

УДК 004.415.25

**Вернер Денис Федорович**

Амурский государственный университет

г. Благовещенск, Россия

*E-mail:* [11rikoshet11@gmail.com](mailto:11rikoshet11@gmail.com)**Галаган Татьяна Алексеевна**

Амурский государственный университет

г. Благовещенск, Россия

*E-mail:* [tgalagan@yandex.ru](mailto:tgalagan@yandex.ru)**Verner Denis Fyodorovich**

Amur State University

Blagoveshchensk, Russia

*E-mail:* [11rikoshet11@gmail.com](mailto:11rikoshet11@gmail.com)**Galagan Tatyana Alekseevna**

Amur State University

Blagoveshchensk, Russia

*E-mail:* [tgalagan@yandex.ru](mailto:tgalagan@yandex.ru)**ПРИМЕНЕНИЕ ПРЕДСТАВЛЕНИЯ ДЕНАВИТА – ХАРТЕНБЕРГА В ПРИЛОЖЕНИИ  
ДЛЯ ПОИСКА РЕШЕНИЯ ПРЯМОЙ ЗАДАЧИ КИНЕМАТИКИ****APPLICATION OF THE DENAVIT-HARTENBERG PARAMETERS IN AN PROGRAM  
FOR FINDING A SOLUTION TO A DIRECT KINEMATICS PROBLEM**

*Аннотация.* В статье рассмотрено использование представления Денавита-Хартенберга для автоматизации процесса поиска решения прямой задачи кинематики промышленного манипулятора.

*Abstract.* The article discusses the use of the Denavit-Hartenberg parameters for automating the process of finding a solution to the forward kinematics problem of an industrial robot.

*Ключевые слова:* манипулятор, звенья манипулятора, прямая задача кинематики, разработка приложения.

*Key words:* industrial robot, forward kinematics solution, industrial robot links, app development.

**Введение**

Применение манипуляторов на производстве позволяет автоматизировать практически любой процесс. Перед их использованием необходимо провести расчеты, – в частности, поиск решения прямой и обратной задач кинематики. Алгоритм поиска решения хорошо

изучен и широко применяем. Однако процесс его поиска вручную может быть трудоемким. Упрощению внедрения манипуляторов в производственный процесс, а также разработки их новых моделей может помочь программа, дающая возможность по виду манипулятора решать для него прямую задачу кинематики.

В настоящее время существуют программные продукты, позволяющие решать прямую задачу кинематики [1-3], но рассматриваемое в нашей работе приложение отличается от аналогов наглядностью представления изучаемого манипулятора, а также поиском решения в аналитической форме, что позволит использовать его при управлении манипулятором.

### Функции приложения

Функционал приложения позволяет пользователю осуществлять: сборку манипулятора с помощью конструктора; поиск решения прямой задачи кинематики в аналитическом виде; вывод полученной модели и решения прямой задачи на экран, а также в текстовый файл.

Для удобства пользователя и ограничения возможных вариантов манипуляторов программа обладает встроенной библиотекой компонентов, содержащей типовые звенья манипулятора – жесткое звено и поворотные шарниры.

Основной функцией приложения является поиск прямого решения задачи кинематики в аналитическом виде (в виде формул, связывающих координаты положения нулевого звена манипулятора и его рабочего органа с помощью обобщенных координат).

Полученные результаты выводятся для чтения на экран, а также в файл. Из-за потенциально большого размера формул вывод на экран ограничен одним параметром зараз – координатой рабочего органа манипулятора в системе координат его основания или одним из векторов, определяющих направление рабочего органа. В файл запись решения производится целиком.

Функциональная декомпозиция приложения показана на рис. 1.

