Исследование скорости работы алгоритма решения обратной задачи кинематики манипулятора, основанном на геометрическом представлении манипулятора и представлении Денавита-Хартенберга, на контроллере Atmega 2560

1. Аннотация

//это не то. В целом про проблему, а не мою проблему.

На данный момент ведется проектирование пяти осевого ременного манипулятора, который будет установлен на борт телеуправляемого катамарана. Управление манипулятором происходит в декартовом пространстве, значит необходимо преобразовывать декартовые координаты, заданные пользователем, в обобщенные координаты манипулятора. Нужно узнать будет ли пользователей сильно бесить задержки в управлении. Задержки могут появляться из-за низкой частоты МК, отсутствия аппаратной обработки плавающей точки и долгого времени вычисления обратных тригонометрических функций. Все эти операции присутствуют в алгоритмах управления манипулятором.

Проведем анализ времени работы некоторых алгоритмов решения обратной задачи кинематики. Так же стоит учитывать, что при реальной работе манипулятора МК необходимо так же считывать данные, обрабатывать скорости вращения звеньев, а также предотвращать столкновение манипулятора с катамараном.

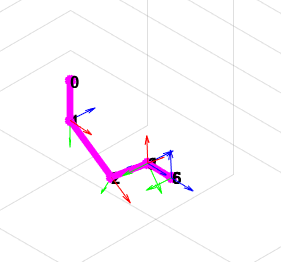
//Я недавно убедил заказчика, что ардуино это плохо и реализацию будет на stm под cortex m4. Так что это статья – это часть ВКР, которую я вырезал за ненадобностью.

1. Обзор эксперимента(задачи)

Отладка алгоритмов управления манипулятора в системе компьютерного моделирования MATLab до его непосредственной сборки. Так как разработка конструкции манипулятора и программного обеспечения ведутся параллельно, то необходимо разработать модель манипулятора и протестировать на ней программные решения.

Для визуализации был применен пакет my-matlab-robotics-toolbox автора ///. Данный пакет позволяет строить кинематическую модель манипулятора, основываясь на обобщенных координатах и длинах звеньев. Алгоритмы управление манипулятора разработан на основе геометрической интерпретации длин звеньев и обобщенных координат, а также на осневе представления ДХ.

Картинка с манипулятором



//Привести UML диаграмму управления манипулятором и сказать что при разных методах решения ОЗК все остальное работает за то же время

Манипулятор спроектирован так, что 4 оси управляются шаговыми двигателями, крен кисти осуществляется серводвигателем. Так же серводвигатель приводит в работу захват манипулятора. Управляющие сигналы на МК приходят с UARTа. Микроконтроллеру необходимо принимать изменяющиеся значения скоростей перемещения, по ним рассчитывать значение обобщенный координат и отправлять управляющие сигналы шаговым двигателям и сервоприводам, следить за правильностью исполнения команд и отправлять текущие обобщенные координаты обратно в UART порт. Так же МК необходимо предотвращать столкновения манипулятора с самим с собой и с корпусом катамарана. Столкновение с самим с собой предотвращаются наложением ограничений на обобщённые координаты. Для предотвращения столкновений манипулятора с корпусом катамарана МК так же необходимо одновременно решать несколько ПЗК. Решение об изменении обобщенных координат может быть принято только после прохождения всех проверок на столкновения.

Так как время обработки алгоритмов управления двигателям, предотвращения столкновений и работы с последовательным портом, при решении обратной задачи двумя методами, будет одинаковое, то рассмотрим подробнее время выполнения разных методов решения обратной задачи кинематики.

