

УДК 621.865.8

Пример синтеза управляющих воздействий для шестиногого шагающего робота

*Каргинов Л.А., инженер
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Гидравлика, гидроприводы и гидропневмоавтоматика»*

*Научный руководитель: Ковальчук А.К., к.т.н, доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
bauman@bmstu.ru*

Введение

Для получения управляющих воздействий необходимо решать обратную задачу кинематики для конечностей исполнительных механизмов шагающих роботов. В первой части статьи представлена критика существующих методов. Вторая часть посвящена рассмотрению предлагаемого подхода. Третья и четвертая части посвящены примеру применения подхода и результатам моделирования соответственно.

1. Критика существующих методов решения обратной задачи кинематики

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, 2, 3, 4], методы Ньютона [2], методы нелинейного программирования [5] и метод интервалов [6] 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) [7], Style-based Inverse Kinematics [8] и Mesh-based Inverse Kinematics [9] требуют обязательного предварительного обучения модели, что недопустимо в условиях непредсказуемости условий окружающей среды.

Метод SIK [10] предназначен для моделирования движений человека, для других систем придется переписывать алгоритмы.

Методы CCD [11] и Triangulation Inverse Kinematics [12] 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 [13] также может быть применен для описания кинематических цепей манипуляторов, особенно в тех случаях, когда нужно проследить 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 – согласно [2] и [14], быстрее из методов, рассмотренных выше. Метод позволяет решать обратную задачу кинематики для систем с несколькими конечностями. Однако для использования FABRIK, как и для других выше перечисленных методов, необходима информация о целевой точке, что в изменяющихся внешних условиях не всегда возможно, например, в темноте или при поврежденной системе технического зрения.

Метод шаговых циклов [15] не требует выбора определенной точки на опорной поверхности. Этот метод удобен при описании движения шагающих роботов, однако, он был разработан для механизмов, имеющих не более трех сочленений в ноге (отсутствует кинематическая избыточность [16]).

2. Новый подход к решению обратной задачи кинематики

Основная идея подхода [17] состоит в следующем:

1. Из всех сочленений робота выбирается ограниченный набор сочленений, необходимых для реализации избранной походки [15]. Для этих сочленений назначается строгая последовательность движения в течение каждого шага и ограничения изменений обобщенных координат.

2. Сочленения, не участвующие в реализации избранной походки, блокируются, для них расчеты не проводятся.

Из сказанного следует, что источниками исходных данных для решения обратной задачи кинематики являются кинематическая схема исполнительного механизма шагающего робота и избранная походка.

Для применения предлагаемого подхода необходимо:

1. Пронумеровать ноги и сочленения в ногах.
2. Выбрать сочленения, которые будут задействованы в реализации избранной походки.
3. Назначить последовательность смены опорных ног при движении избранной походкой.
4. Назначить последовательность движения выбранных сочленений в течение шага для каждой ноги.
5. Назначить ограничения изменения обобщенных координат в выбранных сочленениях.

Процесс решения обратной задачи кинематики заключается в постепенном приближении к решению путем изменения (увеличения или уменьшения) обобщенных координат в том же порядке, в котором соответствующие этим координатам сочленения ноги перемещаются в течение шага при движении избранной походкой. Критерием завершения вычислений является достижение пределов или факт постановки ноги на опорную поверхность, фиксируемый контактным датчиком (или условием в программе моделирования).

Изменения обобщенных координат происходят в цикле, каждая обобщенная координата изменяется на определенную величину на каждой итерации цикла. Полное время цикла соответствует предполагаемому времени совершения шага.

Достоинствами подхода являются: однозначность получаемого решения, возможность рассмотрения вырожденных конфигураций исполнительного механизма шагающего робота, возможность вычислений в условиях кинематической избыточности, высокая скорость вычислений.

Возможность рассмотрения вырожденных конфигураций связана с тем, что при вычислениях не используется аппарат матричных преобразований (в частности, вычисление обратной матрицы Якоби).

Возможность вычисления в условиях кинематической избыточности, однозначность получаемого решения и высокое быстродействие обусловлены тем, что последовательность движения сочленений изначально устанавливает определенную желаемую конфигурацию ноги в пространстве и, по сути, задает направление решения.

