

АВТОМАТИЧЕСКОЕ РЕШЕНИЕ ОБРАТНОЙ ЗАДАЧИ КИНЕМАТИКИ РАЗОМКНУТЫХ МАНИПУЛЯЦИОННЫХ МЕХАНИЗМОВ

Яковлев А.С.

Научный руководитель: Малышенко А.М., д.т.н., профессор
Томский политехнический университет, Институт кибернетики
alex yakovlev90@gmail.com

Разрабатываемые на сегодняшний день манипуляционные роботы имеют огромное значение в развитии современного производства. Среди всей совокупности задач, связанных с кинематикой, наибольший интерес на сегодняшний день имеют обратные задачи. В отличие от прямых задач, решение которых обусловлено необходимостью разработки моделей механизмов манипуляционных роботов, решение обратных задач необходимо для формирования алгоритмов управления подобными механизмами. Основопологающей целью при решении обратных задач является достижение заданного в пространстве положения. Для решения данной проблемы приходится сталкиваться с такими задачами, как планирование положений и движений робота, идентификация его кинематических характеристик. В [1] сформулирован способ описания кинематических цепей роботов, позволяющий идентифицировать их кинематические характеристики в весьма компактной форме. Использование формализованного описания обусловлено удобством представления всех необходимых данных о кинематике механизма в компактной форме, удобной для дальнейшей обработки программными средствами с целью решения задач кинематики.

Исследование алгоритмов решения обратной задачи кинематики и разработка программ для

автоматизированного решения таких задач, имеет огромное значение для как для проектирования новых моделей роботов, так и для разработки алгоритмов управления ими.

В данной работе описывается разработанное ранее универсальное программное средство для формирования траекторий движения манипуляционных роботов. Решение подобных задач играет огромную роль в компьютерной анимации, а также моделировании алгоритмов управления различными механизмами и прочими шарнирными объектами.

Способы решения обратной задачи кинематики представляют собой методы для вычисления положения через оценивание в отдельности каждой индивидуальной степени свободы с целью решения поставленной задачи с заданными ограничениями. Как уже было упомянуто целью решения обратной задачи является правильное планирование движений робота для достижения заданного положения. Традиционно, при отсутствии препятствий выбирают кратчайшую траекторию. Однако, даже при отсутствии каких-либо ограничений для формирования траектории движения робота, возникают сложности при решении алгоритмов достижения цели.

Работа разработанного программного приложения представлена на рисунках 1-2.