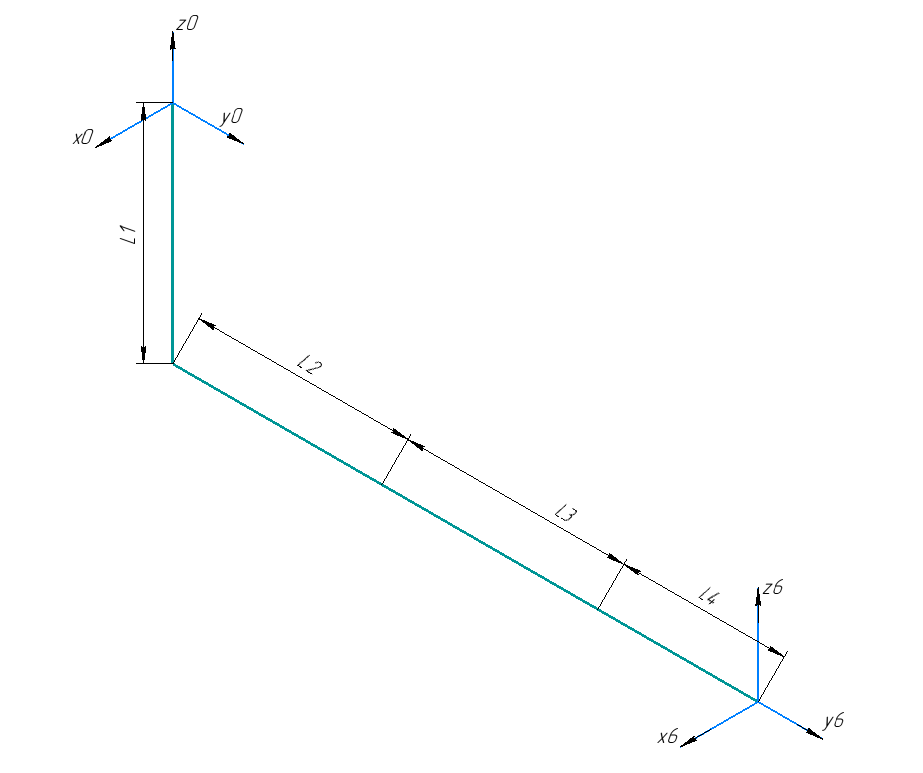
В качестве оценки скорости будем последовательно передовать алгоритму ОЗК координаты квадрата и запоминать время выполнения, что в последствии позволит сравнить время отклика.

Экспериментальным методом установлено, что время, затрачиваемие на каждом шаге алгоритмов управления привадами, предотвращения столкновений и обмен данными по последовательному порту составляет тратата секунд. Для оценки задержки будем прибавлять данное значение к каждому вызову ОЗК.

//Привести таблицу значений других алгоритмов.

1. Решение через ДХ представдение

Представим манипулятор как набор связанных систем координат, который позполит составить таблице значений ДХ представления и найти матрицу T.



Данное представление

Таблица значений

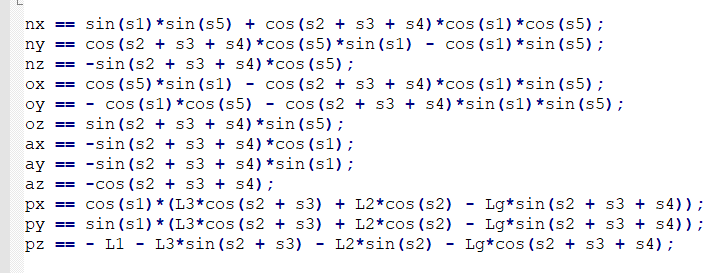
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| α | π/2 | 0 | 0 | -π/2 | 0 | π/2 |
| θ | -π/2 | 0 | 0 | -π/2 | 0 | π/2 |
| a | 0 | L2 | L3 | 0 | 0 | 0 |
| d | -L1 | 0 | 0 | 0 | L4 | 0 |

Шестое сочленение было введено для удобства измерения декартовых координат в матрице T6. В нулевом состоянии оси шестой системы координат сонаправленны с осями базовой системы координат.