Проверка получаемого решения не требуется, так как кинематические ограничения уже учтены в процессе вычислений.

Еще одним важным преимуществом подхода является тот факт, что роботы могут иметь произвольные количества ног, а также сочленений в ногах.

Недостатком подхода можно назвать отсутствие возможности построения универсальных алгоритмов для шагающих роботов с принципиально разными кинематическими схемами (например, антропоморфными и паукообразными) ввиду конструктивных отличий исполнительных механизмов. Но с другой стороны, этим же недостатком обладают и другие эвристические методы, такие как FABRIK, FTL или CCD. С этими методами данный подход роднит тот факт, что не существует определенной формулы, использование которой позволяет сразу получить решение. Есть лишь алгоритмы поиска, проверяющие при вычислениях набор заданных условий, являющихся критериями коррекции или завершения решения.

3. Построение алгоритма для походки «трешки»

Походка «трешки» названа так в связи с тем, что ноги шестиногого робота при движении переставляются группами (трешками), состоящими из первой и последней ног с одной стороны робота и средней ноги с другой [18].

3.1. Стадии подготовки данных

П1. Кинематическая схема робота представлена на рисунке 1.

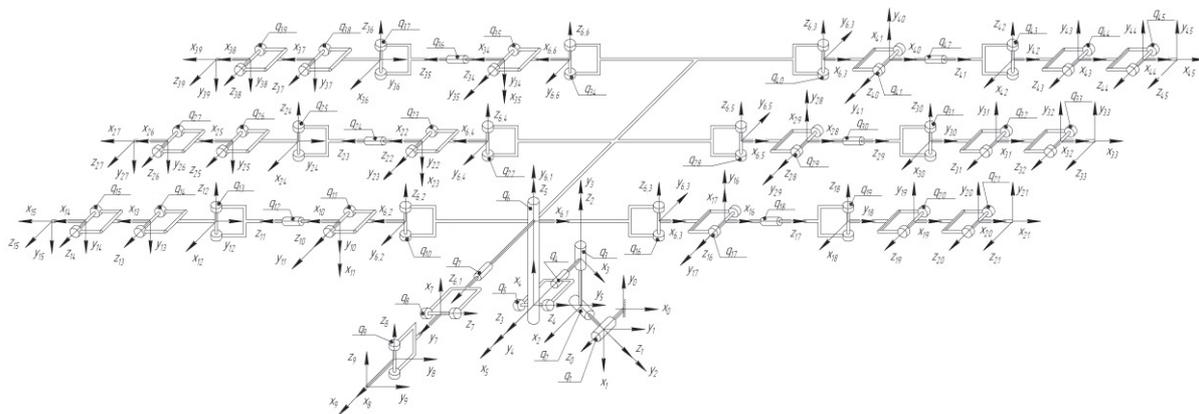


Рис. 1. Кинематическая схема шестиногого шагающего робота

Порядок нумерации ног, соответствующий кинематической схеме на рисунке 1, представлен на рисунке 2. Следовательно, для походки «трешки» имеем две группы ног: (1, 4, 5) и (2, 3, 6).

П2. В качестве начального положения выберем положение, представленное на рисунке 3. Параметры начального положения представлены в таблице 1.

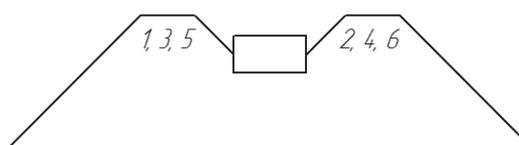
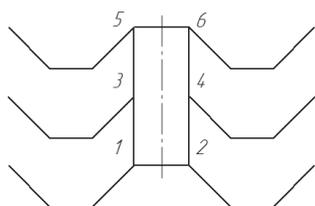


