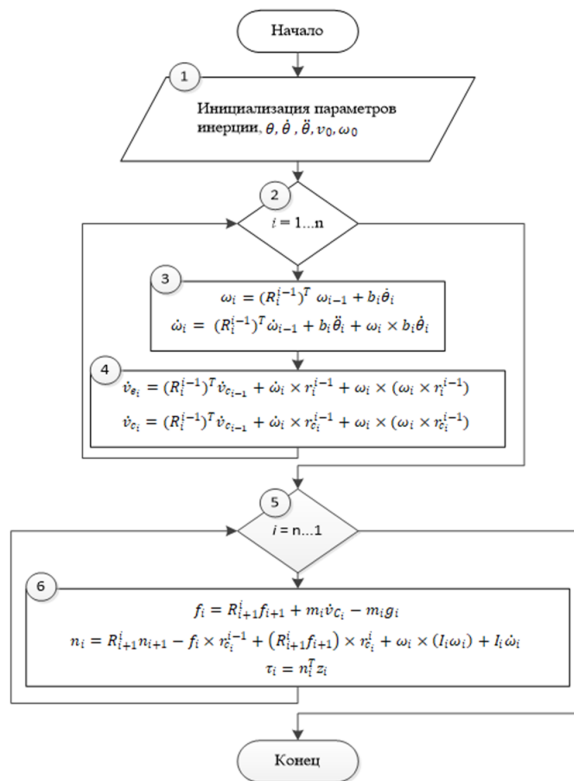


Рис. 10. Блок-схема алгоритма Ньютона – Эйлера для решения обратной задачи динамики манипулятора
Fig. 10. Block diagram of the Newton – Euler algorithm for solving the inverse dynamics problem of manipulator

8. Адаптивная нейро-нечеткая система вывода «АНСВ» для рушения ОЗД. Для прогнозирования и крутящего момента сочленений звеньев манипулятора используются 5 АНСВ. В каждой АНСВ, значения

$(\theta_i, \dot{\theta}_i, \ddot{\theta}_i)$, задающие векторы положений, скоростей и ускорений в шарнирах соответственно работают как входные значения, а значения (τ_i) , задающие векторы крутящего момента, работают как выходные значения. На рисунке 11 представлены структуры АНСВ с двенадцатью входами и одним выходом для прогнозирования ОЗД, а сводка параметров и факторов, учитываемых при разработке модели АНСВ, представлена в таблице 4.

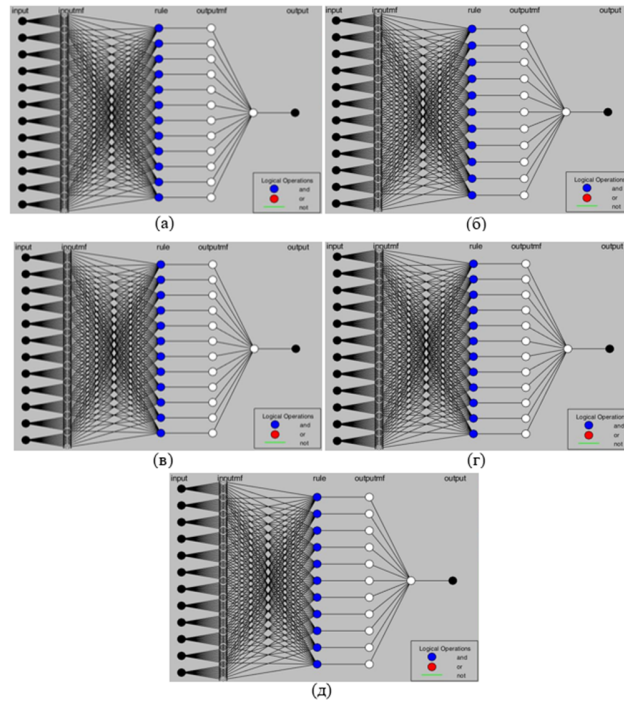


Рис. 11. Структуры АНСВ (а) звена-1, (б) звена-2, (в) звена-3, (г) звена-4 и (д) звена-5 для решения ОЗД
 Fig. 11. Structures of ANFIS (a) link-1, (b) link-2, (c) link-3, (d) link-4 and (e) link-5 for solving inverse dynamics problems

Таблица 4. Параметры и факторы моделей АНСВ для ОЗД
 Table 4. Parameters and Factors of ANFIS Models for inverse dynamics problem

Звено	АНСВ-1	АНСВ-2	АНСВ-3	АНСВ-4	АНСВ-5
Количество входов	12	12	12	12	12
Количество выходов	1	1	1	1	1
Тип функций принадлежности	Gaussian	Gaussian	Gaussian	Gaussian	Gaussian
Количество функций принадлежности	12	11	12	12	11
Количество узлов	327	301	327	327	301

На рисунке 12 приведена блок-схема алгоритма программного компонента обратной задачи динамики.