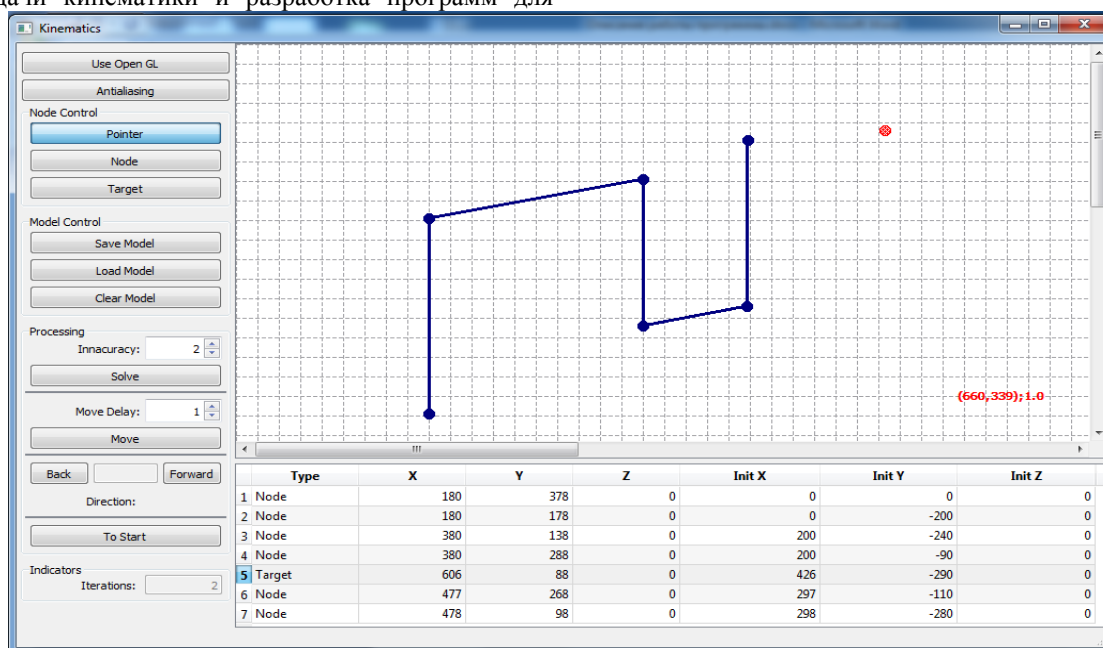


Рисунок 1 - Интерфейс программы – Исходное положение механизма

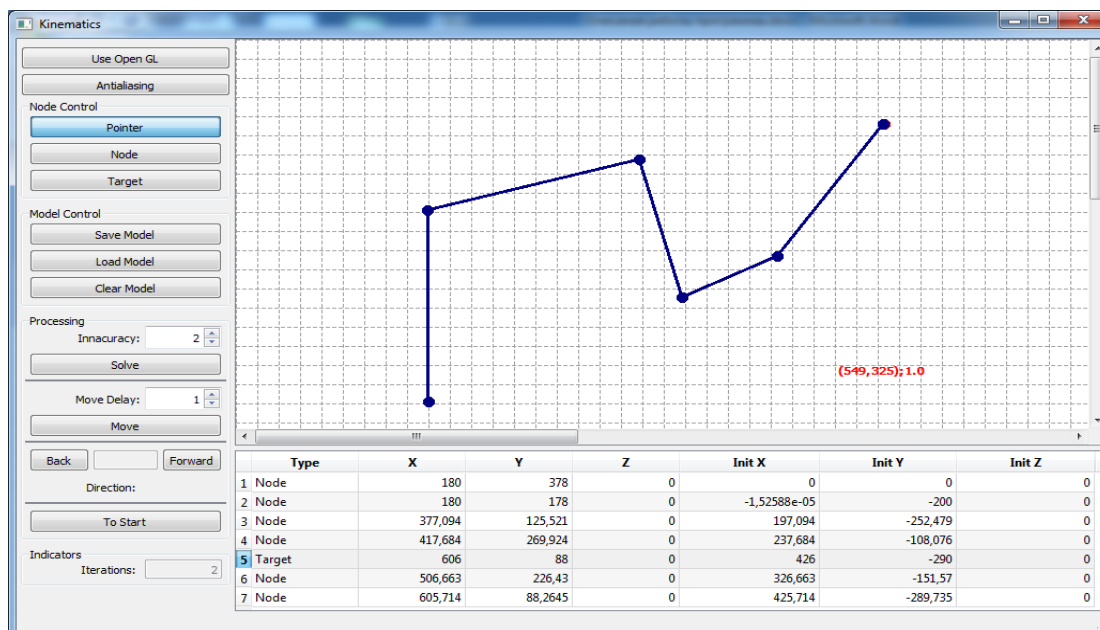


Рисунок 2 - Интерфейс программы – Положение механизма в целевой точке

На рисунках 1 и 2 представлены изначальное и конечное положения механизма. Для поиска решения обратной задачи необходимо нарисовать кинематическую схему механизма (звенья механизма соединяются вращательными сочленениями) и отметить целевую точку. Далее алгоритм определяет оптимальную (кратчайшую) траекторию достижения целевой точки и выполняет анимированное движение.

Для поиска оптимального решения в процессе разработки программного обеспечения было проанализировано множество способов решения обратных задач кинематики, таких как Jacobian Transpose, Damped Least Squares (DLS), алгоритм пошагового циклического координатного спуска (Cyclic Coordinate Descent (CCD) algorithm), а также серия методов, в основе которых лежит алгоритм Ньютона.

Основная проблема решения обратной задачи связана с решением системы нелинейных уравнений. С целью избегания трудоемкого поиска решения системы нелинейных уравнений было принято решение использовать алгоритмический подход прямого и обратного движения, так называемый метод FABRIK [2]. Используемый метод является итерационным и использует прямые и обратные итерационные смещения характеристических точек, определяя позицию каждого сочленения через положение на линии.

Программное приложение было реализовано на языке C++ с использованием кроссплатформенного инструментария для создания графического интерфейса QT. Программирование с использованием библиотек QT предоставляет мощные расширения для языка C++. К тому же программная среда разработки распространяется на условиях свободной лицензии GPL 3.0 или LGPL 2.1, что является весьма

значительным преимуществом, повлиявшим на выбор инструментария.

Набор инструментов разработанного программного средства включает в себя следующие управляющие модули:

1. Node control. Данный модуль предназначен для построения кинематической схемы механизма в виде набора узлов, представляющих собой вращательные сочленения.

2. Model control. Вспомогательный модуль, обеспечивающий возможность сохранения и загрузки построенной модели робота.

3. Processing. Здесь задаются параметры вычислений.

Вывод

Таким образом, было проанализировано множество подходов и методик решения поставленной задачи, выбран и программно реализован наиболее подходящий метод. Уже на данном этапе программное средство позволяет применять данный подход для планирования положений и движений произвольного робота с разомкнутой структурой. А визуализация движений манипулятора в значительной степени экономит время на отладку программного обеспечения для управления роботом.

Литература

1. A. Yakovlev, Obtaining manipulators kinematic models from its formalised description// 2015 International Siberian Conference on Control and Communications (SIBCON), IEEE Computer Society, May 21–23, 2015, Omsk, pp. 38–42.
2. A. Aristidou, J. Lasenby, FABRIK: A fast, iterative solver for the Inverse Kinematics problem// Graphical Models vol. 73, 2011, pp. 243–260.