

# АЛГОРИТМ РЕШЕНИЯ ОБРАТНОЙ КИНЕМАТИКИ ДЛЯ РОБОТОТЕХНИЧЕСКОГО МАНИПУЛЯТОРА НА МОБИЛЬНОЙ ПЛАТФОРМЕ С УЧЕТОМ ОГРАНИЧЕНИЯМИ

Нгуен Х.Т.<sup>1</sup>, Филипас А.А.<sup>2</sup>

<sup>1</sup>ТПУ, ИШИТР, Группа 8ЕМ31, email: htienguyen@tpu.ru

<sup>2</sup>ТПУ, ИШИТР, Заведующий кафедрой - руководитель отделения на правах кафедры, к.т.н.,  
доцент, email: filipas@tpu.ru

## Аннотация

Работа посвящена разработке нового эвристического алгоритма решения обратной кинематики мобильных манипуляторов (кинематических цепей с подвижными корнями) на основе алгоритма FABRIK. Предложенный алгоритм учитывает проблемы ограничений суставов, ориентации конечного эффектора, избыточности, локальных минимумов и приоритета суставов для оптимизации затрат энергии.

**Ключевые слова:** управление манипулятором, обратная кинематика, мобильный манипулятор, эвристический метод, FABRIK

## Введение

Мобильные манипуляторы привлекают все больше внимания благодаря своей гибкости и широкому спектру применений, охватывающему промышленность, образование, научные исследования и повседневную жизнь. Одной из ключевых задач при их управлении является решение обратной кинематики [1]. Однако для обеспечения мобильности такие платформы, как правило, не оснащаются высокопроизводительными процессорами, поскольку это негативно сказывается на их энергоэффективности. В связи с этим использование матричных вычислительных методов, аналогичных тем, что применяются в стационарных системах, оказывается нецелесообразным.

В подобных ситуациях чаще всего применяются аналитические или эвристические методы, позволяющие получать быстрые результаты при минимальных вычислительных затратах [2-3]. Однако аналитические методы строго зависят от конкретной конфигурации робота и становятся чрезвычайно сложными при увеличении числа степеней свободы. Эвристические методы, напротив, продолжают набирать популярность, поскольку обеспечивают баланс между точностью и вычислительной эффективностью [3-4].

Алгоритм FABRIK (Forward and Backward Inverse Kinematics) изначально был разработан для моделирования кинематических цепей в реальном времени в области компьютерной графики [5]. Он основан на геометрическом подходе и обладает высокой вычислительной скоростью по сравнению с другими итеративными методами. Однако его исходное предназначение не включает управление манипуляторами, что делает его непосредственное применение в мобильной робототехнике неэффективным без соответствующих модификаций.

В данной работе предлагается адаптированный алгоритм на основе FABRIK, оптимизированный для управления мобильными манипуляторами с учетом их физических ограничений.

## Направление разработки алгоритма

**Обеспечение ориентации конечного эффектора и физических ограничений суставов:** в отличие от компьютерной графики, где суставы кинематических цепей обычно моделируются как сферические шарниры без угловых ограничений, суставы современных манипуляторов представляют собой двухмерные вращающиеся соединения с жесткими физическими ограничениями. Это создает значительные трудности при применении алгоритма FABRIK, поскольку его работа основана на изменении положения суставов в пространстве, а не на учёте углов их поворота.

Стоит отметить, что двухмерные вращающиеся суставы накладывают ограничения на их положения в трёхмерном пространстве, ограничивая их перемещение плоскостями, перпендикулярными осям вращения. Для определения этих ограниченных областей можно использовать локальные системы координат, позволяющие проектировать положения суставов относительно заданных направлений. Однако внедрение таких систем координат и использование матричных преобразований противоречит изначальной концепции алгоритма FABRIK, а также существенно увеличивает вычислительные затраты, особенно при наличии большого количества суставов.

Одним из возможных решений является представление манипулятора как последовательности взаимосвязанных суставов, вращающихся в одной плоскости. В этом случае ориентация рабочего органа может быть определена через последовательное суммирование углов поворота в осях рыскания, тангажа и крена. Подобная конструкция широко распространена в современных манипуляторах, где один сустав отвечает за угол рыскания, а следующие за ним суставы регулируют углы тангажа и крена [6].

Таким образом, для эффективного функционирования нового алгоритма необходимо учитывать ориентацию конечного эффектора, определяя углы поворота через всю цепь промежуточных суставов. При этом важно учитывать ограничения суставов, задавая минимальные и максимальные предельные значения углов, соответствующие их физическим ограничениям.

**Избежание локальных минимумов путем перемещения основания:** одной из ключевых особенностей мобильного манипулятора по сравнению с другими кинематическими цепями при решении задачи обратной кинематики является наличие подвижного основания. Это означает, что в процессе вычислений, если достижение целевого положения становится невозможным из-за ограничений суставов или проблем с доступностью, основание может перемещаться, обеспечивая возможность продолжения решения и нахождения альтернативного результата.