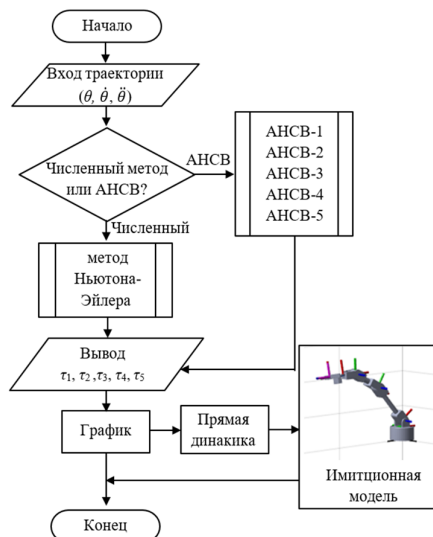


Рис. 12. Блок-схема алгоритма программы обратной задачи динамики
 Fig. 12. Block diagram of the program algorithm for the inverse dynamics

9. Результаты моделирования. При запуске приложения на экране появляется форма, см. рисунок 5. В режиме прямой кинематики пользователь определяет значения углов, перетаскивая ползунок в пяти суставах соответственно или вводя значение углов через текстовое поле (показано в окне в верхней центральной части экрана). При изменении значений углов манипулятор в 3D-пространстве будет перемещаться в соответствующие положения и ориентации в режиме реального времени (отображается в окне в правой части экрана). На основе заданных значений углов звеньев программа вычисляет положение и ориентацию рабочего органа при нажатии кнопки «Calculate». Как показано на рисунке 13, значения углов пяти звеньев манипулятора и их соответствующее положение и ориентация рабочего органа показаны в текстовых полях.

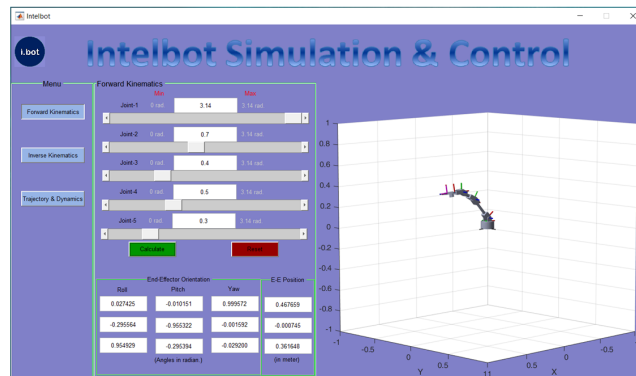


Рис. 13. Программа в режиме прямой кинематики
 Fig. 13. Program in forward kinematics mode

В режиме обратной кинематики определяется вектор обобщенных координат сочленений звеньев, который позволяет манипулятору достичь желаемых положения и ориентаций рабочего органа. Желаемое положение и ориентацию рабочего органа манипулятора можно настроить, введя положение и ориентацию с помощью текстовых полей (показано в окне в верхней левой части экрана) или исходя из результата режима прямой кинематики (рисунок 14). В методе расчета выбор включает возможность использования метода АНСВ или численного метода. Затем вектор обобщенных координат сочленений звеньев будет рассчитано в режиме реального времени по обратному кинематическому алгоритму программы-скрипта (см. рисунок 9).