Рис. 2. Номера ног робота Рис. 3. Начальное положение робота

Таблица 1

Ограничения обобщенных координат и параметры начального положения

Обобщенная координата	Значения обобщенных координат, градусы		
	Минимальное	Максимальное	Начальное
Правые ноги			
q10, q22, q34	-15	15	0
q11, q23, q35	-60	-30	-45
q12, q24, q36	-90	90	0
q13, q25, q37	-30	30	0
q14, q26, q38	30	60	45
q15, q27, q39	30	60	45
Левые ноги			
q16, q28, q40	-15	15	0
q17, q29, q41	30	60	45
q18, q30, q42	-90	90	0
q19, q31, q43	-30	30	0
q20, q32, q44	-60	-30	-45
q21, q33, q45	-60	-30	-45

Примечание: фиктивные сочленения [19], а также сочленения q7, q8 и q9 (сочленения головы робота) в перемещении робота не участвуют, поэтому их параметры отсутствуют в таблице 1.

3.2. Основные стадии использования подхода

1. Последовательность смены опорных ног.

Для походки «трешки» последовательность будет следующей (см. рисунок 2):

$$\dots \rightarrow 1, 4, 5 \rightarrow 2, 3, 6 \rightarrow 1, 4, 5 \rightarrow \dots$$

2. Стадии шага для всех ног в **данном** случае будут одинаковыми. Это связано с тем, что конструкция ног полагается одинаковой. На рисунке 4 показаны фаза переноса и фаза опоры. На рисунках 5 и 6 показаны подъем и опускание ноги робота соответственно.

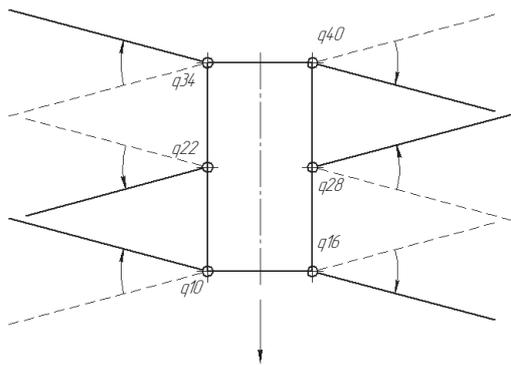


Рис. 4. Перенос корпуса робота

Пунктирными линиями показаны исходные положения ног робота, сплошными линиями – конечные.

Для фазы опоры:

- перенос корпуса робота вперед с помощью сочленений q_{10} , q_{22} , q_{34} , q_{16} , q_{28} , q_{40} .

На рисунке 4 в фазе опоры находятся ноги 1, 4 и 5 (перемещаемые сочленениями q_{10} , q_{28} и q_{34} соответственно), на что указывают соответствующие стрелки, направленные против движения робота.

Для фазы переноса:

- подъем ног над опорной поверхностью с помощью сочленений q_{11} , q_{23} , q_{35} , q_{14} , q_{26} , q_{38} , q_{17} , q_{29} , q_{41} , q_{20} , q_{32} , q_{44} (на рисунке 5 процесс подъема ноги показан на примере первой ноги робота);

- перенос ног вперед с помощью сочленений q_{10} , q_{22} , q_{34} , q_{16} , q_{28} , q_{40} (см. рис. 4: в фазе переноса находятся ноги 2, 3 и 6, перемещаемые сочленениями q_{16} , q_{22} и q_{40} ; на перенос ног указывают соответствующие стрелки, направленные на рисунке «вниз»);

- опускание ног на опорную поверхность с помощью сочленений q_{11} , q_{23} , q_{35} , q_{14} , q_{26} , q_{38} , q_{17} , q_{29} , q_{41} , q_{20} , q_{32} , q_{44} (на рисунке 6 процесс опускания ноги показан на примере первой ноги робота).

