

#### 4.1 РЕШЕНИЕ ОБРАТНОЙ ЗАДАЧИ КИНЕМАТИКИ ДЛЯ ПОСЛЕДОВАТЕЛЬНЫХ МАНИПУЛЯТОРОВ ИЗБЫТОЧНОЙ КИНЕМАТИКИ

Задача организации общего рабочего пространства робот – человек имеет большое практическое значение и находит применение в следующих областях человеческой деятельности:

- Проведение экспериментальных исследований, совместная сборка или изготовление некоторого изделия, при условии невозможности автоматизации всего процесса и исключения человека из него
- Логистика в тех ее проявлениях, где человек не может быть исключён из рабочей области, примером может быть мерчендайзинг и нужды развлекательной индустрии
- Автоматизация клинических исследований, забора анализов, оказания первой медицинской помощи
- Сервисное обслуживание в различных областях человеческой деятельности, например, помощь у ходе за больными и пожилыми людьми, сервис по дому

Основной проблемой обеспечения безопасности при разделении рабочего пространства и организации совместной деятельности людей и роботов является сложность, а порой и невозможность предсказания поведения человека: для кооперации и совместной деятельности люди часто прибегают к невербальным средствам общения – естественным для человека, но тяжело формализуемым и интерпретируемым для робототехнических систем [1].

Постоянный контроль за движениями человека, посредством внешних датчиков (например, оптических или глубинных камер) позволяет успешно избегать столкновений робота и человека в общем рабочем окружении, но постоянное избегание роботом контактов с человеком исключает возможности к той кооперации, которая возникает между людьми.

С другой стороны, поведение робота так же может быть непредсказуемо для человека. Существует огромное множество методов и алгоритмов организации движения сложных робототехнических систем и их возможная неестественность может сбивать человека с толку, отвлекать от основной деятельности, что в свою очередь может мешать роботу выполнять свою часть работ.

Наиболее перспективными коллаборативными типами роботов являются манипуляторы последовательной избыточной кинематики. Движение таких роботов во много определяется способом решения обратной задачи кинематики (ОЗК). Ввиду наличия сингулярных состояний в конфигурационном пространстве робота задача нетривиальна и до сих пор не имеет однозначного решения.

Использование якобиана системы является наиболее распространенным подходом к ее решению. Ввиду вырождения якобиана для сингулярных конфигураций используются различные пути его псевдообращения. К наиболее распространенным методам данного класса относят метод Мора – Пенроуза, метод транспонированного якобиана [2], метод сингулярного разложения матриц, метод затухающих наименьших квадратов и их различные вариации [3]. Данные методы в той или иной мере подвержены проблеме дрожания в сингулярных конфигурациях, а так же, ввиду линейной аппроксимации действительных движений робота, решения могут казаться неестественными.

Также известны так называемые методы Ньютона, основу которых составляет разложение в ряд Тейлора второго порядка следующей целевой функции:

$$f(x + \sigma) = f(x) + [\nabla f(x)]^T \sigma + \frac{1}{2} \sigma^T H_f(x) \sigma,$$

где  $H$  – матрица Гессе. Такой подход позволяет получать гладкие траектории движения, включать ограничения для движения сочленений и не подвержен проблеме дрожания [4]. Методы Ньютона не получили широкого распространения ввиду сложности их реализации и значительных требований к вычислительным ресурсам.

Последние несколько лет набирают популярность алгоритмы, основанные на быстрорастущих деревьях (RRT) и вероятностных дорожных картах [5]. Такие методы обеспечивают высокую скорость исследования пространства

конфигураций и неплохо зарекомендовали себя при решении задач большой размерности, однако, обладают плохой сходимостью, когда области, обеспечивающие связность отдельных компонент пространства конфигураций, имеют малый размер [6].

Проблема генерации естественных движений последние два десятка лет является ключевой в области компьютерной графики и анимации где для решения обратной задачи кинематики наиболее широкое распространение приобрел алгоритм циклического схождения координат (CCD) [7]. Алгоритм легко реализуем программно и вычислительно незатратен – время прохождения одной полной итерации линейно зависит от количества степеней подвижности манипулятора. Алгоритм легко расширяется для учёта ограничений обобщённых координат, но есть сложности при его применении к кинематическим схемам с более чем одним рабочим органом.