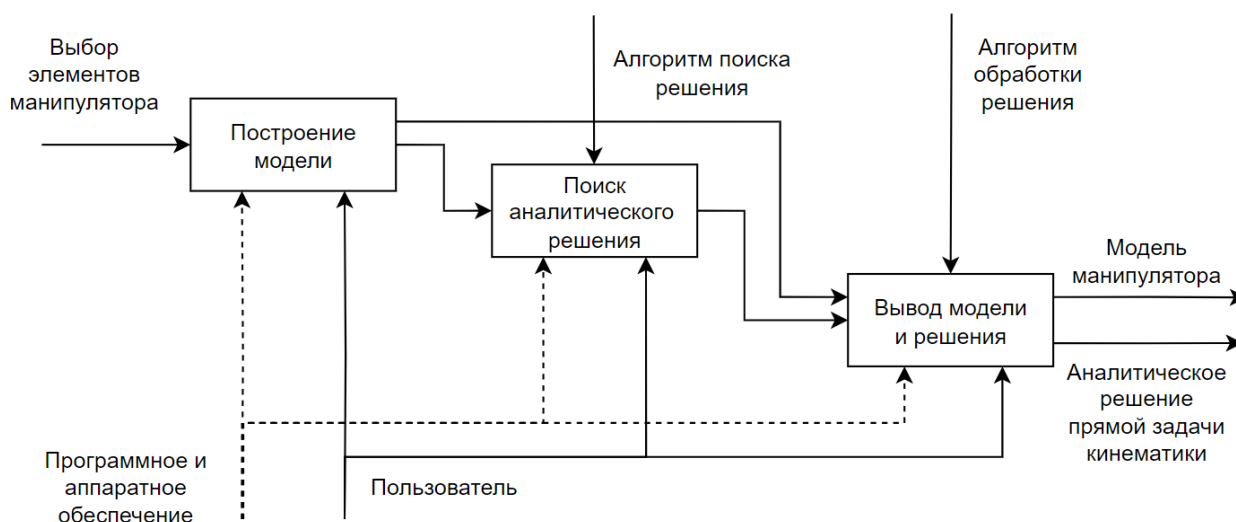


Рис. 1. Функциональная декомпозиция приложения.

**Применение метода представления Денавита – Хартенберга  
в решении прямой задачи кинематики**

Для решения прямой задачи кинематики используется представление Денавита – Хартенберга, позволяющее определить положение рабочего органа манипулятора через положение его основания.

Для описания вращательных и поступательных связей между соседними звеньями используется матричный метод последовательного построения систем координат, связанных с каждым звеном кинематической цепи. Смысл представления Денавита – Хартенберга состоит в формировании однородной матрицы преобразования, описывающей положение системы координат каждого звена относительно системы координат предыдущего звена.

$$A_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

где  $i$  – номер звена, а параметры определяются следующим образом:  $\theta_i$  – присоединенный угол, на который надо повернуть ось  $X_{i-1}$  вокруг оси  $Z_{i-1}$ , чтобы она стала сонаправлена с осью  $X_i$  (знак определяется в соответствии с правилом правой руки);  $d_i$  – расстояние между пересечением оси  $Z_{i-1}$  с осью  $X_i$  и началом  $(i-1)$ -й системы координат, отсчитываемое вдоль оси  $Z_{i-1}$ ;  $a_i$  – линейное смещение – расстояние между пересечением оси  $Z_{i-1}$  с осью  $X_i$  и началом  $i$ -й системы координат, отсчитываемым вдоль оси  $X_i$ , т. е. кратчайшее расстояние между осями  $Z_{i-1}$  и  $Z_i$ ;  $\alpha_i$  – угол, на который надо повернуть ось  $Z_{i-1}$  вокруг оси  $X_i$ , чтобы она стала сонаправленной с осью  $Z_i$  (знак определяется в соответствии с правилом правой руки).

Для вращательных сочленений параметры  $d_i$ ,  $a_i$  и  $\alpha_i$  являются характеристиками сочленения, постоянными для данного типа робота. В то же время  $\theta_i$  является переменной величиной, изменяющейся при движении (вращении)  $i$ -го звена относительно  $(i-1)$ -го.

Решением прямой задачи кинематики будет матрица  $A_0^n$ , которая вычисляется по формуле

$$A_0^n = A_0^1 \cdot A_1^2 \cdot \dots \cdot A_{n-1}^n, \quad (2)$$

где  $n$  – число звеньев манипулятора.

Это дает возможность последовательно преобразовать координаты схвата манипулятора из системы отсчета, связанной с последним звеном, в базовую систему отсчета, являющуюся инерциальной системой координат для рассматриваемой динамической системы.

В большинстве манипуляторов используются три основных вида шарниров. Рассмотрев все их комбинации (рис. 2), получаем матрицы преобразования, которые применяются при поиске аналитического решения в программе.