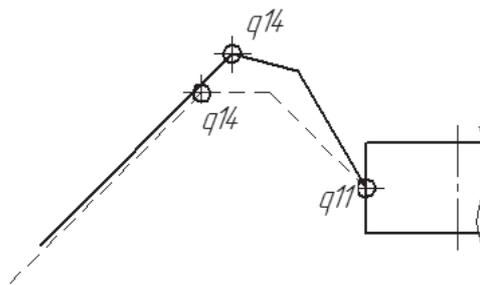


Рис. 5. Подъем первой ноги над опорной поверхностью

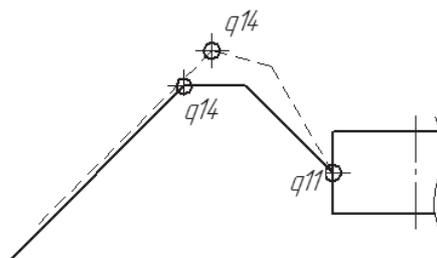


Рис. 6. Опускание первой ноги на опорную поверхность

3. Назначение ограничений изменения обобщенных координат.

Ограничения соответствуют таблице 1.

4. Назначение времени на выполнение одного шага T исходя из требуемой скорости движения корпуса робота.

Назначим $T = 0,5$ с, что соответствует для данного механизма скорости корпуса $V_{корп} = 1,7388$ м/с.

Из рассмотренных выше стадий шага следует, что в течение шага обобщенные координаты изменяются не одновременно, что проиллюстрировано на рисунке 7 на примере первой ноги.

В фазе опоры (участок I) работает сочленение q_{10} , участвующее наряду с аналогичными ему сочленениями других ног в процессе переноса корпуса робота. Изменение соответствующей обобщенной координаты представлено на рисунке 7 наклонным участком.

В фазе переноса:

- на участке II (подъем ноги) работают сочленения q_{11} и q_{14} , остальные сочленения остановлены (горизонтальные участки на графиках соответствуют отсутствию изменений обобщенных координат);

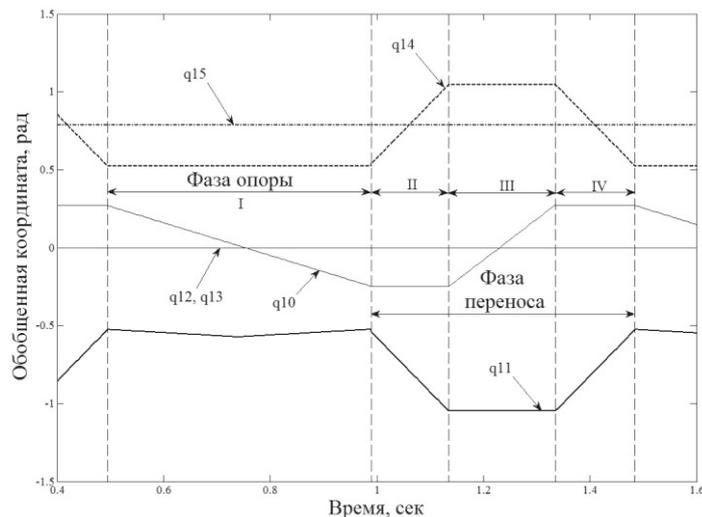


Рис. 7. Фазы опоры и переноса первой ноги робота

- на участке III (перенос ноги вперед) работает сочленение q_{10} , остальные сочленения остановлены;

- на участке IV (опускание ноги на опорную поверхность) работают сочленения q_{11} и q_{14} , остальные сочленения остановлены.

5. Разбиение времени T на определенное число временных промежутков, соответствующих планируемому числу итераций n (число итераций выбирают из соображений удобства при вычислениях).

Назначим число итераций $n = 100$.

6. Выбор приращений обобщенных координат на каждой итерации $i = 1, \dots, n$.

Приращения обобщенных координат будем выбирать по следующим формулам:

$$\Delta q_k = A_k / n$$

где A_k – величины диапазонов изменения обобщенных координат, k – номер сочленения.

7. Данный этап непосредственно связан с программной реализацией алгоритмов. Ниже приведен алгоритм, соответствующий предыдущим этапам.

Алгоритм 1.

Исходные данные: начальные значения обобщенных координат, ограничения обобщенных координат, номера ног, находящихся в опоре.

Выходные данные: новые значения обобщенных координат.

1. **Проверка нахождения ноги в фазе опоры.** Если нога в фазе опоры, то переход к процедуре 1. Иначе переход к процедуре 2.

2. Процедура 1. Фаза опоры.

2.1. Определение приращений обобщенных координат Δq_i .

