

УДК 621.865.8

Иерархический подход к решению обратной задачи кинематики

Каргинов Л. А.^{1,*}

[*herstellung@mail.ru](mailto:herstellung@mail.ru)

¹МГТУ им. Н.Э. Баумана, Москва, Россия

В статье изложен иерархический подход к решению обратной задачи кинематики. Идея подхода заключается в следующем: из всех сочленений робота выбирается ограниченный набор сочленений, необходимый для реализации определенного движения (или набора движений). Для этих сочленений назначается строгая последовательность движения и ограничения изменений обобщенных координат. Сочленения, не участвующие в реализации заданного движения, блокируются, для них расчеты не проводятся. Достоинствами подхода являются: однозначность получаемого решения, возможность рассмотрения вырожденных конфигураций исполнительного механизма робота, возможность вычислений в условиях кинематической избыточности, высокая скорость вычислений. Указанные достоинства обусловлены тем, что последовательность движения сочленений изначально устанавливает определенную желаемую конфигурацию исполнительного механизма робота в пространстве и, по сути, задает направление решения. По этой же причине подход назван иерархическим. В статье показаны примеры решения обратной задачи кинематики для манипулятора, закрепленного на стойке и имеющего десять сочленений. При этом было задействовано до восьми сочленений манипулятора, то есть, решение было получено для восьми неизвестных. В ходе двадцати экспериментов с одинаковыми начальными условиями были получены одинаковые результаты решения, удовлетворяющие ограничениям обобщенных координат.

Ключевые слова: обратная задача кинематики, робот, алгоритм, обобщенные координаты, иерархический подход, решение

Введение

Решение обратной задачи кинематики [1] – основа для получения управляющих воздействий для приводов роботов. Но в ряде случаев решение осложнено кинематической избыточностью механизма, а также ограничениями изменений обобщенных координат (в свою очередь обусловленных геометрией рабочего пространства робота и/или конструкцией его исполнительного механизма).

Кинематическая избыточность – неоднозначность решения обратной задачи кинематики, возникающая в том случае, когда количество степеней свободы исполнительного механизма превышает размерность рабочего пространства [2]. Но и для механизмов,

имеющих всего две степени свободы, возможны ситуации, когда обратная задача кинематики допускает несколько решений (рис. 1).

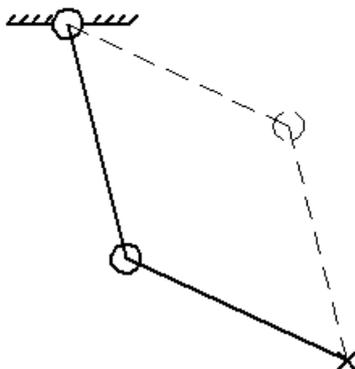


Рис. 1. Два возможных решения для двухзвенного манипулятора

В настоящее время разработано множество методов решения обратной задачи кинематики [1, 3 - 16].

Метод обратных преобразований [1] может не иметь решений или иметь несколько решений и некоторые из них приходится дополнительно отсеивать (например, из-за кинематических ограничений).

Тригонометрический метод [1] может не иметь решений при числе степеней подвижности, превышающем количество получаемых выражений относительно обобщенных координат.

Методы использования матрицы Якоби [1, 3, 4, 5], методы Ньютона [3], методы нелинейного программирования [6] и метод интервалов [7] при большом числе степеней подвижности требуют слишком много времени на вычисления, что затрудняет их использование в режиме реального времени. Кроме того, метод интервалов разработан для простых кинематических цепей.

Методы Sequential Monte Carlo Method (SMCM) [8], Style-based Inverse Kinematics [9] и Mesh-based Inverse Kinematics [10] требуют обязательного предварительного обучения модели, что недопустимо в условиях непредсказуемости условий окружающей среды.

Метод SIK [11] предназначен для моделирования движений человека, для других систем придется переписывать алгоритмы.

Методы CCD [12] и Triangulation Inverse Kinematics [13] разработаны для простых кинематических цепей. При наличии нескольких конечностей возможно применение этих методов путем расчленения механизма на несколько простых цепей и решения обратной задачи кинематики отдельно для каждой цепи, однако при этом не учитывается взаимовлияние частей механизма, придется дополнительно проводить согласование получаемых решений.

Метод FTL [14] позволяет учитывать взаимовлияние приводов звеньев друг на друга. Однако он не был проработан на случай нескольких концов кинематической цепи (то есть, для нескольких схватов или ног).

Метод FABRIK – быстрееший из методов решения обратной задачи кинематики и, по заявлениям авторов, не имеющий указанных выше недостатков. Метод позволяет вести расчеты для кинематических схем с несколькими ветвями, учитывать ограничения в сочленениях. Методика учета ограничений в сочленениях, используемая в FABRIK, разработана для вращательных сочленений, имеющих три степени свободы. Ограничения при этом рассматриваются как параболические или эллиптические конусы, вершина которых находится в рассматриваемом сочленении. В процессе применения ограничений необходимы дополнительные вычисления, включающие в себя использования уравнений параболы, эллипса или окружности (зависит от формы конуса).

Однако, по заявлениям авторов, методика учета ограничений не может быть напрямую применена, например, к поступательным сочленениям, так как подразумевает постоянное расстояние между сочленениями. В [15] указано, что для получения решения необходима дополнительная информация об ограничениях и адаптация методики.

В данной статье будет показана возможность однозначного решения обратной задачи кинематики для многозвенного исполнительного механизма с использованием иерархического подхода, предложенного автором.

1. Иерархический подход

Иерархический подход был изначально разработан для решения обратной задачи кинематики для конечностей исполнительных механизмов шагающих роботов [17, 18, 19]. При этом любая точка опорной поверхности считалась пригодной для постановки в нее стопы робота, т.е. положение целевой точки не задавалось.

Распространим подход, изложенный в [17, 18, 19], на решение обратной задачи кинематики в постановке, приведенной в [1]: по заданному положению и ориентации схвата найти значения обобщенных координат исполнительного механизма.

Основная идея подхода состоит в том, чтобы положить в основу решения некоторое движение (или набор движений), которое механизм должен реализовывать на практике.

Пусть исполнительный механизм робота представляет собой разомкнутую кинематическую цепь, имеющую N сочленений.

Для реализации решения обратной задачи кинематики необходимо:

1. Выбрать движение (или набор движений), которое будет положено в основу решения.
2. Выбрать сочленения, которые будут задействованы в реализации заданного движения. Пусть за реализацию движения отвечают M сочленений, при этом $M < N$ (если для реализации движения необходимы все сочленения, то $M = N$).

Если $M < N$, то оставшиеся $(N - M)$ сочленений будут неподвижными, в алгоритме решения их обобщенные координаты учитываться не будут, значения этих координат будут постоянными.

Если задан набор движений, то для каждого движения выбираем M_i сочленений, i – номер движения.

3. Назначить последовательность движения выбранных M сочленений в процессе реализации заданного движения.
4. Назначить ограничения изменений обобщенных координат в выбранных сочленениях.

Пусть теперь целевые положение и ориентация схвата в системе координат рабочего пространства представлены матрицей преобразования координат T_N [1]:

$$T_N = T_0 \cdot A_1 \cdot A_2 \cdot \dots \cdot A_i \cdot \dots \cdot A_N \quad (1)$$

Начальное положение исполнительного механизма определяется начальными значениями обобщенных координат q_1, q_2, \dots, q_N .

Решение обратной задачи кинематики заключается в постепенном приближении к заданному положению и ориентации схвата путем изменения (увеличения или уменьшения) значений обобщенных координат в той же последовательности, в которой должны двигаться выбранные сочленения.

Процесс решения является циклическим. На каждом шаге цикла обобщенные координаты выбранных M сочленений перебираются в указанной выше последовательности, над каждой координатой производится ряд действий, описанных ниже.

Каждая обобщенная координата изменяется в соответствии с формулой:

$$q_i = q_{i-1} + \text{sign}(q_i) \cdot \Delta q_i \quad (2)$$

где q_{i-1} и q_i – значения обобщенной координаты на предыдущей и текущей итерациях соответственно, $\text{sign}(q_i)$ – знак приращения обобщенной координаты, Δq_i – величина приращения обобщенной координаты.

Затем для этой обобщенной координаты производятся следующие проверки:

- проверка влияния обобщенной координаты на расстояние от схвата до цели и/или на ориентацию схвата;
- проверка достижения обобщенной координатой пределов ее изменения, установленных разработчиком.

В зависимости от результатов проверок предпринимаются следующие действия:

1. Если расстояние до цели уменьшилось и/или ориентация схвата стала ближе к желаемой, то дополнительные корректировки не нужны, переходим к следующей обобщенной координате.

2. Если изменение обобщенной координаты привело к удалению от цели или увеличению углового отклонения от целевого положения, **но ее значение еще не достигло пределов изменения**, знак приращения этой координаты изменяется на противоположный.
3. Если значение обобщенной координаты достигло пределов изменения, то соответствующее ей приращение также меняет свой знак. Если **после изменения знака приращения** схват по-прежнему удаляется от цели, то происходит отмена последнего изменения обобщенной координаты.

Примечание: прием с изменением знака приращения обобщенной координаты позволяет избежать ситуации, когда исполнительный механизм, достигший пределов по нескольким координатам, не в состоянии достичь цели.

В соответствии с идеей иерархического подхода (**положить в основу решения некоторое движение**) каждому из M сочленений, отвечающих за решение, разработчик назначает определенную «роль» в реализации заданного движения. Например, одно сочленение может отвечать за поворот к цели, два других – за сближение с целью, сочленения, находящиеся ближе к схвату – за ориентацию схвата и т.д. Некоторые сочленения могут участвовать как в процессе сближении с целью, так и в установке требуемой ориентации.