В последние несколько лет так же набирает популярность алгоритм FABRIK, представленный в работе [8]. Как и алгоритм циклического схождения координат, FABRIK вычислительно незатратен и прост в реализации, в общем виде способен работать с различными кинематическими схемами (несколько рабочих органов, циклы), а так же позволяет получать плавные, естественные движения. Его основу составляют две итеративные операции. Первая (forward reaching) заключается в следующем: начиная с предпоследнего звена и итеративно опускаясь к первому положение каждого звена выбирается таким, чтобы конец этого звена был направлен на предыдущее. Вторая операция (backward reaching) схожа с первой: начиная с первого звена и итеративно поднимаясь к последнему, положения звеньев задаются так, чтобы они были направлены на следующее звено. На рисунке 1 данный процесс схематично представлен для плоского последовательного манипулятора, где:  $d_i$  – длина  $i$ -го звена;  $p_i$  – исходное положение,  $p_i'$  – положение после выполнения первой операции и  $p_i''$  – положение после выполнения второй операции соответствующих  $i$ -ых сочленений. Работа алгоритма прекращается, когда достигается требуемая точность для положения последнего звена.

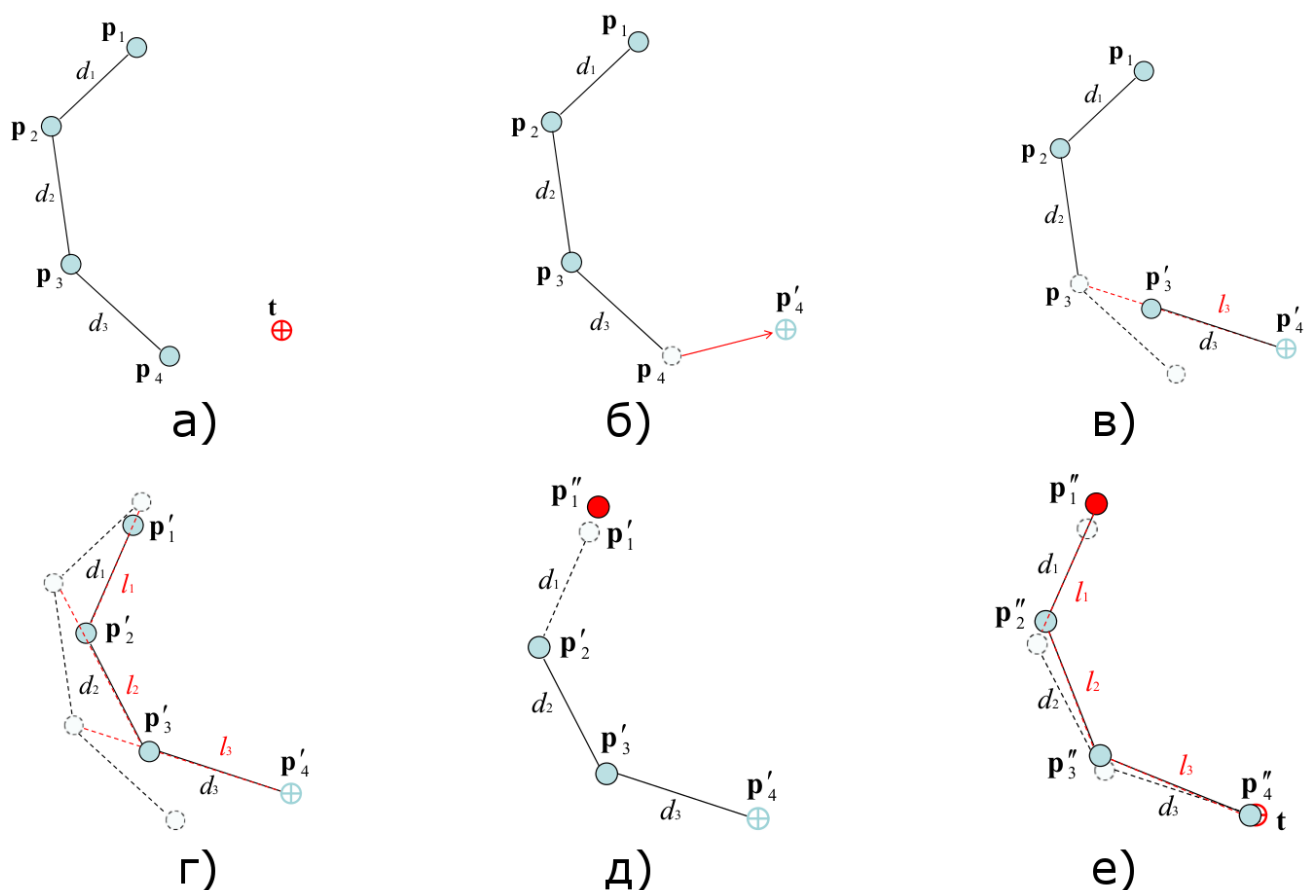


Рисунок 1 – Работа алгоритма FABRIK: а) начальная позиция манипулятора и цели; б) перемещение последнего звена в целевую позицию; в) расположение предпоследнего звена по направлению к выбранному положению последнего звена; г) полностью выполненная первая операция; д) переназначение позиции начального звена; е) полностью выполненная вторая операция

Основным недостатком, ограничивающим применение данного метода в задачах робототехнике является то, что для расчётов используются не значения обобщённых координат, а координаты сочленений в операционном пространстве.

В данном разделе представлен результат адаптации алгоритма FABRIK для решения обратной задачи кинематики с учетом условий и требований, возникающих при работе с робототехническими системами.

Рассмотрим случай последовательного манипулятора избыточной кинематики, чья конфигурация однозначно определяется четырьмя параметрами Денавита-Хантенберга. Тогда соответствующее однородное преобразование из  $O_{i-1}$  в  $O_i$  систему координат представляется в форме:

$${}^{i-1}H_i = {}^{i-1}H_{\theta,i} \cdot {}^{\theta,i}H_{d,i} \cdot {}^{d,i}H_{a,i} \cdot {}^{a,i}H_{\alpha,i}, \quad (1)$$

где  $\theta_i$ ,  $d_i$ ,  $a_i$ ,  $\alpha_i$  – соответствующие параметры Денавита-Хантенберга. Положим, что для всей кинематической цепи задано  $n$  таких однородных преобразований, тогда введем обратные параметры Денавита-Хантенберга следующим образом:

$$\alpha_{i+1,f} = -\alpha_{n-i}, \quad i = 0 \dots n-1, \quad (2)$$

$$a_{i+1,f} = -a_{n-i}, \quad i = 0 \dots n-1, \quad (3)$$

$$d_{i+1,f} = -d_{n-i}, \quad i = 0 \dots n-1, \quad (4)$$

$$\theta_{i+1,f} = -\theta_{n-i}, \quad i = 0 \dots n-1. \quad (5)$$

Тогда, используя введенные обратные параметры Денавита-Хантенберга, введем следующие однородные преобразования из  $O_i$  в  $O_{i-1}$  систему координат как:

$${}^{i,f}H_{i-1,f} = {}^iH_{\alpha,i-1,f} \cdot {}^{\alpha,i-1,f}H_{a,i-1,f} \cdot {}^{a,i-1,f}H_{d,i-1,f} \cdot {}^{d,i-1,f}H_{\theta,i-1,f}. \quad (6)$$

Таким образом, полагая, что параметры  $\theta_i$  являются обобщенными координатами, можно задавать итеративно производить преобразования между различными системами координат от  $O_0$  к  $O_7$  и обратно.

В соответствии с первой операцией, производимой в исходном алгоритме FABRIK, система координат рабочего органа манипулятора  $O_7$  устанавливается в целевую систему координат –  $O_t$ , которая задается соответствующим обобщенным преобразованием  ${}^0H_t$ . Учитывая то, что параметры  $d_{f,i}$ ,  $a_{f,i}$ ,  $\alpha_{f,i}$  определяются конструкцией робота, преобразование из  $O_t = O_{n,f}$  в  $O_{n-1,f}$  в соответствии с (6) однозначно определяется, как  ${}^{i,f}H_{d,i-1,f}$ .

Для задания  $\theta_{i,f}$  выберем такую ближайшую к  $O_{n-1,f}$  систему координат  $O_k$ ,  $k < i$ , чье абсолютное положение будет меняться при изменении  $\theta_{i,f}$ , тогда значение такой обобщенной координаты определяется, как:

$$\theta_{i,f} = \text{atan2}({}^{d,i-1,f}y_k, {}^{d,i-1,f}x_k). \quad (7)$$

Таким образом полностью определяется  ${}^{i,f}H_{i-1,f}$  для всех  $i$ . Вторая операция эквивалентна первой с тем лишь отличием, что для задания  $\theta_i$  выбирается ближайшая система координат из набора получаемых в первой итерации  $O_{k,f}$ ,  $k > i$ .

