

*Махонин А.А.,  
Студент магистратуры  
2-й курс факультета нано- и биомедицинских технологий  
ФГБОУ ВО "СГУ имени Н.Г. Чернышевского"  
Российская Федерация, г. Саратов*

**РЕШЕНИЕ ПРЯМОЙ ЗАДАЧИ КИНЕМАТИКИ С ПОМОЩЬЮ  
ПРОГРАММЫ АВТОМАТИЧЕСКОГО УМНОЖЕНИЯ МАТРИЦ  
ДЕНАВИТА-ХАРТЕНБЕРГА**

***Аннотация:** Решение прямой задачи кинематики является первым шагом к эффективному управлению роботом. Упрощение решения такой задачи позволит исследователю сосредоточиться на других, не менее важных, аспектах управления роботом. В данной статье представлен метод полуавтоматического решения прямой задачи кинематики для роботизированной руки.*

***Ключевые слова:** Искусственный интеллект, прямая задача кинематики, робототехника, матрица Денавита-Хартенберга.*

***Abstract:** Solving a robot's forward kinematics is the first step in effective manipulation of its joints. Simplifying the solution of this problem allows the researcher to focus on other important aspects of making the robot move. This article presents a method for semi-automatic calculation of forward kinematics of a robotic arm.*

***Key words:** Artificial intelligence, forward kinematics, robotics, Denavit-Hartenberg matrix.*

**Введение**

Прямая задача кинематики состоит в следующем: необходимо вычислить координаты и углы поворота схвата роботизированного манипулятора в заданной

системе координат по величинам обобщённых углов звеньев манипулятора. Иначе говоря, входными данными для прямой задачи кинематики являются углы поворота сочленений манипулятора, а выходными -- значения координат и углов поворота схвата относительно наперёд заданной начальной точки. Обычно начальная точка совпадает с точкой крепления манипулятора к опоре.

Решение прямой задачи кинематики является первым и важнейшим шагом для расчёта движения роботизированным манипулятором и, в итоге, для управления таким манипулятором.

В данной работе описана реализация программы, позволяющей упростить и ускорить решение прямой задачи кинематики. Программа обеспечивает автоматическое умножения матриц углов поворота. Программа, описанная в данной статье, разработана для роботизированной руки, разработанной НПО "Андроидная техника" (г. Магнитогорск).

### **Математический аппарат**

Математический аппарат, применённый при составлении рассматриваемой программы, сравнительно прост: используются квадратные матрицы размера  $3 \times 3$ , которые описывают поворот того или иного узла манипулятора вокруг соответствующей оси, и матрицы размера  $3 \times 1$ , описывающие длины сочленений. Вычисление тригонометрических функций производится автоматически, с помощью ЭВМ. Выполняя последовательное умножение и сложение матриц углов поворота, можно вычислить координаты схвата и его ориентацию.

Пример вычисления окончательной матрицы поворота  $R$  сочленения после следующих поворотов: сначала на угол  $\alpha$  вокруг оси  $OX$ , затем на угол  $\beta$  вокруг оси  $OY$ , затем на угол  $\gamma$  вокруг оси  $OZ$ .

$$R = R_{y,\beta} \cdot R_{z,\gamma} \cdot R_{x,\alpha} = \begin{pmatrix} \cos \beta & 0 & \sin \beta & \cos \gamma & -\sin \gamma & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & \sin \gamma & \cos \gamma & 0 & 0 & \cos \alpha & -\sin \alpha \\ -\sin \beta & 0 & \cos \beta & 0 & 0 & 1 & 0 & \sin \alpha & \cos \alpha \end{pmatrix}$$

Очевидно, что такие вычисления являются громоздкими и трудоёмкими;

даже при известных значениях тригонометрических функций их умножение, сложение и вычитание требуют значительных временных затрат. Для упрощения таких вычислений была составлена программа, позволяющая автоматически умножать соответствующие матрицы.

### **Описание программы**

Программа составлена на языке программирования С по модульной схеме. Не имеет смысла приводить полный текст программы ввиду его громоздкости; вместо этого ниже представлены словесное описание алгоритма работы и специфические фрагменты текста программы (модули), описывающие типовые действия.

1. Задать и обнулить переменные, описывающие углы поворота и длины сочленений.

2. Задать и обнулить матрицы размера  $3 \times 1$ , описывающие начальные координаты узлов сочленений (каждый узел отстоит от предыдущей точки на расстояние, равное длине соответствующего (предыдущего) сочленения).

3. Задать и обнулить матрицы размера  $3 \times 3$ , описывающие углы поворота сочленений относительно выбранных осей.

4. Запросить значения переменных: углы поворота сочленений и их длины.

5. Подставить значения переменных в соответствующие матрицы и произвести вычисления.