В алгоритме решения подобные «роли» сочленений проявляются на этапе проверки соответствующих обобщенных координат:

- если сочленение отвечает только за сближение с целью – для него проверяют изменение расстояния от схвата до цели;
- если сочленение отвечает за ориентацию схвата – для него проверяют угловые отклонения ортов системы координат схвата от осей системы координат рабочего пространства или других осей, выбранных разработчиком;
- если сочленение отвечает за положение и ориентацию – проводятся оба вида проверок.

Если в основу решения положен набор движений, то процесс решения будет состоять из нескольких циклов, выполняемых последовательно (от одного движения к другому).

Решение завершается, когда требуемые по условию положение и ориентация схвата достигнуты с заданной точностью.

Фактически предлагаемый подход имеет бесчисленное множество примеров в природе. Одним из них может служить рука человека: для того чтобы взять что-то со стола человек последовательно задействует определенные сочленения руки. При этом «приводы» плеча в большей степени отвечают за наведение на цель, «приводы» локтевого сустава – за сближение с целью, «приводы» кисти – за ориентацию «схвата» и взаимодействие с целью. Так как при реализации движений человек использует все сочленения руки, то блокировка отдельных сочленений не требуется (случай, когда $M = N$).

2. Объект исследования

В качестве объекта исследования выберем манипулятор, закрепленный на стойке и имеющий 10 сочленений. Примем длины звеньев манипулятора равными:

$L_1 = 0,5$ м, $L_2 = 0,5$ м, $L_3 = 0,5$ м, $L_4 = 0,3$ м, $L_5 = 0,3$ м, $L_6 = 0,3$ м, $L_7 = 0,2$ м, $L_8 = 0,2$ м, $L_9 = 0,2$ м, $L_{10} = 0,2$ м. Длины звеньев выбраны произвольно и приведены здесь для того чтобы читатель мог представить себе геометрию рабочего пространства.

Кинематическая схема манипулятора представлена на рис. 2.

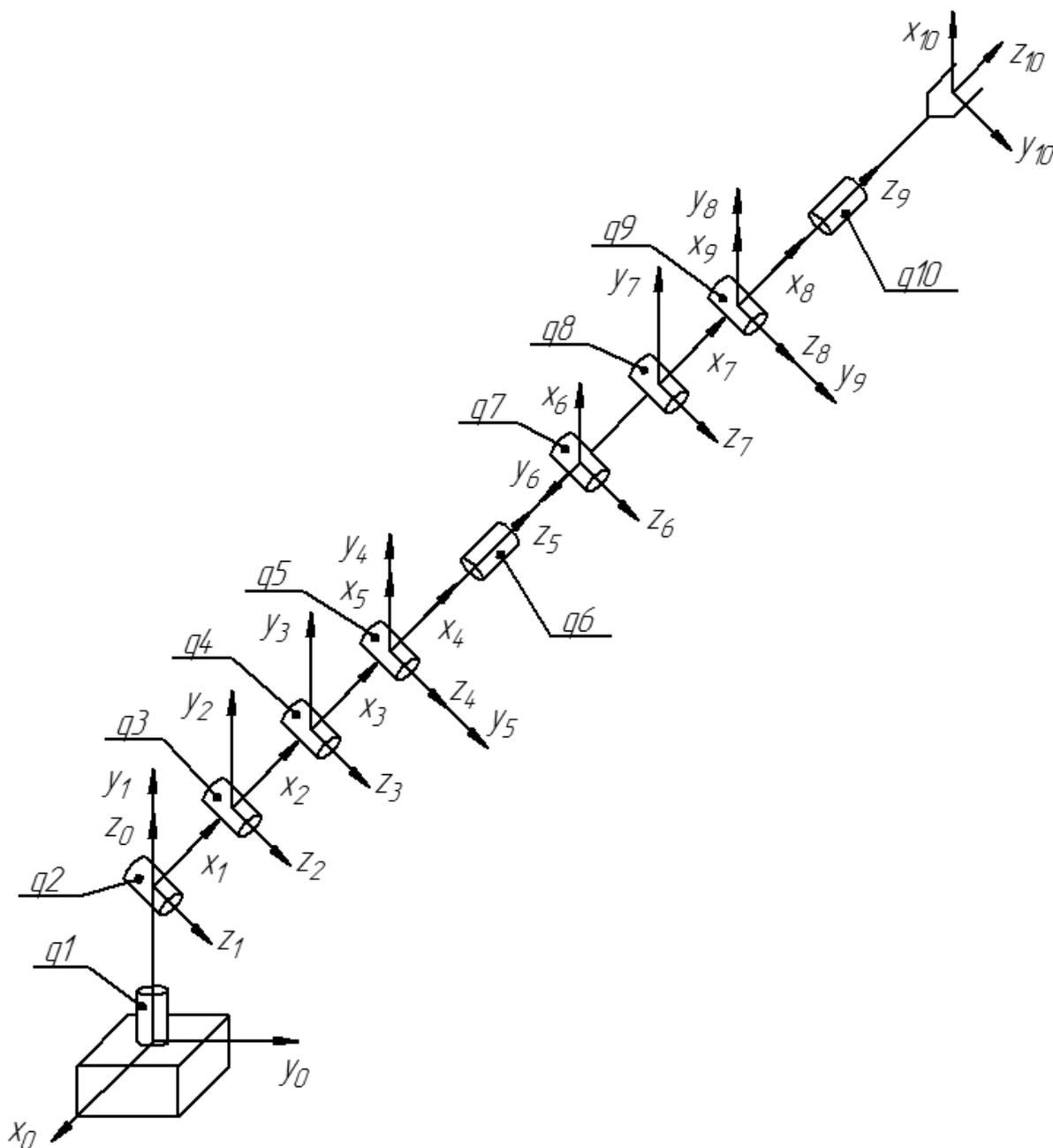


Рис. 2. 10-звенный манипулятор

Для определенности примем все сочленения вращательными.

3. Подготовка к решению

Пусть манипулятор занимает положение, представленное на рис. 3.

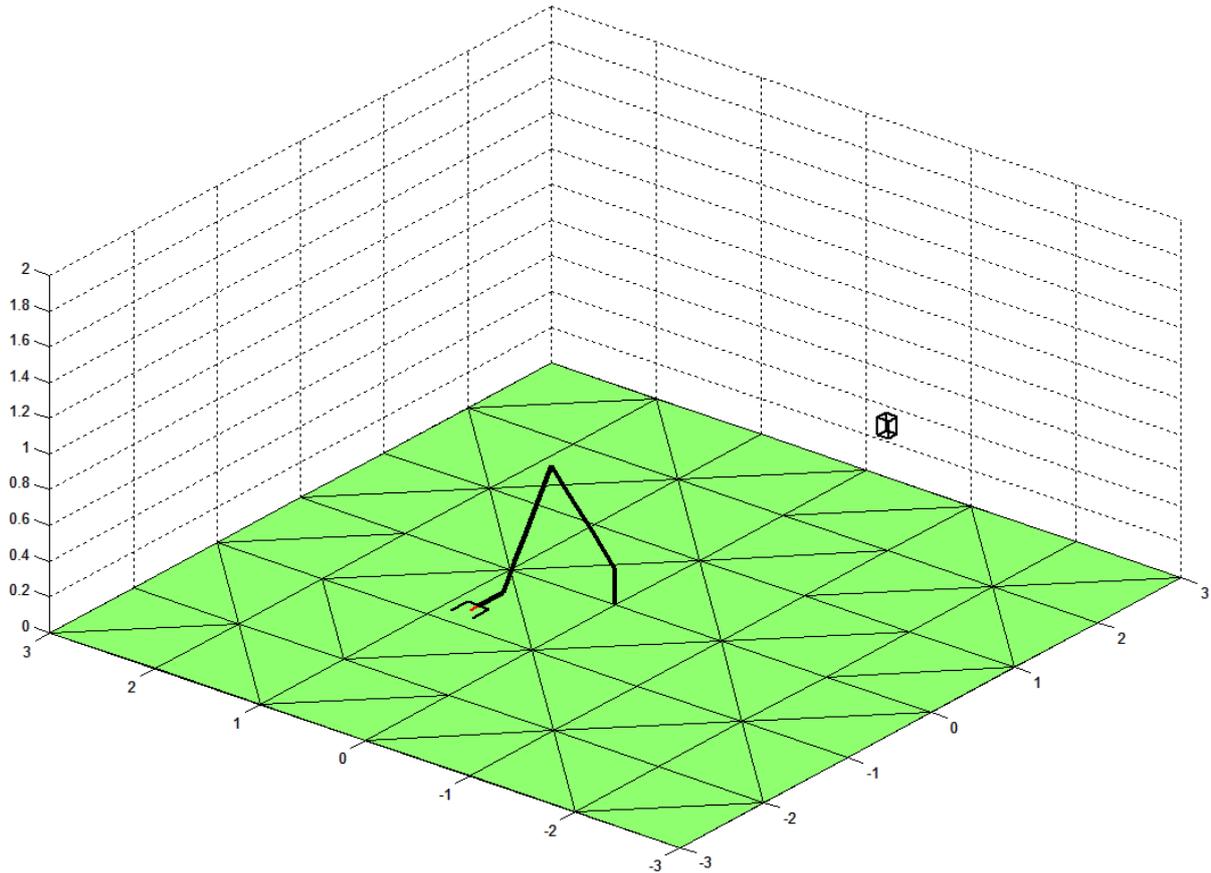


Рис. 3. Начальное положение манипулятора и цели

Параллелепипед «С» представляет собой цель, которой должен коснуться схват манипулятора для завершения процесса решения. При этом схват должен находиться параллельно плоскости xOy системы координат рабочего пространства. Начальным положению и ориентации схвата соответствует матрица $T_{старт}$, конечным – матрица $T_{финиш}$:

$$T_{старт} = \begin{bmatrix} 0 & 0 & -1 & -1,7435 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0,4121 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$T_{\text{финиш}} = \begin{bmatrix} 0 & 0 & 1 & 2,12 \\ 0 & -1 & 0 & -0,9 \\ 1 & 0 & 0 & 0,65 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Таким образом, заданы целевые положение и ориентация схвата манипулятора.