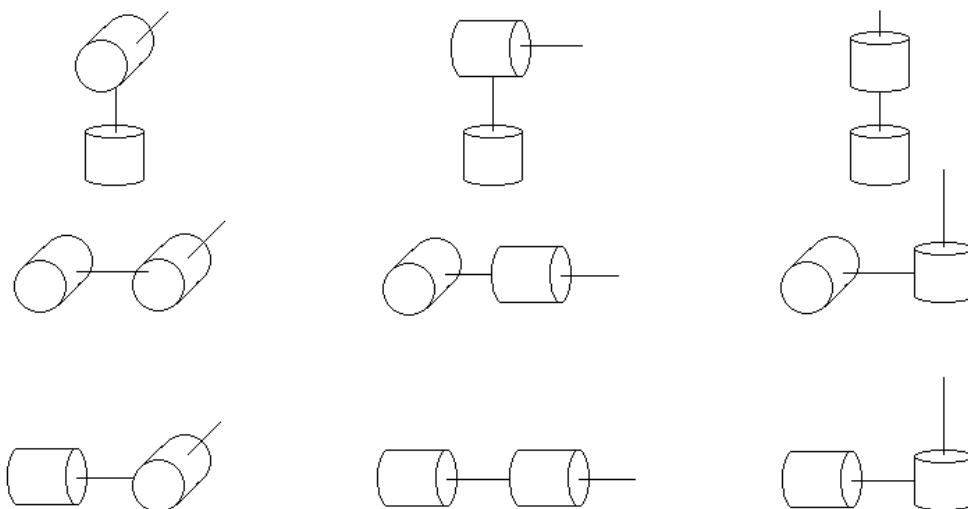


Рис. 2. Возможные комбинации шарниров.

Каждая система координат формируется на основе следующих трех правил: 1) ось  $Z_{i-1}$  направлена вдоль оси  $i$ -го сочленения; 2) ось  $X_i$  перпендикулярна оси  $Z_{i-1}$  и направлена от нее; 3) ось  $Y_i$  дополняет оси  $X_i, Z_i$  до правой декартовой системы координат.

Углы поворота шарниров, а также величины раскрытия телескопических соединений называются обобщенными координатами и обозначаются  $q_i$ , где  $i$  – номер шарнира.

На рис. 3 представлен пример расстановки осей координат и обобщенных координат в соответствии с указанными правилами для трехзвенного манипулятора.

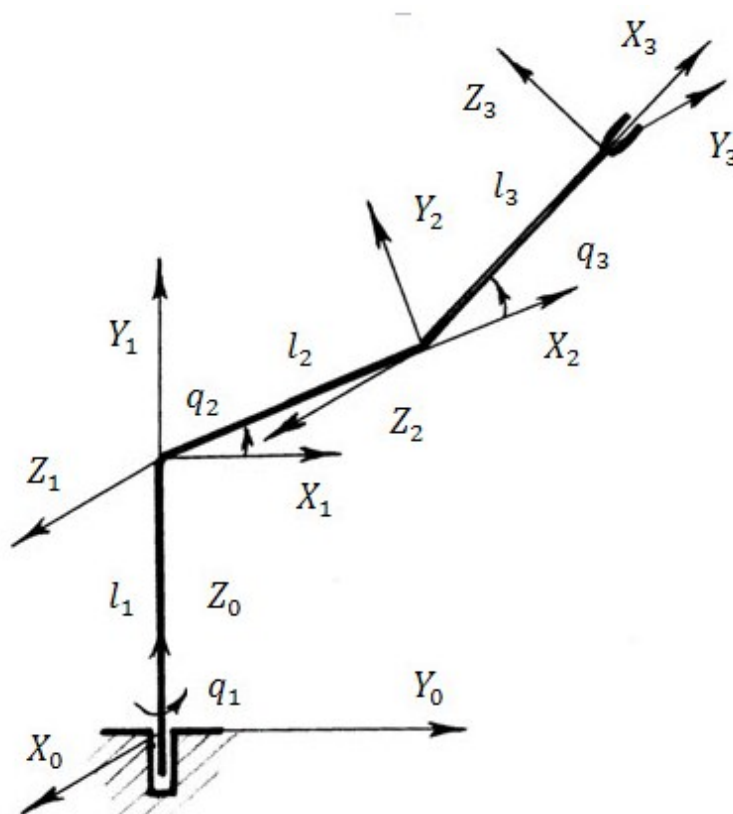


Рис. 3. Пример расположения параметров ДХ для манипулятора.

Параметры матриц преобразования для манипулятора представлены в таблице.