Перемещение основания манипулятора целесообразно только в случае, если после итерации ошибка (отклонение между положением конечного эффектора и целью) не уменьшается. При этом величина смещения определяется в зависимости от этого отклонения.

Дополнительно следует учитывать, что реальные суставы обладают несимметричными угловыми ограничениями. Вследствие этого возможна ситуация, при которой целевая точка остается недостижимой, а решение попадает в локальный минимум из-за предельных значений суставов. Однако, если изначальное положение основания изменить, можно избежать локального минимума и получить глобально оптимальное решение.

Используя возможность изменения положения основания в процессе работы алгоритма, данную проблему можно решить, переместив основание и последующие суставы (до сустава  $m$ , достигшего предела) на противоположную сторону. Это позволяет продолжить выполнение итераций и повысить вероятность успешного нахождения решения.

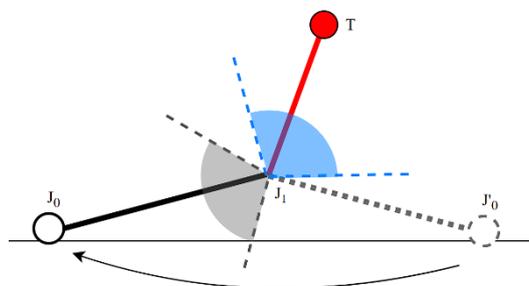


Рис. 1. Пример перемещения основания для избежания локального минимума: для обеспечения целевой ориентации положение красного сустава не может быть изменено, перемещение основания из положения  $J'_0$  в  $J_0$  переустановит сустав  $J_1$  в пределах

**Добавление приоритетного параметра:** для манипуляционных систем с числом степеней свободы, значительно превышающим количество ограничений задачи, оригинальный алгоритм FABRIK продолжает обеспечивать гарантированное решение. Основой работы FABRIK является перемещение и последовательная корректировка положений суставов в пространстве, что позволяет находить оптимальное решение с минимальными смещениями суставов относительно их исходных позиций.

Однако в задаче управления манипулятором могут возникать дополнительные приоритеты, такие как минимизация энергозатрат (например, вращение сустава, расположенного ближе к основанию, обычно требует больше энергии). В связи с этим в новом алгоритме предлагается внедрение дополнительного параметра в процесс регулировки положений суставов. Этот параметр позволяет учитывать приоритеты и направлять процесс решения таким образом, чтобы определенные суставы вращались в большей степени, оптимизируя управление манипулятором в соответствии с заданными критериями.

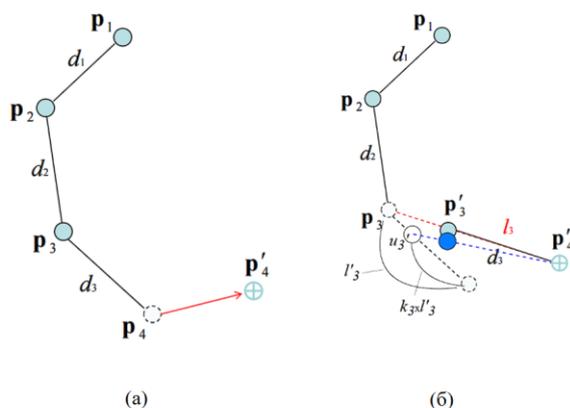


Рис. 2. Работы дополнительного параметра в этапе «прямого прохода» для сустав  $p_3$  :  
 а) перемещать  $p_4$  в  $p_4'$ ,  
 б) новая позиция  $p_3'$  установится в направлении к  $u_3'$ , положение которого определяется на основе предыдущих  $p_4$  и  $p_3$

### Предложенный алгоритм

Функция  $project(p)$  (в линиях 1.11, 1.12 и 1.18) является проекцией точки  $p$  на плоскость, на которой основания движется; функция  $normalize(v)$  (в линиях 1.32, 1.33, 1.41) – нормализация вектор  $v$ ; функция  $rotate(v, d, t_o)$  (в линии 1.39) – вращение вектора  $v$  длиной  $d$  в зависимости от ориентации цели  $t_o$ ; функция  $rotate(v, d, a, o)$  (в линии 1.45) – вращение вектора  $v$  длиной  $d$  на угол  $a$  вокруг оси  $o$ ; функция  $rotate\_angle(v_1, v_2)$  (в линии 1.43) – угол поворота от вектора  $v_2$  к вектору  $v_1$ ; функция  $clamp(a)$  – функция возвращает  $true$ , если угол  $a$  выходит за пределы и удерживает  $a$  в допустимых пределах.