3.1. Пусть **нумерация сочленений** соответствует рис. 2.

3.2. Выберем необходимые для движения сочленения.

Пусть это будут все сочленения манипулятора, кроме q7. Сочленение q1 отвечает за поворот манипулятора относительно стойки на первом этапе решения, сочленения q2, q3, q4, q5, q6, q8 и q9 - за сближение с целью на втором этапе решения, сочленения q5, q9 и q10 – за достижение требуемой ориентации схвата на последнем этапе решения.

Таким образом, имеем 9 сочленений, для которых необходимо решить обратную задачу кинематики.

3.3. Последовательность движения сочленений.

3.3.1. Поворот манипулятора сочленением q1 к цели. Поворот осуществляется до тех пор, пока схват не окажется к цели так близко как возможно.

3.3.2. Сближение с целью с помощью сочленений q2, q3, q4, q5, q6, q8 и q9. При этом порядок изменения обобщенных координат будет следующим:

$$q2 \rightarrow q3 \rightarrow q4 \rightarrow q5 \rightarrow q6 \rightarrow q8 \rightarrow q9$$

3.3.3. Достижение требуемой ориентации схвата сочленениями q5, q9 и q10. При этом порядок изменения обобщенных координат будет следующим:

$$q9 \rightarrow q10 \rightarrow q5$$

Примечание: перебирать обобщенные координаты можно в любом порядке, не обязательно по возрастанию или убыванию номеров сочленений.

3.4. Назначим ограничения изменений обобщенных координат. Ограничения представлены в табл. 1.

Таблица 1. Ограничения значений обобщенных координат

Сочленение	Значение обобщенной координаты, рад		
	Минимальное	Начальное	Максимальное
q1	0	0	2π
q2	-π/2	π/4	π/2
q3	-π/2	0	π/2
q4	-π/2	0	π/2
q5	-π/2	-π/2	π/2
q6	-π	0	π
q8	-π/3	0	π/3
q9	-π/4	π/4	π/4
q10	0	0	2π

Также в табл. 1 приведены начальные значения обобщенных координат. Параметры обобщенной координаты q7 в таблице не представлены, так как она не участвуют в решении обратной задачи кинематики.

4. Некоторые аспекты программной реализации решения

4.1. Дополнительные данные. В процессе решения необходимы данные о положении и ориентации схвата, поэтому необходимо постоянно решать прямую задачу кинематики.

Для этого помимо данных, представленных в пунктах 3.1 – 3.4, в процессе решения необходимы данные о геометрии механизма, т.е. о длинах звеньев и их взаимном расположении. В программе моделирования эти данные представлены в виде параметров Дена-вита-Хартенберга [1], определенных по схеме на рис. 2. Однако разработчики могут использовать системы координат, построенные по другим правилам.

4.2. Коррекция обобщенных координат. На первых двух этапах решения коррекция обобщенных координат проводится на основе расстояния от схвата до цели.

На третьем этапе коррекция обобщенных координат q9 и q10 проводится на основе отклонения оси X10 системы координат схвата от оси Z0 системы координат рабочего пространства, коррекция q5 – на основе расстояния от схвата до цели.

4.3. Выбор приращений обобщенных координат. Приращения для всех обобщенных координат вычислены по следующей формуле:

$$\Delta q_i = A_i / n \quad (4)$$

где A_i – величины диапазонов изменения обобщенных координат, i – номер сочленения. Диапазоны определяются ограничениями, представленными в табл. 1, $n = 100$ для всех сочленений.

Разработчики могут выбирать коэффициент n по своему усмотрению. Увеличение n повышает точность, но снижает скорость решения, т.к. обобщенные координаты изменяются медленнее. Также разработчики вместо формулы (4) могут использовать свой способ расчета приращений.

5. Результаты решения обратной задачи кинематики

На рис. 4 - 6 представлены несколько положений исполнительного механизма манипулятора в процессе решения.

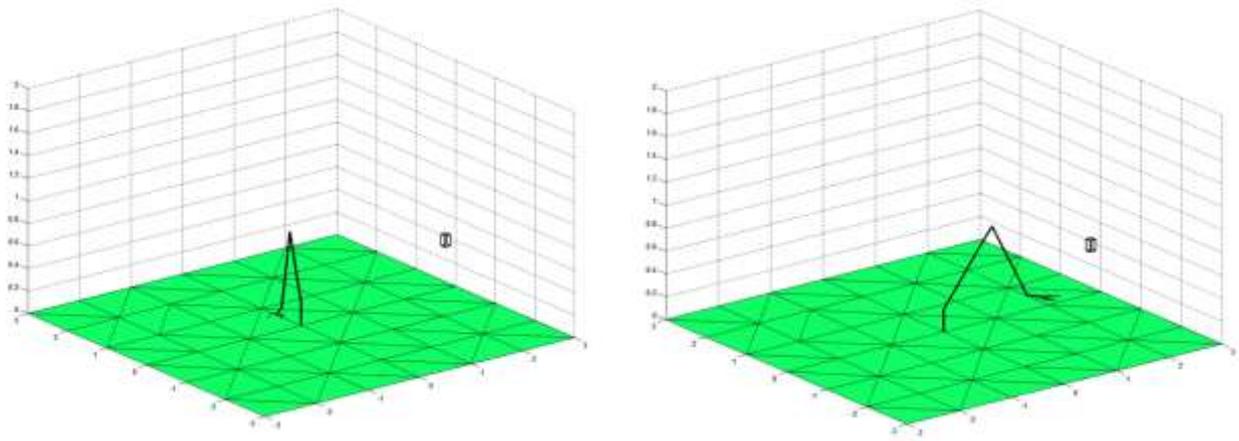


Рис. 4. Поворот манипулятора к цели

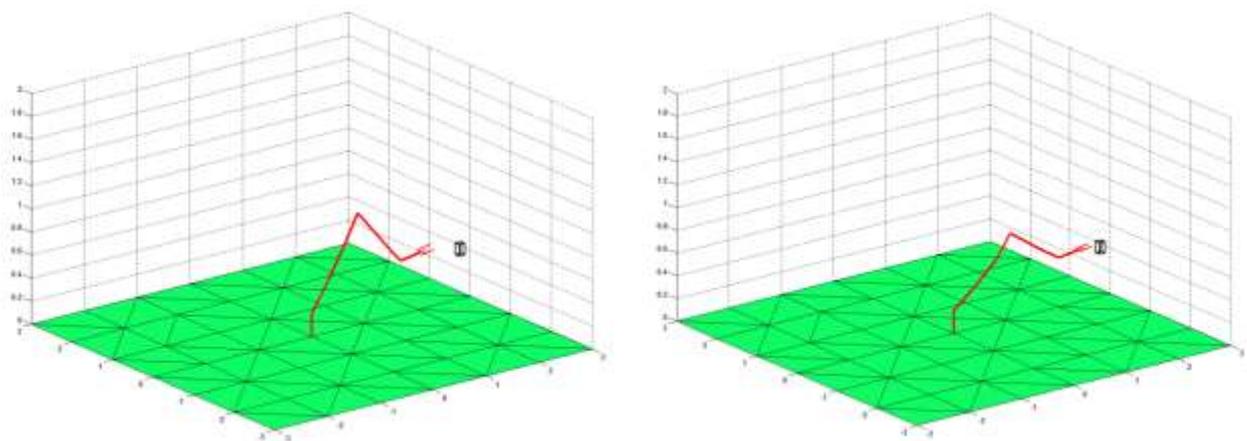


Рис. 5. Сближение с целью

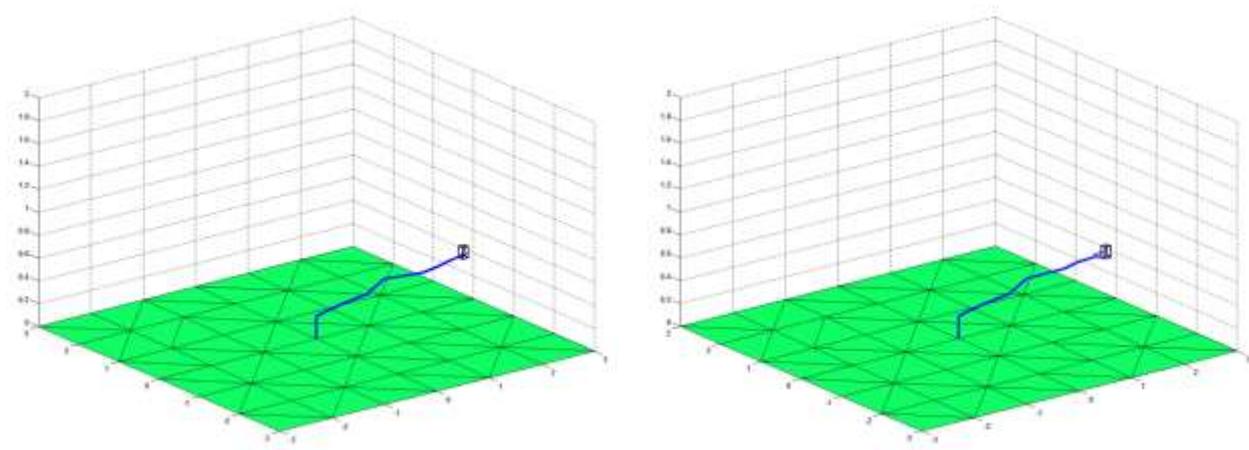


Рис. 6. Достижение требуемой ориентации схвата

Положение манипулятора в момент завершения решения представлено на рис. 7 - 9.