**Параметры для матриц преобразования манипулятора**

Параметр	Системы координат		
	0-1	1-2	2-3
$\alpha$	$\frac{\pi}{2}$	0	$-\pi$
$\theta$	$q_1$	$q_2$	$q_3$
$d$	0	$l_2$	$l_3$
$a$	$l_1$	0	0

Матрицы преобразования вида (1), с учетом параметров таблицы, примут вид:

$$A_1 = \begin{bmatrix} \cos q_1 & 0 & \sin q_1 & 0 \\ \sin q_1 & 0 & -\cos q_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & l_2 \cos q_2 \\ \sin q_2 & \cos q_2 & 0 & l_2 \sin q_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

$$A_3 = \begin{bmatrix} \cos q_3 & 0 & -\sin q_3 & l_3 \cos q_3 \\ \sin q_3 & 0 & \cos q_3 & l_3 \sin q_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Искомым решением будет произведение матриц (3)

$$A_0^3 = \begin{bmatrix} \cos q_1 \cos(q_2 + q_3) & -\sin q_1 & -\cos q_1 \sin(q_2 + q_3) & \cos q_1 (l_3 \cos(q_2 + q_3) + l_2 \cos q_2) \\ \sin q_1 \cos(q_2 + q_3) & \cos q_1 & -\sin q_1 \sin(q_2 + q_3) & \sin q_1 (l_3 \cos(q_2 + q_3) + l_2 \cos q_2) \\ \sin(q_2 + q_3) & 0 & -\cos(q_2 + q_3) & l_1 + l_2 \sin q_2 + l_3 \sin(q_2 + q_3) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Поиск матрицы (4) – основная цель разработанного приложения.

### Проектирование и реализация приложения

Проектирование и реализация приложения выполнены на основе объектно-ориентированного подхода. Итоговый проект включает класс для хранения информации о звеньях, а также класс, хранящий информацию о матрицах преобразования и методах их умножения и обработки.

Рабочая область условно разделена на три зоны: панель элементов, содержащую доступные для моделирования звенья; пространство для отображения кинематической схемы моделируемого манипулятора; область вывода аналитического решения прямой задачи кинематики.

Панель элементов содержит кнопки для добавления в схему жестких звеньев, трех видов шарниров и рабочего органа манипулятора.

При нажатии кнопки очередное звено появляется в кинематической схеме модели и программа добавляет информацию, необходимую для поиска решения.

После нажатия кнопки поиска решения программа составляет матрицы преобразования для каждого подвижного звена, после чего проводит расчеты и преобразует получаемый ответ в удобную для чтения форму. Так как для большого числа звеньев аналитическое решение может быть достаточно объемным, на форму выводится лишь запрашиваемый компонент. Полное решение при этом сохраняется в отдельном текстовом файле.

Для реализации алгоритма была использована среда разработки Visual Studio, языком разработки выбран C#. Платформа пользовательского интерфейса для создания классических приложений для Windows – Windows Forms позволяет быстро собирать и вносить изменения в разработанный макет пользовательского интерфейса.

На рис. 4 и 5 представлены результаты поиска решения с помощью приложения для трехзвенного и шестизвенного манипуляторов.

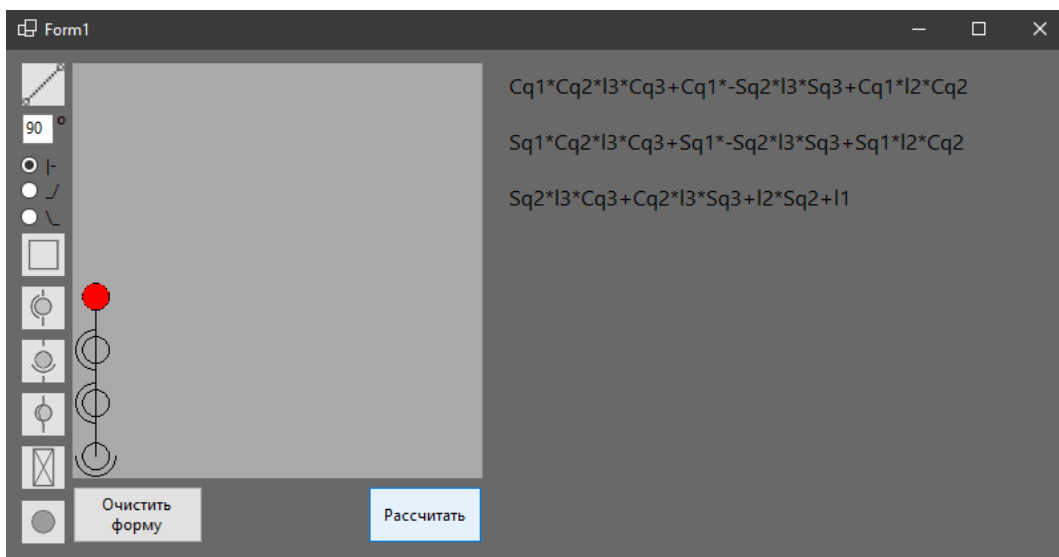


Рис. 4. Результат работы приложения для трехзвенного манипулятора.