Для задания целевой ориентации рабочего органа рассматривается матрица вращения  ${}^{n-1,b}R_t$ , из которой  $\theta_n$  определяется как компонента вращения по оси  ${}^{n-1,b}z_t$ .

Предлагаемый алгоритм не подвержен проблемам дрожания в случаях, когда цель недостижима или сингулярных состояниях, позволяет работать с различными кинематическими схемами и позволяет генерировать более естественные движения в сравнении с алгоритмами на основе псевдообращения Якобиана, быстрорастущих деревьев и вероятностных дорожных картах. Так же позволяет легко учитывать ограничения в сочленениях, а сложность вычислений растет линейно, и зависит от количества обобщенных координат.

Рассмотрим предлагаемый алгоритм на примере манипулятора KUKA iiwa 820. Кинематическая схема и системы координат, заданные в соответствии с соглашением Денавита-Хартенберга представлены на рисунке 2.

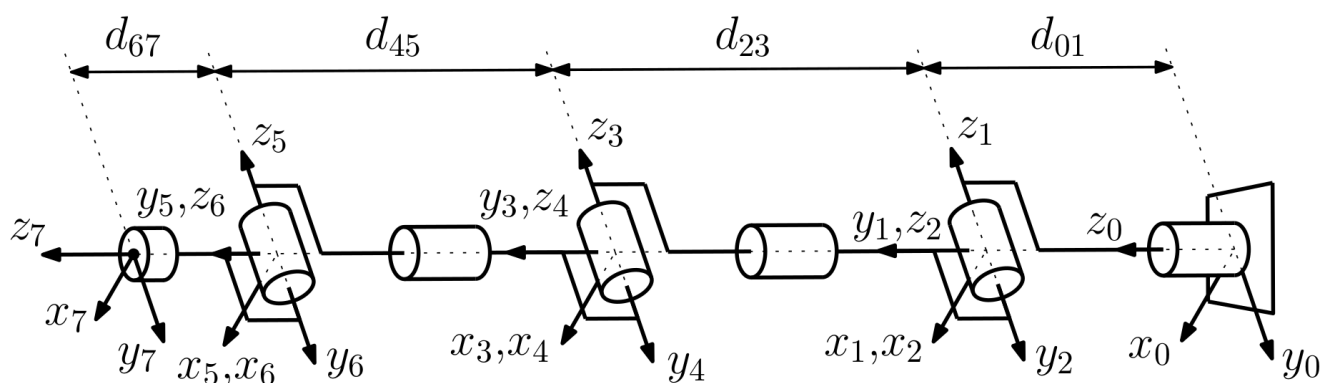


Рисунок 2 – Кинематическая схема манипулятора KUKA iiwa 820

Для моделирования системы использовался пакет MatLab\Simulink. На рисунке 3 представлены результаты. Системами координат  $O_{i,f}$  и  $O_{i,b}$  обозначены  $i$ -ые системы координат манипулятора после проведения первой и второй операций соответственно. На рисунке 3(д), после повторного запуска алгоритма ошибка по абсолютному положению и ориентации составила 0.034[мм] и 0.0027[рад] соответственно.