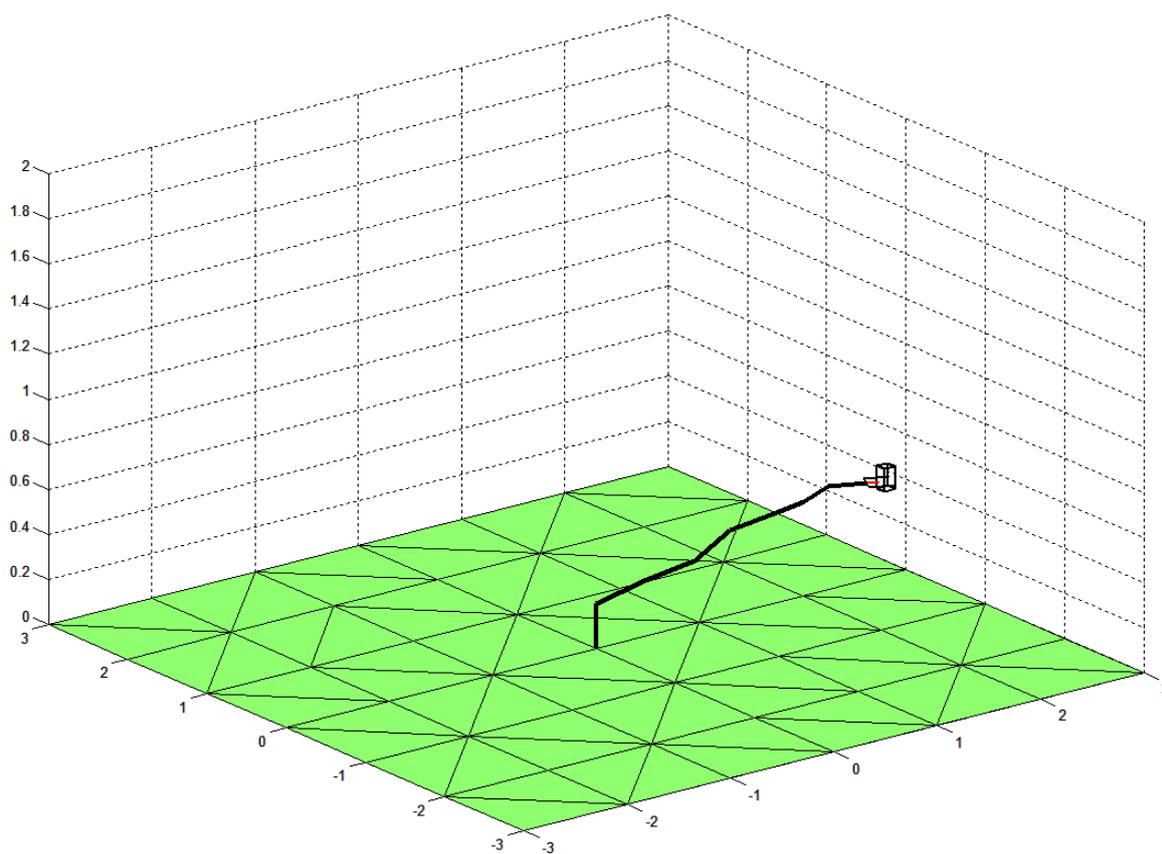


Рис. 7. Момент завершения решения (общий вид)

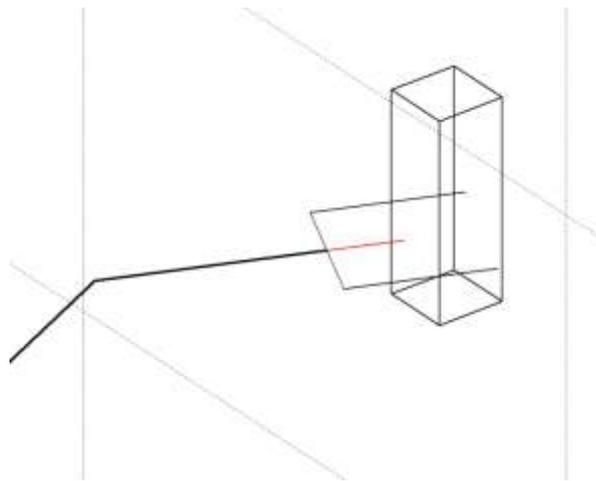


Рис. 8. Схват и цель

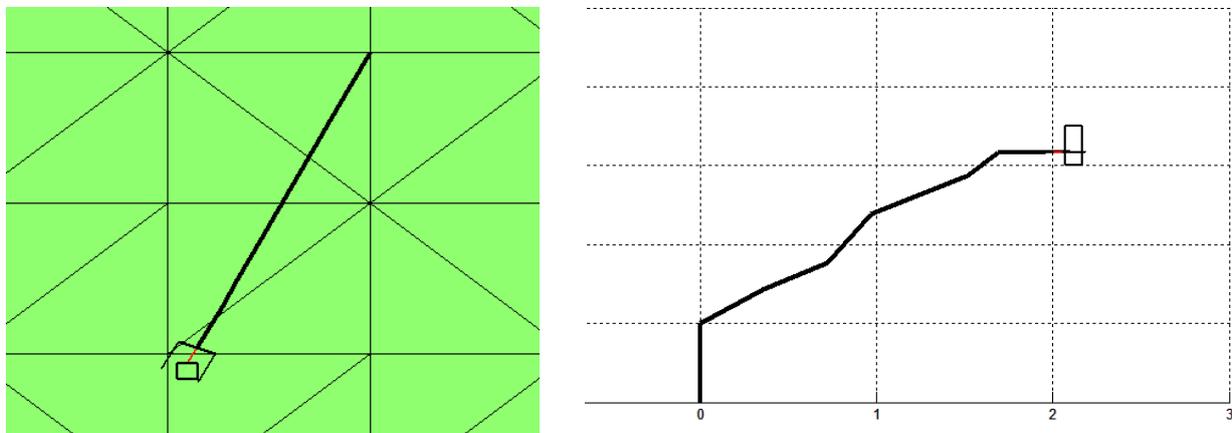


Рис. 9. Манипулятор: слева – вид сверху, справа – вид сбоку

Из представленных рисунков видно, что манипулятор успешно достиг цели. Представленному на рисунке положению манипулятора соответствует следующая матрица положения и ориентации:

$$T_N = \begin{bmatrix} -0,0070 & -0,4291 & 0,9032 & 2,0545 \\ 0,0071 & -0,9032 & -0,4291 & -0,9012 \\ 1 & 0,0034 & 0,0093 & 0,6384 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Первый столбец матрицы показывает, что ось X10 системы координат близка по направлению к оси Z0 системы координат рабочего пространства, что и требовалось по условию задачи.

Отклонение оси X10 от Z0 составило 34°, расстояние от центра цели до начала системы координат схвата составило 0,0665 м.

Значения обобщенных координат, полученные в процессе решения, а также начальные значения, соответствующие табл. 1, приведены в табл. 2.

Таблица 2. Значения обобщенных координат манипулятора

Сочленение	Значение обобщенной координаты, рад		
	До решения	После решения	Приращение Δq
q1	0	2,7332	2,7332
q2	0,7854	0,2749	-0,5105
q3	0	-0,0785	-0,0785
q4	0	0,1855	0,1855
q5	-1,5708	-0,2258	1,3450
q6	0	-0,2356	-0,2356
q8	0	0,0628	0,0628
q9	0,7854	-0,2136	-0,9990
q10	0	0,2293	0,2293

Примечание: в табл. 2 значения обобщенных координат выражены числами, в табл. 1 эти значения выражены через константу π .

Из табл. 2 видно, что все значения обобщенных координат лежат внутри ограниченных, приведенных в табл. 1.

В табл. 3 представлены затраты времени на решение обратной задачи кинематики при моделировании в пакете Matlab. Определение времени производилось с помощью встроенного профайлера пакета.

Таблица 3. Скорость решения обратной задачи кинематики

№ эксперимента	Время, затраченное на различные процессы в ходе решения, с		Полное время решения, с
	Собственное время работы алгоритма решения	Решение прямой задачи кинематики для схвата	
1	0,048	0,186	0,234
2	0,049	0,190	0,239
3	0,048	0,188	0,236
4	0,049	0,191	0,240
5	0,046	0,183	0,230
6	0,050	0,204	0,254
7	0,045	0,185	0,230
8	0,047	0,195	0,242
9	0,045	0,186	0,231
10	0,049	0,197	0,246
11	0,047	0,193	0,240
12	0,049	0,197	0,247
13	0,049	0,194	0,243
14	0,048	0,188	0,235
15	0,045	0,182	0,226
16	0,047	0,192	0,239
17	0,048	0,190	0,238
18	0,054	0,201	0,255
19	0,049	0,192	0,241
20	0,046	0,186	0,232

Из данных табл. 3 видно, что время, затраченное непосредственно на решение обратной задачи кинематики, измеряется сотыми долями секунды.

Время, затраченное на решение прямой задачи кинематики для схвата, значительно больше. Это время зависит от программной реализации функции решения прямой задачи и от частоты вызова этой функции. В каждом из экспериментов функция вызывалась 4758 раз.

6. Решение другим методом

Для обоснования преимуществ иерархического подхода проведем решение обратной задачи кинематики другим методом.