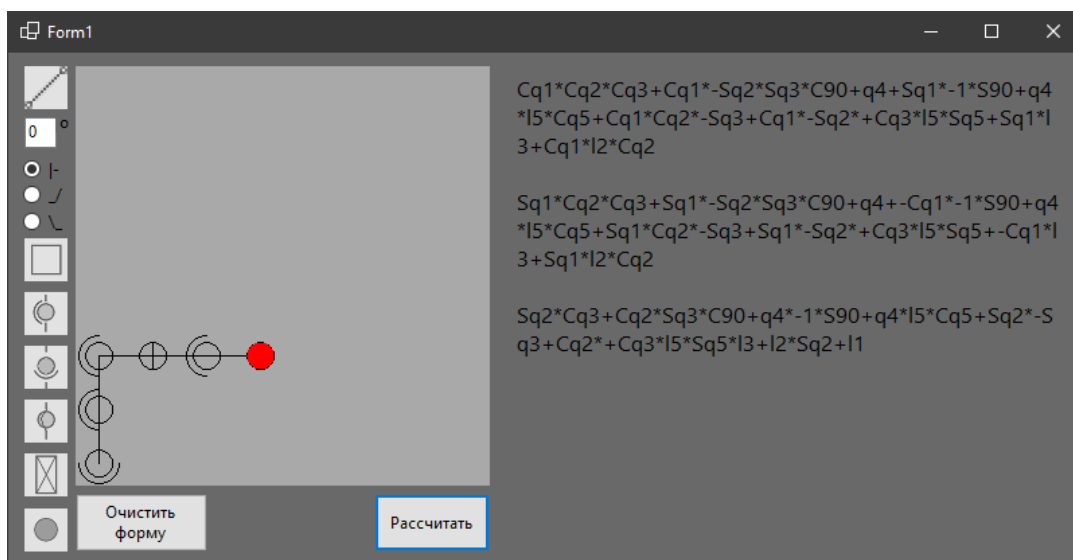


Рис. 5. Результат работы приложения для шестизвенного манипулятора.

Создавать модель манипулятора необходимо с учетом его кинематической схемы, так как программа не делает проверок на правильность и реализуемость.

В дальнейшем рассматривается перенос процесса моделирования в 3D-среду. Для реализации был выбран игровой движок Unity. Скрипты движка также пишутся на языке C#, что значительно облегчит перенос существующих ресурсов и алгоритмов. Движок Unity не так требователен к ресурсам компьютера, как аналоги по типу Unreal, но при этом его возможностей хватает для отрисовки небольшого количества объектов. Для создания манипуляторов в среде Unity могут быть использованы 3D-модели, которые можно найти в открытых источниках.

---

1. Смирнов, П.А., Яковлев, Р.Н. Решение прямой и обратной задач кинематики в системе позиционирования звеньев манипулятора // Мехатроника, автоматизация, управление. – 2019. – 20(12). – С. 732-739.

2. Колтыгин, Д.С., Седельников, И.А. Метод и программа решения прямой и обратной задачи кинематики для управления роботом-манипулятором // Системы. Методы. Технологии. – Братск: БрГУ. – 2020. – №4 (48). – С. 65-74.

3. Пашенко, В.Н., Романов, А.В., Артемьев, С.Ю., Орехов, С.Ю. Решение прямой задачи о положении шестистепенного манипулятора параллельной структуры на базе кривошипно-шатунного механизма // Наука и образование. – МГТУ им. Н.Э. Баумана. – 2015. – №11. – С. 136-152.

4. Вернер, Д.Ф. Разработка приложения для поиска решения прямой задачи кинематики манипулятора // Кооперация науки и общества – путь к модернизации и инновационному развитию. Сб. статей Международной научно-практической конференции – Уфа: Аэтерна, 2024. – С. 36-36.