2.2. Цикл вычислений (максимум 100 итераций). На каждой итерации:

Если $q_i \in [q_{i\min}, q_{i\max}]$: $q_i = q_i \pm \Delta q_i$. Если $q_i \notin [q_{i\min}, q_{i\max}]$: выход из цикла.

3. Процедура 2. Фаза переноса.

3.1. Определение приращений обобщенных координат Δq_i .

3.2. Разбиение времени шага T на три интервала, соответствующие подъему, переносу и опусканию ноги.

3.3. Цикл вычислений (максимум 100 итераций).

3.3.1. Подъем ноги. Если $q_i \in [q_{i\min}, q_{i\max}]$: $q_i = q_i \pm \Delta q_i$.

3.3.2. Перенос ноги вперед. Если $q_i \in [q_{i\min}, q_{i\max}]$: $q_i = q_i \mp$.

(знак операции « \pm » противоположен знаку той же операции в фазе опоры).

3.3.3. Опускание ноги на опорную поверхность.

Если $q_i \in [q_{imin}, q_{imax}]$ и нога не касается опорной поверхности: $q_i = q_i \mp$

Если $q_i \notin [q_{imin}, q_{imax}]$ или нога коснулась опорной поверхности: выход из цикла.

4. **Конец вычислений.** Считывание значений обобщенных координат.

Примечание: индекс i принимает значения номеров сочленений в соответствии с определенными выше стадиями шага для каждой ноги.

Описанные выше этапы реализованы автором в среде Matlab. Ниже приведены результаты моделирования.

4. Результаты моделирования походки «трешки»

На рисунке 8 представлено начальное положение робота, соответствующее рисунку 3 и таблице 1. На рисунке 9 показаны несколько кадров движения робота походкой «трешки». В процессе моделирования использовались элементы разработанной ранее программы [20]. При моделировании полагалось, что:

- опорная поверхность – ровная;
- робот осуществляет прямолинейное движение с постоянной скоростью.

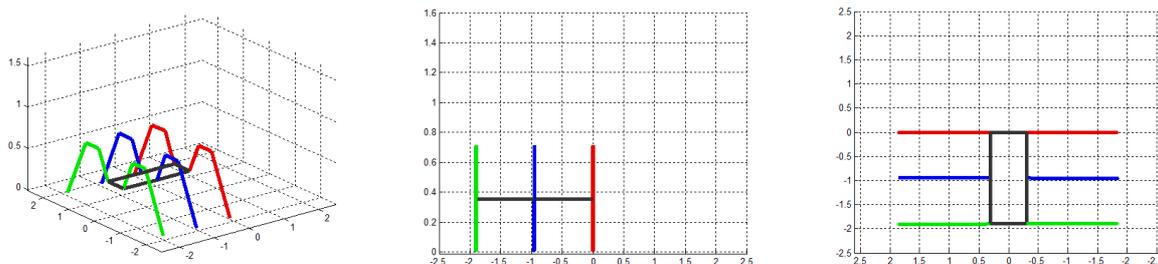


Рис. 8. Начальное положение робота

(слева направо: трехмерный вид, вид справа, вид сверху)

Пройденное роботом расстояние: 12 м.

Время, потраченное на движение: 6,93 с.

Количество шагов, сделанных роботом: 14

Скорость движения робота: 1,7388 м/с.