Выберем метод решения, основанный на использовании обратной матрицы Якоби. Так как число сочленений манипулятора равно 10, то матрица Якоби будет иметь размерность 10×6 . Следовательно, обращение такой матрицы невозможно. Поэтому в процессе решения прибегнем к псевдообращению Мура-Пенроуза [1].

Решение обратной задачи кинематики будем искать на основе соотношения:

$$\Delta q = J^+ \cdot \Delta s \quad (6)$$

где Δq – вектор приращений обобщенных координат, Δs – вектор приращения положения и ориентации схвата, J^+ – псевдообратная матрица Якоби, получаемая по формуле (7):

$$J^+ = J^T \cdot (J \cdot J^T)^{-1} \quad (7)$$

где J – матрица Якоби исполнительного механизма.

Стоит отметить, что формула (6) применима лишь при малых величинах компонент вектора Δs , т.е. при малых перемещениях схвата. Поэтому для применения метода с использованием псевдообратной матрицы Якоби необходимо знать величины приращений Δs на каждом шаге вычислений. Кроме того, на каждом шаге необходимо заново пересчитывать матрицу J^+ .

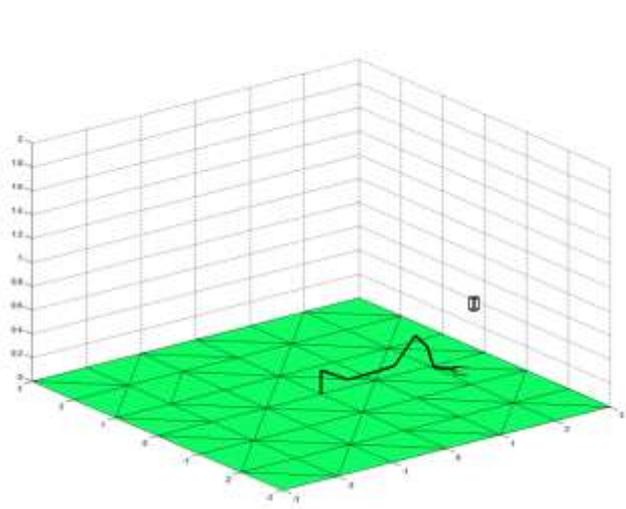
Чтобы определить Δs на каждом шаге необходимо, в общем случае, знать траекторию движения схвата и его ориентацию в каждой точке этой траектории.

Для реализации решения примем траекторию движения от исходного положения схвата до целевого прямолинейной. Начальные и конечные положения схвата определим из матриц $T_{\text{старт}}$ и $T_{\text{финиш}}$, приведенных выше.

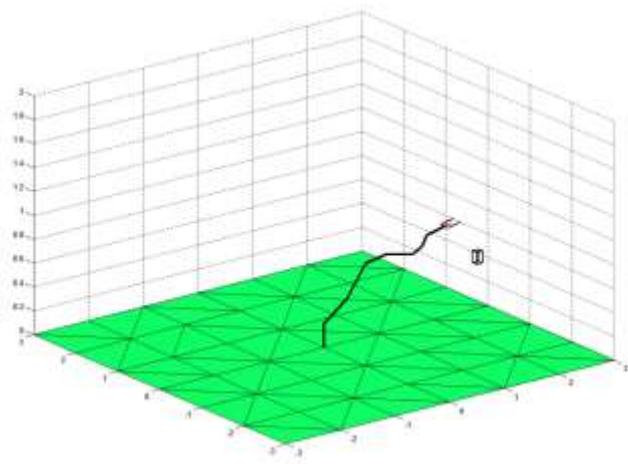
Затем с помощью функции **linspace** (встроенной функции пакета Matlab) получим набор точек, лежащих на прямой, соединяющей начальное и конечное положения схвата.

Таким же образом определим значения углов ориентации схвата в пространстве в каждой точке траектории движения.

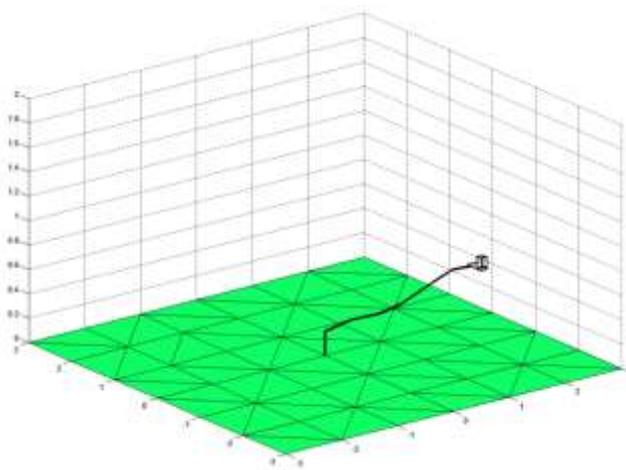
Результаты решения представлены на рис. 10.



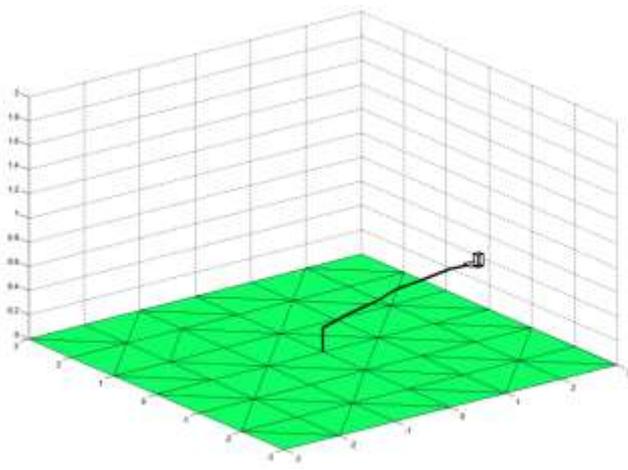
а)



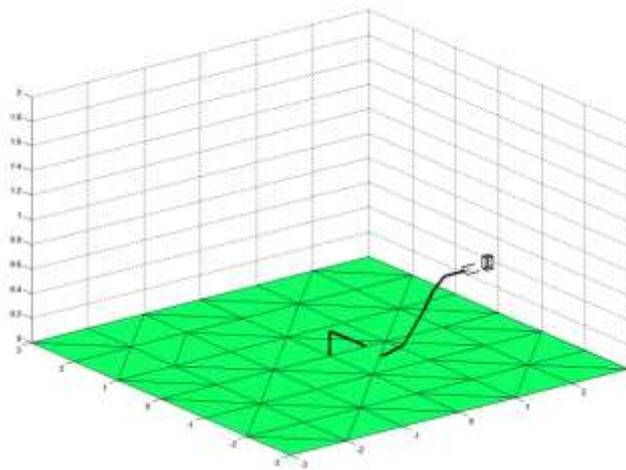
б)



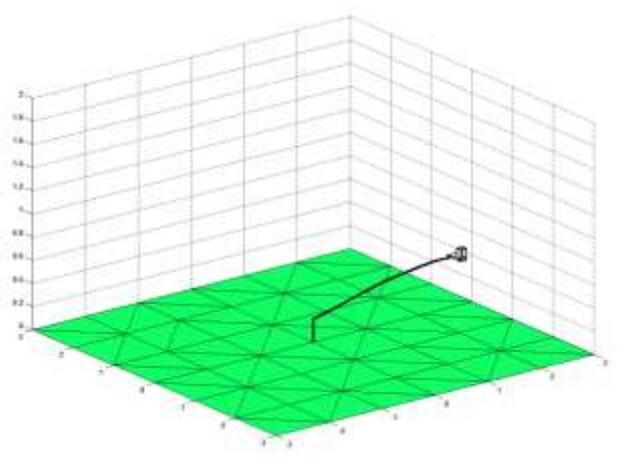
в)



г)



д)



е)

Рис. 10. Результаты решения для разного числа итераций: а) $n = 100$, б) $n = 200$, в) $n = 400$, г) $n = 1100$, д) $n = 1200$, е) $n = 2000$.

Из рисунка видно, что точность решения метода с использованием псевдообратной матрицы Якоби зависит от количества точек на траектории движения схвата. При небольшом числе точек (рис. а) и б)) схват не достиг цели, что объясняется тем, что величины приращений Δs были слишком велики.

Наиболее точное решение удалось получить при $n = 1100$. Матрица преобразования координат, соответствующая схвату, имеет вид (8):

$$T_N = \begin{bmatrix} -0,0142 & 0,0682 & 0,9976 & 2,0385 \\ 0,0029 & 0,9977 & -0,0682 & -0,8844 \\ -0,9999 & 0,0019 & -0,0144 & 0,6204 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Решения для $n = 1200$ и $n = 2000$ будут рассмотрены ниже в разделе «Обсуждение».

В табл. 4 приведены значения обобщенных координат и их приращения, полученные в результате решения.

Таблица 4. Значения обобщенных координат манипулятора

Сочленение	Значение обобщенной координаты, рад		
	До решения	После решения	Приращение Δq
q1	0	2,662546	2,662546
q2	0,7854	0,251219	-0,534181
q3	0	0,022366	0,022366
q4	0	-0,00965	-0,00965
q5	-1,5708	-0,07002	1,50078
q6	0	1,185966	1,185966
q7	0	0,00527	0,00527
q8	0	-0,00577	-0,00577
q9	0,7854	5,798383	5,012983
q10	0	1,999774	1,999774

В таблице 5 приведены данные быстродействия метода в зависимости от числа итераций.