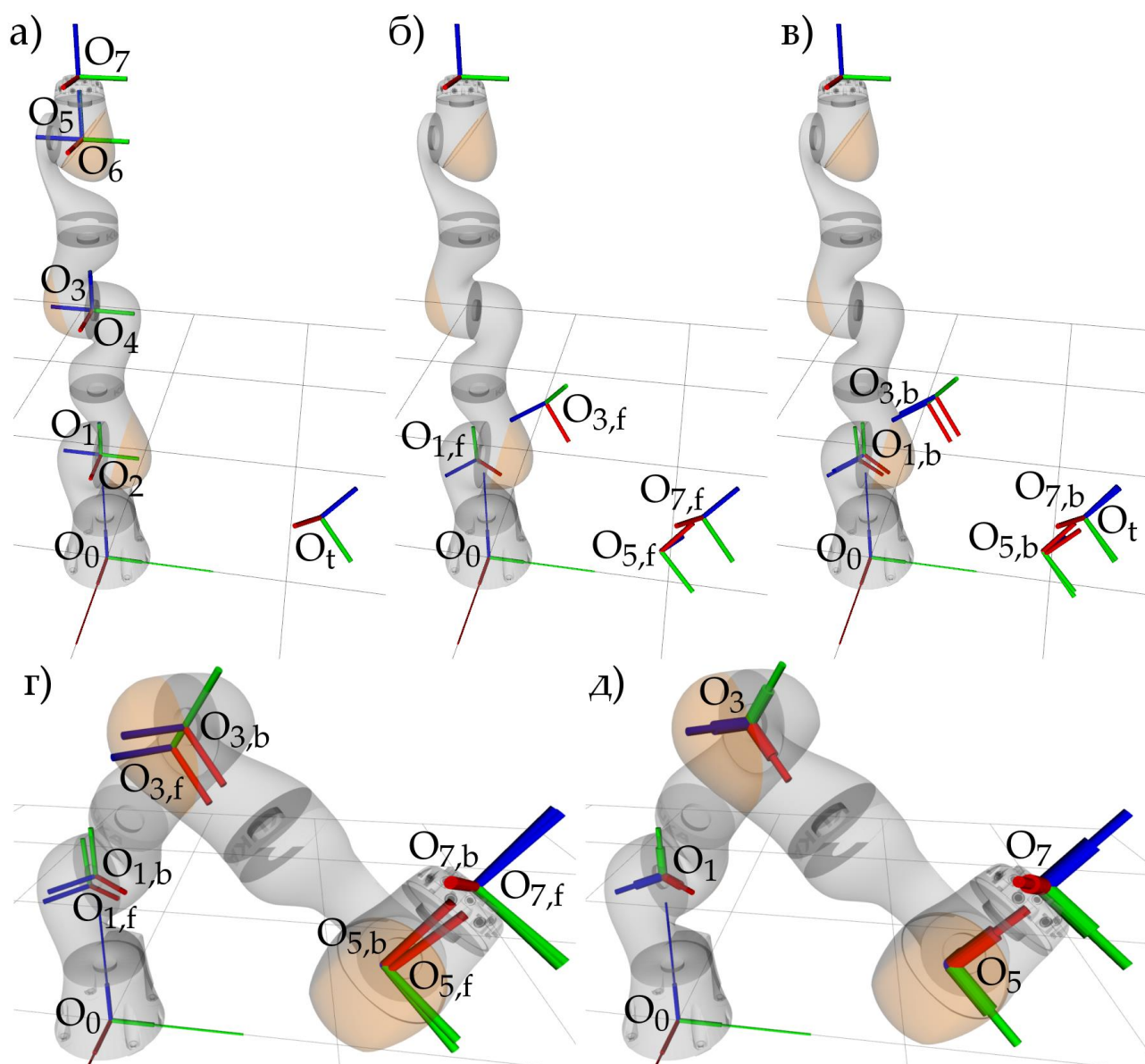


Рисунок 3 – Решение ОЗК предлагаемым алгоритмом: а) робот в начальной конфигурации; б) результат выполнения первой операции; в) результат выполнения второй операции; г) робот в конфигурации заданной результатом второй операции; д) результаты повторного запуска алгоритма

- [1] The Encyclopedia of Human-Computer Interaction / Mads Soegaard, Rikke Friis Dam — The Interaction Design Foundation, 2nd Ed.
- [2] Buss S. R. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods //IEEE Journal of Robotics and Automation. – 2004. – T. 17. – No. 1-19. – C. 16.
- [3] Samuel R. Buss and Jin-Su Kim. Selectively damped least squares for inverse kinematics. *Journal of Graphics Tools*, 10(3):37–49, 2005.
- [4] Jianmin Zhao and Norman I. Badler. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics (TOG)*, 13(4):313–336, 1994.
- [5] L. Janson, E. Schmerling, A. Clark, M. Pavone. Fast marching tree: a fast marching sampling-based method for optimal motion planning in many dimensions. *The International Journal of Robotics Research*, 34(7):883-921, 2015.
- [6] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, New York, NY, USA, 2006.
- [7] Li-Chun Tommy Wang and Chih Cheng Chen. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 7(4):489–499, 1991.
- [8] Andreas Aristidou and Joan Lasenby. FABRIK: a fast, iterative solver for the inverse kinematics problem. Submitted to *Graphical Models*, 2010.