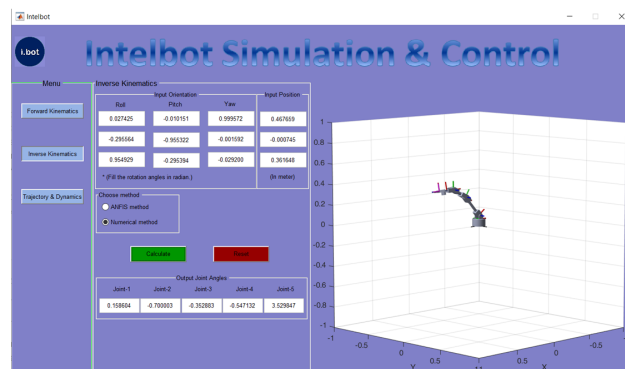


Рис. 14. Программа в режиме обратной кинематики
 Fig. 14. Program in inverse kinematics mode

Режим обратной динамики определяет требуемые крутящие моменты для обеспечения движения исполнительного звена манипулятора по заданной траектории. В этом режиме, во-первых, необходимо настроить траекторию рабочего органа в обратной динамике. Здесь в программе используется метод планирования траектории на основе вектора обобщенных координат шарнирных сочленений звеньев манипулятора. Положения, скорости и ускорения сочленений будут рассчитываться с помощью программы формирования траектории. В методе расчета выбор включает возможность использования метода АНСВ или численного метода. Крутящие моменты каждого сочленения, требуемые для обеспечения движения по заданной траектории, будут рассчитываться по обратному динамическому алгоритму программы-скрипта (см. рисунок 12). В виртуальной среде можно отобразить траекторию движения

рабочего органа манипулятора, рисунок 15. А также требуемые крутящие моменты, положение, угловая скорость и угловое ускорение каждого звена могут отобразиться с помощью графиков, которые могут служить ориентиром для проектирования системы управления и конструкции манипулятора.

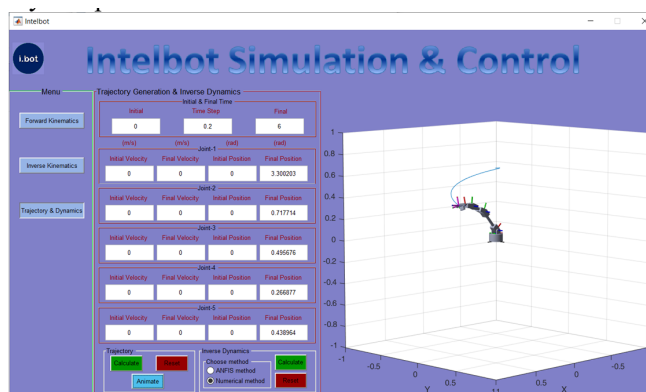


Рис. 15. Программа в режиме обратной динамики
Fig. 15. Program in inverse dynamics mode

Заключение. В данной работе использован язык программирования MATLAB для программирования математического моделирования. Для реализации 3D-модели манипулятора использована САПР-программа «SolidWorks». Разработаны программные модули для компьютерного и имитационного моделирования перемещения манипулятора в рабочей зоне со следующими функциями: визуализация 3D-моделирования; взаимодействие с пользователем через графический интерфейс; расчет ОЗК манипулятора на основе метода Левенберга – Марквардта; расчет ОЗК манипулятора на основе АНСВ; расчет ОЗД манипулятора на основе метода Ньютона – Эйлера и визуализация результатов; расчет ОЗД манипулятора на основе АНСВ и визуализация результатов; формирование траектории для целевого положения и ориентации рабочего органа манипулятора; имитационное моделирование движения манипулятора по заданной траектории.

Список литературы

1. Раин Т., Довгаль В. М., Ян Н. С. 2018. Моделирование кинематического управления роботоманипулятором «Intelbot» на основе адаптивной нейро-нечеткой системы вывода (ANFIS). Научные ведомости БелГУ. Сер. Экономика. Информатика, 45(3): 497–509.
2. Раин Т., Ян Н. С. 2019. Моделирование динамики манипулятора с использованием адаптивной нейро-нечеткой системы вывода. Научный журнал «Моделирование, оптимизация и информационные технологии», 7(4): 1–14.
3. Adam S. A., Zhou Ji-Pin, Zhang Yi-hua. 2017. Modeling and simulation of 5DOF robot manipulator and trajectory using MATLAB and CATIA. 3rd International Conference on Control, Automation and Robotics (ICCAR): 36-40. DOI:10.1109/ICCAR.2017.7942657.
4. Amit T., Vinod K. 2020. Design, simulation, and analysis of a 6-axis robot using robot visualization software. IOP Conference Series: Materials Science and Engineering, 872: 1–9. DOI 10.1088/1757-899X/872/1/012040
5. Chauhan S. S., Khare A. K. 2020. Kinematic Analysis of the ABB IRB 1520 Industrial Robot Using RoboAnalyzer Software. Evergreen. Joint Journal of Novel Carbon Resource Sciences & Green Asia Strategy, 07(04): 510–518.
6. Dey U., Cheruvu S. K. 2020. A web-based integrated GUI for 3D modeling, kinematic study, and control of robotic manipulators. Computer Applications in Engineering Education, 28(4): 1028–1040.
7. George E. I., Smith R., Levy J. S., Brand T. C. 2019. Simulation in Robotic Surgery. Comprehensive Healthcare Simulation: Surgery and Surgical Subspecialties, Comprehensive: 191–220. https://doi.org/10.1007/978-3-319-98276-2_17
8. Kuruganti Y. S., Ganesh A. S. D., Ivan D. D., Chittawadigi R. G. 2021. Effective Teaching of Homogenous Transformations and Robot Simulation Using Web Technologies. Congress on Intelligent Systems. CIS 2020. Advances in Intelligent Systems and Computing, 1335: 687–699.