Вначале находятся значения элементов матриц; затем производится умножение матриц между собой: вычисляются матрицы поворота сочленений и умножаются на матрицы координат; затем модифицированные матрицы координат складываются между собой.

6. По вычисленным значениям положения схвата найти углы его поворота относительно координатных осей выбранной системы координат: вычисляются величины проекций схвата на каждую из осей, затем находится отношение проекций к длине сочленения; по данному отношению восстанавливается угол поворота схвата относительно каждой оси.

7. Вывести значения координат и углов поворота схвата. При необходимости вводятся поправки на расстояние между начальной точкой манипулятора, выбранной для удобства вычислений, и действительной начальной точкой.

Пример матрицы начальных координат (соответствующее сочленение направлено вертикально вниз, его длина `first_joint_length` вводится пользователем):

```
first_joint_default_coordinates [0][0] = 0.0;
first_joint_default_coordinates [1][0] = 0.0;
first_joint_default_coordinates [2][0] = -
first_joint_length;
```

Пример матрицы, описывающей углы поворота (`a` -- угол поворота соответствующего узла, вводимый пользователем; умножение на  $\pi$  и деление на  $180^0$  используется для перехода от радиан, применяемых в языке C по умолчанию, к градусам):

```
first_joint_rotation [0][0] = cos (a * PI / 180.0);
first_joint_rotation [0][1] = 0.0;
first_joint_rotation [0][2] = sin (a * PI / 180.0);
first_joint_rotation [1][0] = 0.0;
first_joint_rotation [1][1] = 1.0;
first_joint_rotation [1][2] = 0.0;
first_joint_rotation [2][0] = sin (a * PI / 180.0) * (-
1.0);
first_joint_rotation [2][1] = 0.0;
first_joint_rotation [2][2] = cos (a * PI / 180.0);
```

Пример фрагмента текста программы, описывающего умножение матрицы начальных координат на матрицу углов поворота (`i, j, k, v` -- служебные переменные):

```
for (i = 0; i < 3; i++) {
```

```

    for (j = 0; j < 3; j++) {
    for (k = 0; k < 3; k++) {
    v      =      v      +      first_joint_rotation      [i][k]      *
first_joint_default_coordinates [k][j]; }
    first_joint_coordinates [i][j] = v;
    v = 0.0;
    }
    }

```

Используя приведённые выше фрагменты текста программы на языке С, можно в короткие сроки составить программу для решения прямой задачи кинематики для манипулятора с произвольным конечным количеством сочленений и узлов путём подстановки необходимых модулей в соответствующие места в тексте программы с минимальной их модификацией.

### **Заключение**

С помощью описанной программы достигается значительное упрощение расчётов, необходимых для решения прямой задачи кинематики, и снижается вероятность получить неверный результат из-за человеческой ошибки. В ходе экспериментов (практического решения прямой задачи кинематики) было установлено, что временные затраты для расчётов одного и того же манипулятора снижаются в 10-15 раз, а вероятность ошибки из-за человеческого фактора уменьшается в 3-4 раза. Кроме того, упрощается проверка решения обратной задачи кинематики: исследователю необходимо лишь ввести восстановленные значения обобщённых углов, а затем сравнить восстановленные координаты и ориентацию схвата с заданными.

Модульная структура программы обеспечивает её быстрые и удобные модификацию и расширение для любых роботизированных манипуляторов с произвольным (конечным) числом узлов и звеньев. Простота алгоритма позволяет быстро перевести программу на практически любой существующий язык

программирования с минимальными затратами.

## СПИСОК ЛИТЕРАТУРЫ

1. Фу К. Робототехника: Пер. с англ. / Фу К., Гонсалес Р., Ли К. -- М.: Мир, 1989. -- 624 с., ил. ISBN 5-03-000805-5.
2. Юревич Е.И. Основы робототехники. -- 2-е изд., перераб. и доп. -- СПб.: БХВ-Петербург, 2005. -- 416 с.: ил. ISBN 5-94157-473-8.
3. Шахинпур М. Курс робототехники: пер. с англ. -- М.: Мир, 1990. -- 527 с., ил. ISBN 5-03-001375-X.
4. Механика промышленных роботов: Учеб. пособие для вузов: В 3 кн. /Под ред. К. В. Фролова, Е. И. Воробьева. Кн. 1: Кинематика и динамика/Е. И. Воробьев, С. А. Попов, Г. И. Шевелева. —М.: Высш. шк. , 1988 г. — 304 с.: ил. ISBN 5-06-001201-8.
5. Караваев А.С. Разработка систем на базе микроконтроллеров и ПЛИС. Введение. Учебно-методическое пособие для студентов факультета nano- и биомедицинских технологий Саратовского госуниверситета. Саратов, 2010. 104 с.
6. Kernighan, Brian W. The C programming language. Ritchie, Dennis M., joint author. QA76.73.C15K47 001.6'424 77-28983 ISBN 0-13-110163-3.