Таблица 5. Быстродействие метода в зависимости от числа итераций n

Число итераций n	Время решения, с
100	0,065
200	0,128
400	0,265
1100	0,692
1200	0,789
2000	1,243

Отклонение оси X10 от Z0 составило $179^{\circ}10'10,6''$, расстояние от центра цели до начала системы координат схвата составило 0,0881 м. Таким образом, схват повернут относительно заданного положения почти на 180° .

Из приведенных данных также видно, что время, необходимое для получения достаточно точного решения, составляет 0,692 с.

7. Результаты решения для другого набора сочленений

Чтобы показать еще несколько преимуществ иерархического подхода проведем решение для шести сочленений: q1, q2, q5, q8, q9 и q10.

Поворот манипулятора к цели будет осуществляться с помощью сочленения q1, сближение с целью – с помощью сочленений q2, q5, q8 и q9 в следующем порядке:

$$q2 \rightarrow q5 \rightarrow q8 \rightarrow q9$$

За достижение требуемой ориентации схвата, как и прежде, будут отвечать сочленения q5, q9 и q10. Порядок изменения обобщенных координат будет следующим:

$$q9 \rightarrow q10 \rightarrow q5$$

Все остальные исходные данные оставим прежними. Как и прежде, было проведено 20 экспериментов, все они дали одинаковый результат.

Момент завершения решения представлен на рис. 11. Как видно из этого рисунка, модель манипулятора имеет меньше изломов, чем на рис. 4, что связано с меньшим количеством используемых сочленений. Представленному на рисунке положению манипулятора соответствует следующая матрица положения и ориентации:

$$T_N = \begin{bmatrix} -0,0070 & -0,4291 & 0,9032 & 2,0545 \\ 0,0071 & -0,9032 & -0,4291 & -0,9012 \\ 1 & 0,0034 & 0,0093 & 0,6384 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

Отклонение оси X10 от Z0 составило $30^{\circ}56,5''$, расстояние от центра цели до начала системы координат схвата составило 0,0595 м.

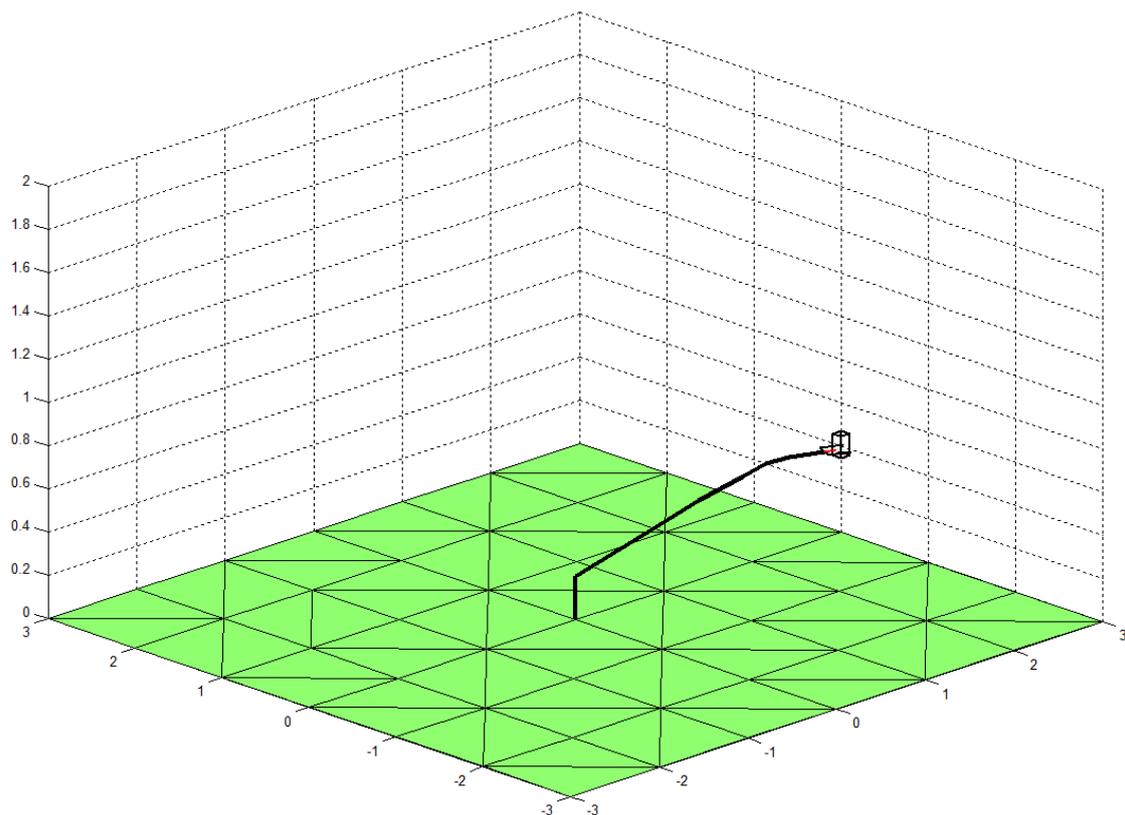


Рис. 11. Момент завершения решения\

Данные о скорости алгоритма приведены в табл. 6.

Таблица 6. Скорость решения обратной задачи кинематики

№ экспери- мента	Время, затраченное на различные процессы в ходе решения, с		Полное вре- мя решения, с
	Собственное время работы алгоритма решения	Решение прямой задачи кинематики для схвата	
1	0,045	0,179	0,225
2	0,043	0,173	0,217
3	0,044	0,183	0,229
4	0,043	0,174	0,218
5	0,042	0,172	0,216
6	0,043	0,174	0,219
7	0,042	0,172	0,216
8	0,042	0,173	0,217
9	0,042	0,173	0,216
10	0,044	0,181	0,226
11	0,045	0,189	0,236
12	0,045	0,180	0,227
13	0,044	0,179	0,225
14	0,045	0,186	0,233
15	0,045	0,177	0,224
16	0,043	0,175	0,220
17	0,045	0,177	0,223
18	0,045	0,187	0,233
19	0,043	0,174	0,219
20	0,044	0,175	0,220

В процессе решения прямая задача кинематики была решена 4486 раз.

Значения обобщенных координат, полученные в процессе решения, приведены в табл. 7.

Таблица 7. Значения обобщенных координат манипулятора

Сочленение	Значение обобщенной координаты, рад		
	До решения	После решения	Приращение Δq
q1	0	2,7332	2,7332
q2	0,7854	0,2553	-0,5301
q5	-1,5708	-0,0353	1,5355
q8	0	-0,1257	-0,1257
q9	0,7854	-0,0852	-0,8706
q10	0	$1,5 \times 10^{-18}$	$1,5 \times 10^{-18}$

8. Обсуждение

Данный раздел посвящен обсуждению преимуществ и недостатков иерархического подхода, а также некоторых аспектов реализации решения на его основе.

Приведенные выше результаты моделирования (для девяти и для шести сочленений) показывают, что иерархический подход позволяет получить однозначные, не требующие дополнительных проверок решения в условиях реального времени и при наличии кинематической избыточности исполнительного механизма.

Однозначность полученных решений определяется тем, что в основу решения изначально положено движение, которое исполнительный механизм должен реализовывать на практике, т.е. во время отработки полученного решения приводами.

В программной реализации решения это проявляется в 1) разбиении указанного движения на отдельные этапы (поворот к цели, сближение с целью, получение требуемой ориентации схвата) и 2) задании строгой последовательности движения сочленений на каждом из этапов.

Отсутствие необходимости дополнительных проверок полученного решения определяется тем, что ограничения изменений обобщенных координат изначально учтены в алгоритме решения. Значения обобщенных координат, полученные в процессе решения, лежат внутри заданных ограничений, что подтверждается данными таблиц 2 и 7. Следовательно, исполнительный механизм сможет отработать полученное решение.

Возможность решения в условиях реального времени подтверждается данными таблиц 3 и 6, из которых видно, что максимальное время решения не превышает 0,255 с для девяти сочленений.

Возможность решения при наличии кинематической избыточности исполнительного механизма подтверждается тем, решения получены для манипулятора, исполнительный механизм которого имеет 10 сочленений, из которых от 6 до 9 задействованы в процессе решения.

Сравнение с методом решения обратной задачи кинематики, основанным на использовании псевдообратной матрицы Якоби, показывает, что иерархический подход дает более точное решение за меньшее время.

Кроме того, в процессе подготовки к решению указанным методом возникают свои сложности.

1. Формула (6) применима лишь при малых величинах приращения компонент вектора положения и ориентации схвата Δs , что увеличивает временные затраты на решение.

2. Чтобы определить указанные приращения необходимо знать положение и ориентацию схвата в промежуточных точках рабочего пространства.

3. Чтобы знать положение и ориентацию схвата в промежуточных точках рабочего пространства разработчику необходимо задавать траекторию движения схвата, выбор которой является многовариантным.

Однако даже большое число промежуточных точек не гарантирует получение подходящего решения. Примером являются решения, полученные для 1200 и 2000 точек на траектории схвата.

При 1200 точках исполнительный механизм занял, положение, приведенное на рисунке 10, д. Если плоскости, на которой стоит модель манипулятора, соответствует опорная поверхности реального механизма, то манипулятор не сможет отработать полученное решение. При 2000 точках решение оказалось менее точным, чем при 1100.

Следующим преимуществом иерархического подхода является отсутствие необходимости предварительного обучения модели перед решением, что позволяет использовать подход в непредсказуемых внешних условиях.

Следует отметить, что иерархический подход применим в тех случаях, когда в исполнительном механизме имеются кинематические пары с числом степеней свободы $k > 1$. В этих случаях можно назначить последовательность изменения обобщенных координат, соответствующих кинематической паре.

Возможность применения подхода к механизмам с несколькими конечностями продемонстрирована в [17, 18, 19] на примере шагающих роботов.