9. Maram S., Yashaswi K., Chittawadigi R., Saha S. 2019. Effective Teaching and Learning of Homogenous Transformation Matrix using RoboAnalyzer Software. In Proceedings of the Advances in Robotics (AIR 2019): 4th International Conference of the Robotics Society: 1-5.
<https://doi.org/10.1145/3352593.3352611>
10. Mickoski I., Mickoski H., Djidrov M. 2018. Dynamic modeling and simulation of three-member robot manipulator. *Mathematical Models in Engineering*, 4(4), Issue 4: 183–190.
<https://doi.org/10.21595/mme.2018.20319>
11. Othayoth R. S., Chittawadigi R. G., Joshi R. P., Saha S. K. 2017. Robot kinematics made easy using RoboAnalyzer software. *Computer Applications in Engineering Education* 25(5): 669–680.
12. Sabnis C. V., Anjana N. R., Talli A., Giriyaapur A. C. 2021. Modelling and Simulation of Industrial Robot Using SolidWorks. *Advances in Industrial Machines and Mechanisms. Select Proceedings of IPROMM 2020*: 173–182. https://doi.org/10.1007/978-981-16-1769-0_16
13. Shivani R., Shruti T., Chaitali P. 2017. A Parallel Study of Designing and Simulation of Industrial Robotics. *International Journal of Electrical, Electronics and Data Communication (IJEEDC)*, 5(2): 55–60.
14. Sinha S. S., Chittawadigi R. G., Saha S. K. 2018. Inverse Kinematics for General 6 R Manipulators in RoboAnalyzer. *The 5th Joint International Conference on Multibody System Dynamics (IMSD 2018)*: 1–9.
15. Talli A., Marebal D. 2021. Kinematic Analysis and Simulation of Robotic Manipulator Based on RoboAnalyzer. *Smart Sensors Measurements and Instrumentation. Lecture Notes in Electrical Engineering*, 750: 59–69.
16. Xu X. et al., 2018. Robotic kinematics teaching system with virtual reality, remote control and an on-site laboratory, *Int. J. Mech. Eng. Educ.* 48(3): 197–220.
17. Yang L., Zhang X. 2015. Dynamics Modeling and Simulation of Robot Manipulator. *Intelligent Robotics and Applications. Lecture Notes in Computer Science*, 9246: 524–535.
https://doi.org/10.1007/978-3-319-22873-0_47