**Предложенный алгоритм управления мобильным манипулятором**

**Вход:** Положения поворота суставов  $\mathbf{p}_i$  для  $i=1, \dots, n$ ; положение и ориентация цели  $\mathbf{t}_p, t_o$ ; расстояния между суставами

$d_i = |\mathbf{p}_{i+1} - \mathbf{p}_i|$ , для  $i=1, \dots, n$ ; приоритетные коэффициенты суставов  $k_i$  для  $i=1, \dots, n-1$ .

**Выход:** Новые положения и углы поворота суставов  $\mathbf{p}_i, a_i$  для  $i=1, \dots, n$ .

```

1.1  % Установить  $\mathbf{b}_p$  в качестве начальных положения сустава  $\mathbf{p}_1$ 
1.2   $\mathbf{b}_p = \mathbf{p}_1$ 
1.3  for  $iter = 1, \dots, iter_{max}$  do
1.4      % Вычислить отклонение между конечным эффектором и целью в положении и в ориентации
1.5       $\mathbf{e}_p = \mathbf{t}_p - \mathbf{p}_n$ 
1.6      if  $|\mathbf{e}_p| < tol_p$  then
1.7          % Сходится
1.8          return true
1.9      % Выполнить смещения базы, если отклонение не уменьшается
1.10     else if  $\mathbf{e}_p^{prev} = \mathbf{e}_p$  then
1.11         if  $project(\mathbf{e}_p) \neq 0$  then
1.12              $s = project(\mathbf{e}_p)$ 
1.13              $\mathbf{b}_p = \mathbf{b}_p + s$ 
1.14             for  $i = 1, \dots, n-1$  do
1.15                  $\mathbf{p}_i = \mathbf{p}_i + s$ 
1.16             end
1.17         else if  $m \neq -1$  then
1.18              $s = project(\mathbf{p}_m - \mathbf{p}_{m-1})$ 
1.19              $\mathbf{b} = \mathbf{b} + s$ 
1.20             for  $i = 1, \dots, m-1$  do
1.21                  $\mathbf{p}_i = \mathbf{p}_i + s$ 
1.22             end
1.23              $m = -1$ 
1.24         else
1.25             % Не сходится
1.26             return false
1.27         end
1.28     end
1.29     % ЭТАП 1: Прямой проход
1.30      $\mathbf{p}_n = \mathbf{t}_p$ 
1.31     for  $i = n-1, \dots, 1$  do
1.32          $\mathbf{u} = \mathbf{p}_{i+1}^{prev} - k_i \cdot d_i \cdot normalize(\mathbf{p}_{i+1}^{prev} - \mathbf{p}_i^{prev})$ 
1.33          $\mathbf{p}_i = \mathbf{p}_{i+1} - d_i \cdot normalize(\mathbf{u} - \mathbf{p}_i)$ 
1.34     end
1.35     % ЭТАП 2: Обратный проход
1.36      $\mathbf{p}_1 = \mathbf{b}_p$ 
1.37     for  $i = 1, \dots, n-1$  do
1.38         if  $i = n-1$  then
1.39              $\mathbf{p}_{i+1} = rotate(\mathbf{p}_i, d_i, t_o)^*$ 
1.40         else
1.41              $\mathbf{p}_{i+1} = \mathbf{p}_i + d_i \cdot normalize(\mathbf{p}_{i+1} - \mathbf{p}_i)$ 
1.42         end
1.43          $a_i = rotate\_angle(\mathbf{p}_{i+1} - \mathbf{p}_i, \mathbf{p}_i - \mathbf{p}_{i-1})$ 
1.44         if  $clamp(a_i)$  then
1.45              $\mathbf{p}_{i+1} = rotate(\mathbf{p}_i, d_i, a_i, \mathbf{p}_i - \mathbf{p}_{i-1})$ 
1.46              $m = i$ 
1.47         end
1.48     end
1.49      $\mathbf{e}_p^{prev} = \mathbf{e}_p$ 
1.50 end
1.51 % Не сходится
1.52 return false

```

Рис. 3. Псевдокод разработанного алгоритма обратного кинематики манипулятором

### Результат тестирования алгоритма

Разработанный алгоритм был протестирован на модели манипулятора, обладающего восемью степенями свободы: тремя степенями свободы платформы и пятью степенями свободы самого манипулятора [7]. Тестирование и моделирование результатов проводилось в среде имитационного моделирования с использованием специального пакета.