В заключение стоит сказать об актуальности предложенного подхода с точки зрения адаптации роботов не только к меняющимся условиям рабочего пространства, но и к меняющимся возможностям самих роботов. Действительно, в процессе эксплуатации робота может заклинить то или иное сочленение. В случае с шагающими роботами возможна утрата отдельных конечностей (к примеру, в результате подрыва на mine). Если оставшихся сочленений достаточно для выполнения задачи, робот сможет перестроить свой алгоритм управления, исключив сломанные сочленения (или даже конечности) из расчетов.

Подобные исследования проводятся в мире в настоящее время. Так, в [20] приведены результаты адаптации роботов к новым условиям работы, которые вызваны потерей конечностей или заклиниванием отдельных сочленений.

Представленный в данной статье подход как нельзя лучше соответствует этим новым веяниям, так как позволяет использовать разные наборы сочленений для одного и

того же механизма, что было продемонстрировано в данной статье на примере получения решения с использованием девяти и шести сочленений исполнительного механизма манипулятора.

Тем не менее, наряду с преимуществами иерархический подход на данный момент имеет недостатки.

Первым из них является необходимость решения прямой задачи кинематики для определения текущего положения и ориентации схвата при коррекции обобщенных координат, о чем говорилось выше. В случае с девятью сочленениями прямая задача кинематики была решена 4758 раз в каждом из двадцати экспериментов. Из табл. 3 и 6 видно, что временные затраты на решение прямой задачи составляют основную часть затрат при решении обратной задачи.

Данный недостаток можно устранить или существенно снизить его влияние, оптимизируя функцию решения прямой задачи кинематики и число вызовов этой функции, чем автор и занимается в настоящее время.

Второй недостаток состоит в том, что разработчику придется самому выбирать способ коррекции величин приращений обобщенных координат в процессе решения. Величины приращений, рассчитанные по формуле (4) при $n = 100$, могут в определенных условиях оказаться слишком большими, что приведет к тому, что исполнительный механизм начнет постоянно перескакивать через цель в процессе решения и сам процесс решения может заиклиться.

Во избежание подобной ситуации автор предлагает использовать следующий механизм коррекции величин приращений обобщенных координат: при достижении схватом некоторого расстояния до цели или ориентации относительно цели, величина приращения одной или нескольких обобщенных координат, отвечающих за положение и/или ориентацию схвата, уменьшается в m раз, где m – выбирается разработчиком. Подобным образом (при достижении схватом некоторого положения или ориентации) можно уменьшать величины приращений несколько раз в процессе решения. Также разработчик может назначить уникальный коэффициент m для каждой обобщенной координаты.

Данным приемом автор пользовался при написании статьи. Разработка методики коррекции величин приращений является отдельным исследованием и выходит за рамки данной статьи.

Заключение

Предложенный подход позволяет реализовывать решения обратной задачи кинематики для исполнительных механизмов с различными кинематическими схемами, в том числе древовидными. При этом не имеет значения, какую роль выполняет та или иная конечность исполнительного механизма – «ноги» или схвата.

Общность подхода при реализации решения обратной задачи кинематики обеспечивается тем, что в основу решения положены движения, которые исполнительные механизмы должны отрабатывать на практике. Таким образом, особенности конструкции, за-

дачи, стоящие перед исполнительным механизмом, учитываются разработчиком при назначении реализуемого движения.

Список литературы

1. Зенкевич С.Л., Ющенко А.С. Основы управления манипуляционными роботами: учеб. для вузов. 2-е изд., перераб. и доп. М.: Изд-во МГТУ им. Н.Э. Баумана, 2004. 480 с.
2. Kumar V., Gardner J.F. Kinematics of redundantly actuated closed chains // IEEE Transactions on Robotics and Automation. 1990. Vol. 6, iss. 2. P. 269-274. DOI: [10.1109/70.54745](https://doi.org/10.1109/70.54745)
3. Aristidou A., Lasenby J. Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver. Technical Report. CUED/F-INFENG/TR-632. University of Cambridge, 2009. 74 p.
4. Welman C. Inverse kinematics and geometric constraints for articulated figure manipulation: Master of Science in Computing Science. Simon Fraser University, Department of Computer Science, Burnaby, Canada, 1993. 84 p. Режим доступа: <https://www.cse.iitb.ac.in/~paragc/teaching/2014/cs775/notes/welman.pdf> (дата обращения 01.02.2016).
5. Pechev A.N. Inverse Kinematics without matrix inversion // Proc. 2008 IEEE International Conference on Robotics and Automation (ICRA 2008), 2008, 19-23 May, Pasadena, USA. Pasadena, 2008. P. 2005-2012. DOI: [10.1109/ROBOT.2008.4543501](https://doi.org/10.1109/ROBOT.2008.4543501)
6. Zhao J., Badler N.I. Inverse kinematics positioning using nonlinear programming for highly articulated figures // ACM Transactions on Graphics (TOG). 1994. Vol. 13, iss. 4. P. 313-336. DOI: [10.1145/195826.195827](https://doi.org/10.1145/195826.195827)
7. Castellet A., Thomas F. Using interval methods for solving inverse kinematic problems // Proc. of the NATO Advanced Study Institute on Computational Methods in Mechanisms (NATO-ASI), vol. 2, 1997, 16-28 June, Varna, Bulgaria. Varna, 1997. P. 135-145.
8. Courty N., Arnaud E. Inverse Kinematics Using Sequential Monte Carlo Methods // Proc. of the 5th International Conference on Articulated Motion and Deformable Objects (AMDO 2008), 2008, 9-11 July, Port d'Andratx, Mallorca, Spain. Port d'Andratx, 2008. P. 1-10. DOI: [10.1007/978-3-540-70517-8_1](https://doi.org/10.1007/978-3-540-70517-8_1)
9. Grochow K., Martin S.L., Hertzmann A., Popovic Z. Style-based inverse kinematics // ACM Transactions on Graphics (TOG) - Proc. of ACM SIGGRAPH 2004. 2004. Vol. 23, iss. 3. P. 522-531. DOI: [10.1145/1015706.1015755](https://doi.org/10.1145/1015706.1015755)
10. Sumner R.W., Zwicker M., Gotsman C., Popovic J. Mesh-based inverse kinematics // ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2005. 2005. Vol. 24, iss. 3. P. 488-495. DOI: [10.1145/1073204.1073218](https://doi.org/10.1145/1073204.1073218)
11. Unzueta L., Peinado M., Boulic R., Suescun Á. Full-body performance animation with Sequential Inverse Kinematics // Graphical Models. 2008. Vol. 70, iss. 5. P. 87-104. DOI: [10.1016/j.gmod.2008.03.002](https://doi.org/10.1016/j.gmod.2008.03.002)

12. Wang L.C.T., Chen C.C. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators // IEEE Transactions on Robotics and Automation. 1991. Vol. 7, iss. 4. P. 489-499. DOI: [10.1109/70.86079](https://doi.org/10.1109/70.86079)
13. Müller-Cajar R., Mukundan R. Triangulation: A new algorithm for inverse kinematics // Proceedings of Image and Vision Computing, 2007, December, Hamilton, New Zealand. Hamilton, 2007. P. 181-186. Режим доступа: <http://ir.canterbury.ac.nz/handle/10092/743> (дата обращения 01.02.2016).
14. Brown J., Latombe J.-C., Montgomery K. Real-time knot-tying simulation // The Visual Computer. 2004. Vol. 20, iss. 2. P. 165-179. DOI: [10.1007/s00371-003-0226-y](https://doi.org/10.1007/s00371-003-0226-y)
15. Aristidou A., Lasenby J. FABRIK: A fast, iterative solver for the Inverse Kinematics problem // Graphical Models. 2011. Vol. 73, iss. 5. P. 243-260. DOI: [10.1016/j.gmod.2011.05.003](https://doi.org/10.1016/j.gmod.2011.05.003)
16. Охоцимский Д.Е., Голубев Ю.Ф. Механика и управление движением автоматического шагающего аппарата. М.: Наука, 1984. 312 с.
17. Каргинов Л.А. Синтез управляющих воздействий для приводов исполнительных механизмов шагающих роботов без решения обратной задачи кинематики // Инженерный журнал: наука и инновации. 2013. Вып. 4 (16). DOI: [10.18698/2308-6033-2013-4-692](https://doi.org/10.18698/2308-6033-2013-4-692)
18. Каргинов Л.А. Пример синтеза управляющих воздействий для шестиногого шагающего робота // Молодежный научно-технический вестник. МГТУ им. Н.Э. Баумана. Электрон. журн. 2014. № 9. Режим доступа: <http://sntbul.bmstu.ru/doc/731009.html> (дата обращения 14.11.2015).
19. Каргинов Л.А. Пример синтеза управляющих воздействий для шестиногого шагающего робота при передвижении по неровной поверхности // Машины и Установки: проектирование, разработка и эксплуатация. 2015. № 03. С. 70-88. Режим доступа: <http://maplantsjournal.ru/doc/782457.html> (дата обращения 01.02.2016).
20. Cully A., Clune J., Tarapore D., Mouret J.-B. Robots that can adapt like animals // Nature. 2015. Vol. 521. P. 503-507. Режим доступа: <http://www.nature.com/nature/journal/v521/n7553/abs/nature14422.html> (дата обращения 01.02.2016).

Hierarchical Approach to Solving an Inverse Problem of Kinematics

L.A. Karginov^{1,*}

[*herstellung@mail.ru](mailto:herstellung@mail.ru)

¹Bauman Moscow State Technical University, Moscow, Russia

Keywords: inverse kinematic problem, robot, algorithm, generalized coordinates, hierarchical approach, solving

The paper presents a hierarchical approach to solving an inverse problem of kinematics.