References

1. Rain T., Dovgal V. M., Yan N. S. 2018. Modelling of the adaptive neuro-fuzzy inference system based control of 5-dof robotic manipulator “Intelbot”. *Belgorod State University Scientific Bulletin. (Economics. Information Technologies)*, 45(3): 497–509 (in Russian).
2. Rain T., Yan N. S. 2019. Dynamic modelling of manipulator using adaptive neuro fuzzy inference system Modeling, Optimization And Information Technology (MOIT), 7(4): 1–14 (in Russian).
3. Adam S. A., Zhou Ji-Pin, Zhang Yi-hua. 2017. Modeling and simulation of 5DOF robot manipulator and trajectory using MATLAB and CATIA. *3rd International Conference on Control, Automation and Robotics (ICCAR)*: 36-40. DOI:10.1109/ICCAR.2017.7942657.
4. Amit T., Vinod K. 2020. Design, simulation, and analysis of a 6-axis robot using robot visualization software. *IOP Conference Series: Materials Science and Engineering*, 872: 1–9.
DOI 10.1088/1757-899X/872/1/012040
5. Chauhan S. S., Khare A. K. 2020. Kinematic Analysis of the ABB IRB 1520 Industrial Robot Using RoboAnalyzer Software. *Evergreen. Joint Journal of Novel Carbon Resource Sciences & Green Asia Strategy*, 07(04): 510–518.
6. Dey U., Cheruvu S. K. 2020. A web-based integrated GUI for 3D modeling, kinematic study, and control of robotic manipulators. *Computer Applications in Engineering Education*, 28(4): 1028–1040.
7. George E. I., Smith R., Levy J. S., Brand T. C. 2019. Simulation in Robotic Surgery. *Comprehensive Healthcare Simulation: Surgery and Surgical Subspecialties, Comprehensive*: 191–220.
https://doi.org/10.1007/978-3-319-98276-2_17
8. Kuruganti Y. S., Ganesh A. S. D., Ivan D. D., Chittawadigi R. G. 2021. Effective Teaching of Homogenous Transformations and Robot Simulation Using Web Technologies. *Congress on Intelligent Systems. CIS 2020. Advances in Intelligent Systems and Computing*, 1335: 687–699.
9. Maram S., Yashaswi K., Chittawadigi R., Saha S. 2019. Effective Teaching and Learning of Homogenous Transformation Matrix using RoboAnalyzer Software. In Proceedings of the Advances in Robotics (AIR 2019): 4th International Conference of the Robotics Society: 1–5.
<https://doi.org/10.1145/3352593.3352611>

10. Mickoski I., Mickoski H., Djidrov M. 2018. Dynamic modeling and simulation of three-member robot manipulator. *Mathematical Models in Engineering*, 4(4), Issue 4: 183–190.
<https://doi.org/10.21595/mme.2018.20319>
11. Othayoth R. S., Chittawadigi R. G., Joshi R. P., Saha S. K. 2017. Robot kinematics made easy using RoboAnalyzer software. *Computer Applications in Engineering Education* 25(5): 669–680.
12. Sabnis C. V., Anjana N. R., Talli A., Giriyaapur A. C. 2021. Modelling and Simulation of Industrial Robot Using SolidWorks. *Advances in Industrial Machines and Mechanisms. Select Proceedings of IPROMM 2020*: 173–182. https://doi.org/10.1007/978-981-16-1769-0_16
13. Shivani R., Shruti T., Chaitali P. 2017. A Parallel Study of Designing and Simulation of Industrial Robotics. *International Journal of Electrical, Electronics and Data Communication (IJEEDC)*, 5(2): 55–60.
14. Sinha S. S., Chittawadigi R. G., Saha S. K. 2018. Inverse Kinematics for General 6 R Manipulators in RoboAnalyzer. *The 5th Joint International Conference on Multibody System Dynamics (IMSD 2018)*: 1–9.
15. Talli A., Marebal D. 2021. Kinematic Analysis and Simulation of Robotic Manipulator Based on RoboAnalyzer. *Smart Sensors Measurements and Instrumentation. Lecture Notes in Electrical Engineering*, 750: 59–69.
16. Xu X. et al., 2018. Robotic kinematics teaching system with virtual reality, remote control and an on-site laboratory, *Int. J. Mech. Eng. Educ.* 48(3): 197–220.
17. Yang L., Zhang X. 2015. Dynamics Modeling and Simulation of Robot Manipulator. *Intelligent Robotics and Applications. Lecture Notes in Computer Science*, 9246: 525–535.
https://doi.org/10.1007/978-3-319-22873-0_47

Конфликт интересов: о потенциальном конфликте интересов не сообщалось.

Conflict of interest: no potential conflict of interest related to this article was reported.

Поступила в редакцию 26.12.2022

Поступила после рецензирования 13.02.2023

Принята к публикации 17.02.2023

Received 26.12.2022

Revised 13.02.2023

Accepted 17.02.2023

СВЕДЕНИЯ ОБ АВТОРЕ

Ту Раин – научный соискатель кафедры математического и программного обеспечения информационных систем, Белгородский государственный национальный исследовательский университет
ул. Победы, 85, Белгород, 308015, Россия

INFORMATION ABOUT THE AUTHOR

Thu Rain – Research Applicant, Department of Mathematical and Software Information Systems, Belgorod National Research University, Belgorod, Russia