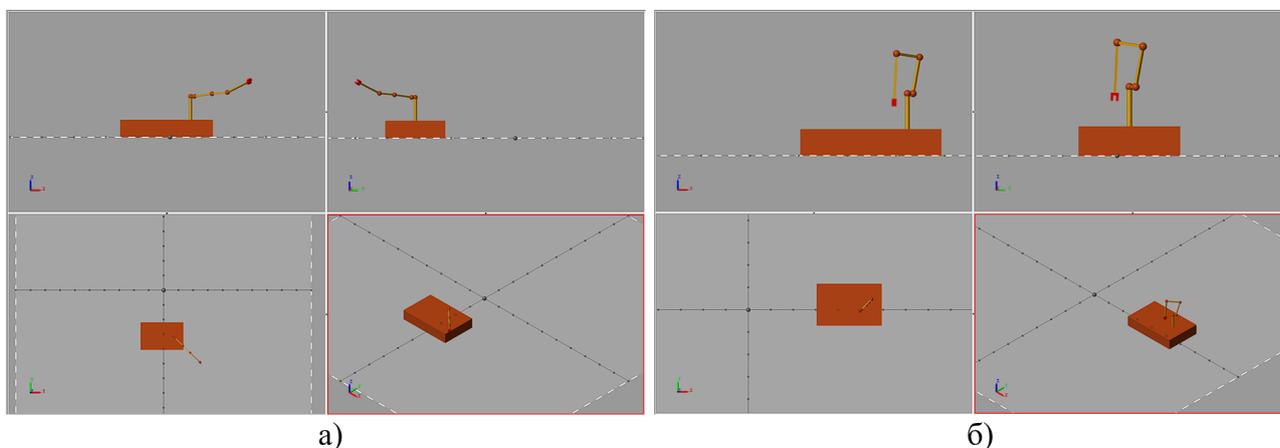


Рис. 4. Поза манипулятора в результате алгоритма:

а) цель  $x = 0,5$ ,  $y = -1$ ,  $z = 0,357$ ,  $roll = 60^\circ$ ,  $pitch = 20^\circ$ ,  $yaw = -45^\circ$

б) цель  $x = 1$ ,  $y = 0$ ,  $z = 0,197$ ,  $roll = 45^\circ$ ,  $pitch = 270^\circ$ ,  $yaw = 45^\circ$

Проведено сравнение скорости решения обратной кинематики, предложенного выше алгоритма и решения обратной кинематики геометрическим методом для модели манипулятора.

Для оценки эффективности предложенного алгоритма проведено сравнение скорости решения задачи обратной кинематики с традиционным геометрическим методом на модели манипулятора. В ходе эксперимента оба алгоритма обрабатывали один и тот же набор из 20 различных целевых положений. Время выполнения измерялось на персональном компьютере с процессором AMD Ryzen 7 7840H (3,8 ГГц). Среднее время обработки геометрическим методом составило 35 мс, тогда как предложенный алгоритм выполнял вычисления в среднем за 142 мс.

### Заключение

В данном исследовании разработан эвристический алгоритм решения задачи обратной кинематики для мобильных манипуляторов. Алгоритм основан на адаптации метода FABRIK с модификацией, учитывающей физические свойства двухмерных вращающихся соединений. Для повышения эффективности добавлены механизмы избегания локальных минимумов и оптимизации энергозатрат за счет приоритетного параметра. Предложенный подход обладает высокой гибкостью благодаря учету подвижности основания, что позволяет продолжать итерации даже при наличии ограничений суставов или недостижимых целей. Экспериментальная проверка на модели показала, что разработанный алгоритм демонстрирует конкурентоспособную производительность по сравнению с геометрическим методом. Хотя его скорость обработки несколько ниже, он обладает высокой универсальностью, позволяя применять его к различным манипуляторным системам без необходимости повторного построения сложных систем уравнений замкнутой формы.

### Список использованной литературы

1. Kalasariya M., Patel V., Thakkar A. Comparative Study of Iterative Inverse Kinematics Methods for Serial Manipulator // International Journal of Engineering Research and Technology. – 2018. – Vol 7. – P. 1-5.
2. Welman C. Inverse kinematics and geometric constraints for articulated figure manipulation // Master Dissertation // Simon Fraser University, Department of Computer Science – 1993.
3. Kwan W. C., von Kinsky B. R., Marriott A. Closed-form and generalized inverse kinematics solutions for the analysis of human motion. – 2015. – Vol 5. – P. 1911-1914.
4. Aristidou A., Lasenby J. Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver // Report number: CUEDF-INFENG, TR-632 // University of Cambridge. – 2009.
5. Aristidou A., Lasenby J. FABRIK: a fast, iterative solver for the inverse kinematics problem // Graphical Models. – 2010. – Vol 73. – P. 243-260.
6. He R., Wang Y., Li T. Design and control of the Six-DOF robotic manipulator // Applied and Computational Engineering. – 2021. – Vol 78. – P. 129-135.
7. Bischoff R., Prassler E., Huggenberger U. KUKA youBot – A Mobile Manipulator for Research and Education // Proc. of the IEEE International Conference on Robotic and Automation. – 2011.