The main idea of the approach is to take, as a basis of solution, a motion (or set of motions) that the robot device has to make in practice.

Suppose that the robot device is open kinematics having N joints.

To implement the solution to the inverse kinematics problem it is necessary:

1. Select a motion (or a set of motions) to be taken as a basis of solution.
2. Select joints that will be involved in making the specified motion. Let M joints be responsible for making motion, wherein $M < N$ (if to make motion is necessary to use all the joints, then $M = N$).

If $M < N$, then the remaining $(N - M)$ joints will be fixed, the algorithm of solution will not take into consideration their generalized coordinates, and the values of these coordinates will be constant.

If a set of motions is specified, we select M_i joints for each motion where i - number of motion.

3. Assign a motion succession of selected M joints while making the specified motion.
4. Assign the limits for changes of generalized coordinates in the selected joints.

A solution of the inverse kinematics problem is a gradual approximation to the specified gripper position and orientation by changing (increasing or decreasing) the values of generalized coordinates in the same succession in which the selected joints have to move.

The solving process is cyclic. At each step of the cycle generalized coordinates of the selected M joints move in the above succession, each coordinate being under a series of actions described below.

Each generalized coordinate changes by the value of the increment that corresponds to this coordinate at each step of the solution cycle.

Then, the following verifications are made for this generalized coordinate:

- to verify an impact of the generalized coordinates on the gripper – target distance and / or on the gripper orientation;

- to verify if a generalized coordinate achieves the developer-specified limits of its change. Depending on the results of verifications the following actions are undertaken:
 1. If the distance to the target has decreased and / or the gripper orientation has become closer to the desired, additional corrective actions are unnecessary, and we go on to the next generalized coordinate.
 2. If the change of the generalized coordinate has led to a longer distance from the target or increasing angular deviation from the target position, but **its value has not yet reached the limits of change**, the increment sign of the this coordinate becomes inverse.
 3. If the value of the generalized coordinate reached the limits of change, then its corresponding increment also changes its sign. If **after changing the increment sign** the gripper is still distant from the target, the last change of the generalized coordinates is cancelled.

Note: a technique for changing increment signs of generalized coordinate enables to avoid a situation where the robot device that has reached the limits of several coordinates, fails to reach the target.

In accordance with the idea of a hierarchical approach (**taking some motion as a basis of solution**), to each of the M joints responsible for the solution a developer assigns a specific "role" in the making a specified motion. For example, one joint may be responsible for rotation towards the target; the other two - for a target approach, joints that are closer to the gripper - for the orientation of the gripper, etc. Some joints may be involved both in the target approach process and in specifying a desired orientation.

The solving algorithm shows such "roles" of joints at the stage of verification of appropriate generalized coordinates:

- a changing target – gripper distance is verified for the joint being responsible only for target approach;
- angular deviations of the gripper unit vectors of the gripper coordinate system from the coordinate system axes of the working space, or other axes selected by the developer are verified for the joint being responsible for gripper orientation;
- if the joint is responsible for position and orientation both types of verifications are carried out.

If the set of motions is taken as basis of solution the solving process has a succession of cycles performed (from one motion to another).

The solution is completed when the requirements for a specified precision of gripper position and orientation are met.

In fact, the proposed approach has innumerable number of examples in nature. One of them may be a human hand: to take something off the table people consistently use the certain joints of a hand. Thus, a shoulder "drive" is more responsible for guidance to the target, "drives" of the elbow joints – for target approach, "drives" of wrist - for "gripper" orientation and interaction with the target. Since, when making the motions, a human uses all hand joints, there is no need to lock individual joints (the case where $M = N$).

References

1. Zenkevich S.L., Yushchenko A.S. *Osnovy upravleniya manipulyatsionnymi robotami: ucheb. dlya vuzov* [Fundamentals of management manipulation robots: tutorial for universities]. Moscow, Bauman MSTU Publ., 2004. 480 p. (in Russian).
2. Kumar V., Gardner J.F. Kinematics of redundantly actuated closed chains. *IEEE Transactions on Robotics and Automation*, 1990, vol. 6, iss. 2, pp. 269-274. DOI: [10.1109/70.54745](https://doi.org/10.1109/70.54745)
3. Aristidou A., Lasenby J. *Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver. Technical Report. CUED/F-INFENG/TR-632*. University of Cambridge, 2009. 74 p.
4. Welman C. *Inverse kinematics and geometric constraints for articulated figure manipulation. Master of Science in Computing Science*. Simon Fraser University, Department of Computer Science, Burnaby, Canada, 1993. 84 p. Available at: <https://www.cse.iitb.ac.in/~paragc/teaching/2014/cs775/notes/welman.pdf> , accessed 01.02.2016.
5. Pechev A.N. Inverse Kinematics without matrix inversion. *Proc. 2008 IEEE International Conference on Robotics and Automation (ICRA 2008)*. 2008, 19-23 May, Pasadena, USA. Pasadena, 2008, pp. 2005-2012. DOI: [10.1109/ROBOT.2008.4543501](https://doi.org/10.1109/ROBOT.2008.4543501)
6. Zhao J., Badler N.I. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics (TOG)*, 1994, vol. 13, iss. 4, pp. 313-336. DOI: [10.1145/195826.195827](https://doi.org/10.1145/195826.195827)
7. Castellet A., Thomas F. Using interval methods for solving inverse kinematic problems. *Proc. of the NATO Advanced Study Institute on Computational Methods in Mechanisms (NATO-ASI). Vol. 2*. 1997, 16-28 June, Varna, Bulgaria. Varna, 1997, pp. 135-145.
8. Courty N., Arnaud E. Inverse Kinematics Using Sequential Monte Carlo Methods. *Proc. of the 5th International Conference on Articulated Motion and Deformable Objects (AMDO 2008)*. 2008, 9-11 July, Port d'Andratx, Mallorca, Spain. Port d'Andratx, 2008, pp. 1-10. DOI: [10.1007/978-3-540-70517-81](https://doi.org/10.1007/978-3-540-70517-81)
9. Grochow K., Martin S.L., Hertzmann A., Popovic Z. Style-based inverse kinematics. *ACM Transactions on Graphics (TOG) - Proc. of ACM SIGGRAPH 2004*, 2004, vol. 23, iss. 3, pp. 522-531. DOI: [10.1145/1015706.1015755](https://doi.org/10.1145/1015706.1015755)
10. Sumner R.W., Zwicker M., Gotsman C., Popovic J. Mesh-based inverse kinematics. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2005*, 2005, vol. 24, iss. 3, pp. 488-495. DOI: [10.1145/1073204.1073218](https://doi.org/10.1145/1073204.1073218)
11. Unzueta L., Peinado M., Boulic R., Suescun Á. Full-body performance animation with Sequential Inverse Kinematics. *Graphical Models*, 2008, vol. 70, iss. 5, pp. 87-104. DOI: [10.1016/j.gmod.2008.03.002](https://doi.org/10.1016/j.gmod.2008.03.002)

12. Wang L.C.T., Chen C.C. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 1991, vol. 7, iss. 4, pp. 489-499. DOI: [10.1109/70.86079](https://doi.org/10.1109/70.86079)
13. Müller-Cajar R., Mukundan R. Triangulation: A new algorithm for inverse kinematics. *Proceedings of Image and Vision Computing*. 2007, December, Hamilton, New Zealand. Hamilton, 2007, pp. 181-186. Available at: <http://ir.canterbury.ac.nz/handle/10092/743> , accessed 01.02.2016.
14. Brown J., Latombe J.-C., Montgomery K. Real-time knot-tying simulation. *The Visual Computer*, 2004, vol. 20, iss. 2, pp. 165-179. DOI: [10.1007/s00371-003-0226-y](https://doi.org/10.1007/s00371-003-0226-y)
15. Aristidou A., Lasenby J. FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graphical Models*, 2011, vol. 73, iss. 5, pp. 243-260. DOI: [10.1016/j.gmod.2011.05.003](https://doi.org/10.1016/j.gmod.2011.05.003)
16. Okhotsimskii D.E., Golubev Yu.F. *Mekhanika i upravlenie dvizheniem avtomaticheskogo shagayushchego apparata* [Mechanics and control of movement automatic walking apparatus]. Moscow, Nauka Publ., 1984. 312 p. (in Russian).
17. Karginov L.A. The synthesis of control actions to drive the actuators of walking robots without solving the inverse kinematics task. *Inzhenernyy zhurnal: nauka i innovatsii = Engineering Journal: Science and Innovation*, 2013, iss. 4. DOI: [10.18698/2308-6033-2013-4-692](https://doi.org/10.18698/2308-6033-2013-4-692) (in Russian).
18. Karginov L.A. An example of synthesis control actions for six-legged walking robot. *Molodezhnyi nauchno-tehnicheskii vestnik MGTU im. N.E. Baumana = Youth Science and Technology Herald of the Bauman MSTU*, 2014, no. 9. Available at: <http://sntbul.bmstu.ru/doc/731009.html> , accessed 14.11.2015. (in Russian).
19. Karginov L.A. Example of Synthesis of Control Actions for Six-Legged Walking Robot when Moving on Rough Surface. *Machines and Plants: Design and Exploiting*, 2015, no. 03, pp. 70-88. Available at: <http://maplantsjournal.ru/doc/782457.html> , accessed 01.02.2016. (in Russian).
20. Cully A., Clune J., Tarapore D., Mouret J.-B. Robots that can adapt like animals. *Nature*, 2015, vol. 521, pp. 503-507. Available at: <http://www.nature.com/nature/journal/v521/n7553/abs/nature14422.html> , accessed 01.02.2016.