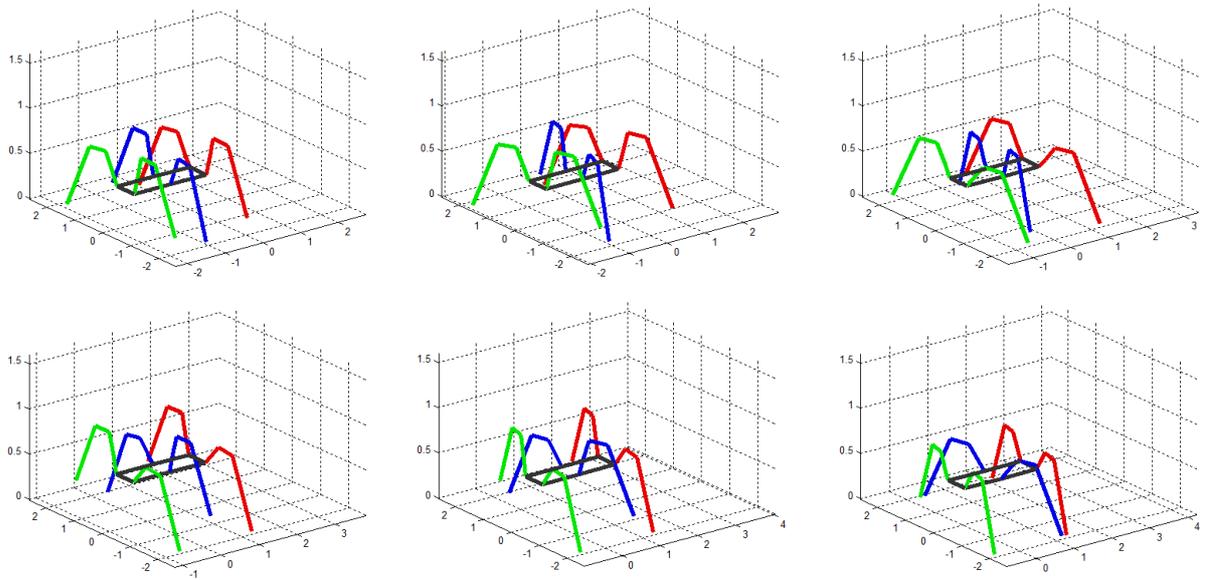


Рис. 9. Походка «трешки»

На рисунке 10 приведены графики изменения обобщенных координат сочленений, которые связывают ноги робота с корпусом, соответствующие рисунку 9. При этом пунктирными линиями показаны графики «левых» ног робота (ноги с номерами 2, 4 и 6, см. рисунок 2), сплошными линиями – «правых».

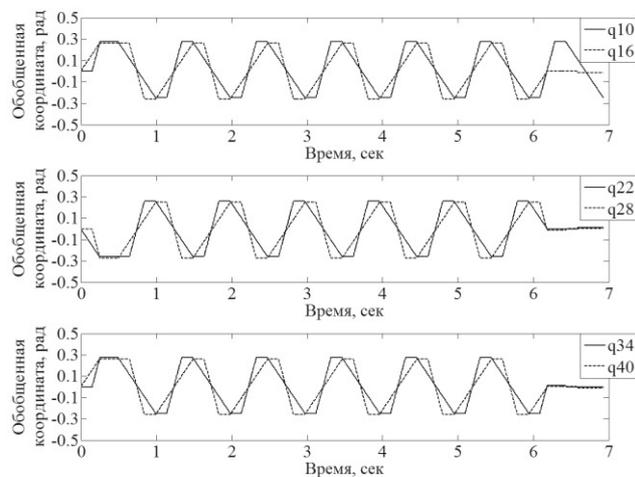


Рис. 10. Обобщенные координаты первых сочленений ног робота

Из приведенных графиков видно, что:

1. Сочленения каждой пары ног (например, ног 1 и 2) работают в противофазе, что соответствует походке «трешки».
2. Изменения обобщенных координат сочленений «правых» ног противоположно изменению обобщенных координат «левых» (например, поворот ноги вперед для

сочленения q_{10} имеет знак «+», для сочленения q_{16} – знак «-»), что обусловлено симметрией исполнительного механизма робота.

3. Горизонтальные участки на графиках изменения обобщенных координат соответствуют временным отрезкам, когда то или иное сочленение неподвижно.

На рисунке 11 показаны графики изменения обобщенных координат передней правой ноги робота. Из приведенных графиков видно, что:

1. Изменения обобщенных координат соответствуют выбранным стадиям шага.

2. Обобщенные координаты изменяются от минимальных значений до максимальных и наоборот. Таким образом, предложенный подход реализует условия повторяемости значений обобщенных координат через определенные (примерно равные в общем случае) промежутки времени [21].

3. Значения обобщенных координат q_{12} , q_{13} и q_{15} остаются постоянными в течение движения и соответствуют таблице 1.

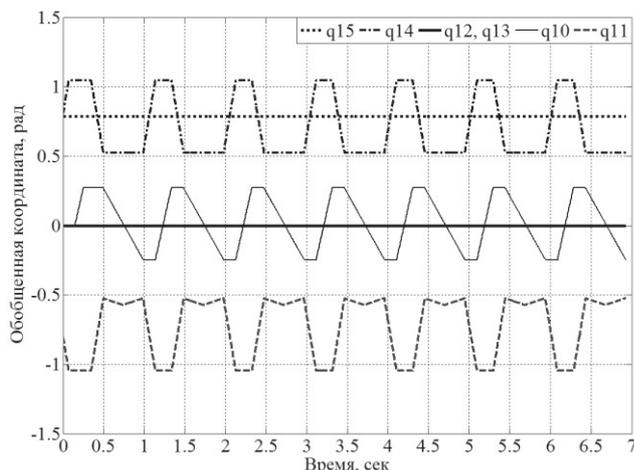


Рис. 11. Обобщенные координаты первой ноги робота

В таблице 2 представлены результаты тестов быстродействия алгоритма моделирования. Время приведено для расчета одного шага шестиногого шагающего робота. При этом расчеты проводились для всех шести ног, то есть количество вычисляемых обобщенных координат равно 18. Время вычислений для одной ноги составило менее 0,001 с.

В [2] и [14] приведены данные анализа быстродействия некоторых методов для решения обратной задачи кинематики. Самым быстрым признан метод FABRIK. Анализ производился для простой кинематической цепи, имеющей 10 сочленений. Скорость вычислений составила 0.01328 секунд в случае, если цель достижима, и 0.06207 секунд, если цель недостижима.

Сравнение полученных автором результатов с результатами из [2] и [14] может показаться некорректным по ряду причин: во-первых, авторы [2] и [14] использовали ЭВМ с другими рабочими параметрами, во-вторых, рассматривали другой механизм, в-третьих, для метода FABRIK необходима информация о целевой точке, а для предложенного подхода эта информация не нужна.

Таблица 2

Анализ времени работы алгоритма

№ теста	Время расчетов, сек	№ теста	Время расчетов, сек
1	0,002	11	0,002
2	0,007	12	0,004
3	0,004	13	0,004
4	0,007	14	0,002
5	0,003	15	0,007
6	0,007	16	0,007
7	0,004	17	0,004
8	0,002	18	0,007
9	0,004	19	0,004
10	0,007	20	0,004
Среднее значение		0,00425	

Тем не менее, приведенные результаты показывают исключительное быстродействие предложенного подхода, дающее возможность его использования для вычислений управляющих воздействий в реальном времени. На данном этапе подход специализирован именно для получения решения обратной задачи кинематики для шагающих роботов.

Заключение

Полученные результаты подтверждают возможность решения обратной задачи кинематики алгоритмами, формируемыми на основании предложенного подхода. Алгоритмы при этом будут различными для разных роботов и походок.

При использовании подхода не предъявляются высокие требования к точности позиционирования опорных точек («стоп») робота. Это связано с невозможностью спрогнозировать влияние опорной поверхности, в результате которого точное решение (для определенной точки) может оказаться бесполезным (например, из-за проскальзывания ног по опорной поверхности).

Работа по расширению функциональности предложенного подхода и формируемых на его основе алгоритмов движения ведется автором в настоящее время.

В настоящий момент реализуются походки по ровной опорной поверхности. Ведется разработка генератора случайных поверхностей, позволяющего моделировать неровности местности. Корректировка процесса получения управляющих воздействий будет производиться с учетом неровностей опорной поверхности непосредственно в процессе движения механизма.

Список литературы

1. Зенкевич С.Л., Ющенко А.С. Основы управления манипуляционными роботами: учебник для вузов. 2-е изд., перераб и доп. М.: Изд-во МГТУ им. Н.Э.Баумана, 2004. 480 с.
2. Aristidou Andreas, Lasenby Joan. Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver: Technical Report CUED/F-INFENG/TR-632 Cambridge University, 2009. 74 p.
3. Welman Chris. Inverse kinematics and geometric constraints for articulated figure manipulation: Master Dissertation, Simon Fraser University, Department of Computer Science, Burnaby (Canada), 1993. 84 p.
4. Pechev Alexandre N. Inverse kinematics without matrix inversion // Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena (CA, USA). 2008. P. 2005–2012.
5. Zhao Jianmin, Badler Norman I. Inverse kinematics positioning using nonlinear programming for highly articulated figures // ACM Transactions on Graphics (TOG). 1994. Vol. 13(4). P 313–336.
6. Castellet A., Thomas F. Using interval methods for solving inverse kinematic problems // Computational Methods in Mechanisms, Varna (Bulgaria): NATO Advanced Study Institute. 1997. Vol. 2. P. 135-144.
7. Courty Nicolas, Arnaud Elise. Inverse kinematics using sequential monte carlo methods // Proceedings of the V Conference on Articulated Motion and Deformable Objects, AMDO'08, Mallorca (Spain). Vol. 5098. P. 1–10. 2008.
8. Grochow Keith, Martin Steven L., Hertzmann Aaron, Popovic Zoran. Style-based inverse kinematics // SIGGRAPH '04: ACM Transactions on Graphics, New York (NY, USA). 2004. ACM. P. 522–531.
9. Sumner Robert W., Zwicker Matthias, Gotsman Craig, Popovic Jovan. Mesh-based inverse kinematics // ACM Transactions of Graphics. 2005. Vol. 24(3). P. 488–495.

10. Unzueta Luis, Peinado Manuel, Boulic Ronan, Suescun Ángel. Full-body performance animation with sequential inverse kinematics // *Graph. Models*. 2008. Vol. 70(5). P. 87–104.
11. Wang Li-Chun Tommy, Chen Chih Cheng. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators // *IEEE Transactions on Robotics and Automation*. 1991. Vol. 7(4). Pp. 489–499.
12. Müller-Cajar R., Mukundan R. Triangulation: A new algorithm for inverse kinematics // *Proceedings of the Image and Vision Computing New Zealand*, 2007. Pp. 181–186.
13. Brown Joel, Latombe Jean-Claude, Montgomery Kevin. Real-time knot-tying simulation // *The Visual Computer: International J. of Computer Graphics*, 2004. Vol. 20(2). P. 165–179.
14. Aristidou Andreas, Lasenby Joan. FABRIK: a fast, iterative solver for the inverse kinematics problem // *Graphical Models*, 2011. Vol. 73. P. 243-260.
15. Охоцимский Д.Е., Голубев Ю.Ф. Механика и управление движением автоматического шагающего аппарата. М.: Наука, 1984. 312 с.
16. Kumar V., Gardner J.F. Kinematics of Redundantly Actuated Closed Chains // *IEEE Transactions on Robotics and Automation*, 1990. Vol. 6(2)2. P. 269 - 274.
17. Каргинов Л.А. Синтез управляющих воздействий для приводов исполнительных механизмов шагающих роботов без решения обратной задачи кинематики // *Инженерный журнал: Наука и инновации*. МГТУ им. Н.Э. Баумана. Электрон. журн. 2013. № 4. Режим доступа: <http://engjournal.ru/catalog/machin/hydro/692.html> (дата обращения 25.08.2014).
18. Лапшин В.В. Механика и управление движением шагающих машин. М.: Изд-во МГТУ им. Н.Э. Баумана, 2012. 200 с.
19. Ковальчук А.К., Кулаков Б.Б., Кулаков Д.Б., Семенов С.Е., Яроц В.В. Основы теории исполнительных механизмов шагающих роботов. М.: Рудомино, 2010. 170 с., ил.
20. Ковальчук А.К., Каргинов Л.А., Кулаков Д.Б. и др. Моделирование древовидных исполнительных механизмов шагающих роботов с учетом внешних наложенных связей: свидетельство о регистрации программ для ЭВМ № 2014612547. 28.02.2014.
21. Вукобратович М. Шагающие роботы и антропоморфные механизмы. М.: Мир, 1976. 543 с.