Решая матричное уравнение

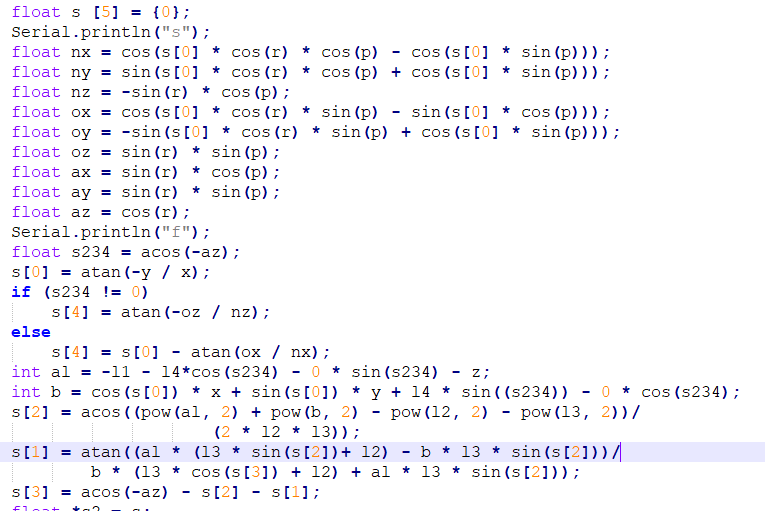


Найдем решение прямой задачи



Найдем решение обратной задачи кинематики и проверим скорость работы алгоритма

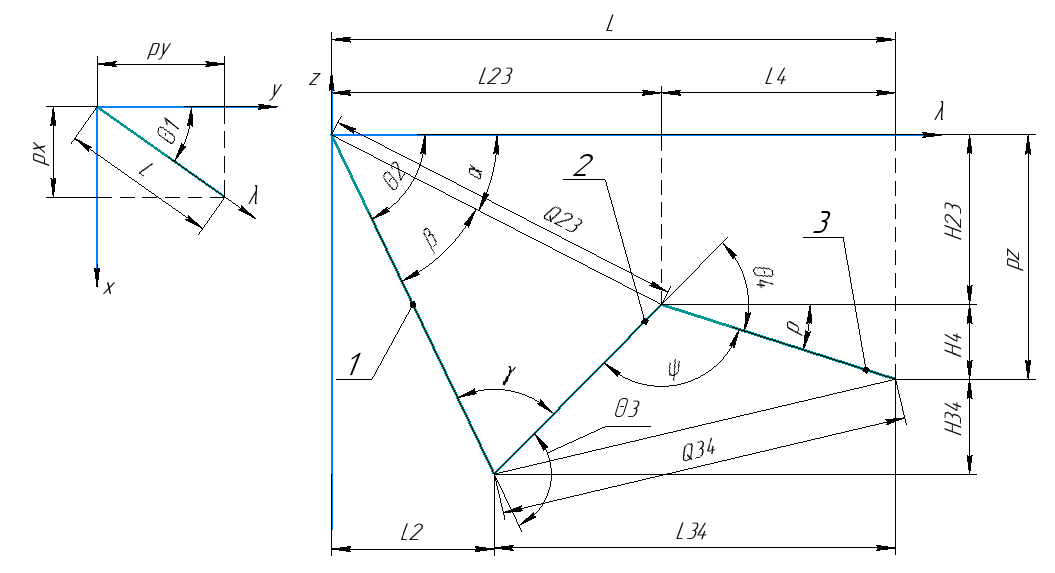
//тут решение и вывод по времени работы. Еще я добавил триггер, чтобы посмотреть сколько eul2rot будет работать. Возможно основная разница во времени работы будет именно из-за перевода углов эйлера в матрицу поворота



1. Решение через геометрическое представление

Геометрический метод решения

Для сокращения вычислений прибегнем к геометрическому способу решения ОЗК.

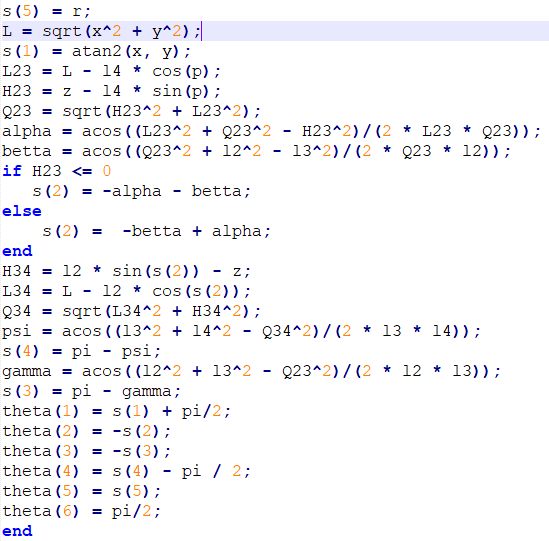


На рисунке слева представлена проекция манипулятора на плоскость xOy. Введем новую ось λ так, что звенья манипулятора лежат в плоскости zOλ. На рисунке справа представлено расположение манипулятора в плоскости zOλ.

Использование геометрического подхода позволило находить угол сигма 5 без вычислений, сократило количество уравнений и избавило от необходимости перевода углов Эйлера в матрицу поворота.

//Может я еще придумаю как уменьшить время вызова функции.

Решение:



Наверное, можно решение чуть подробнее в обоих случаях написать

//может следует посчитать теоретическое время выполнения и сравнить его с практическим.

//если все будет слишком плохо, то я попробую перейти в поле целых чисел и может переделать тригонометрические функции на ряд тейлора

1. Ход эксперимента.

Скорее всего этот раздел уже лишний, так как ход я буду описывать в каждом разделе отдельно

1. Выводы

Скорее всего быстродействия МК не хватит чтобы решить поставленную задачу, но нужно экспериментальное подтверждение. Нарисую какой-нибудь красивый график с временем работы, приду к выводу что требуется замена МК на cortex m4.

Предыдущий вывод бесполезен, так как я победил, и неделю назад разработка переехала на m4, что можно и отразить в статье и в НИР. По ардуино уже много наработано, так что оптимизацию под ардуино я перемещаю в статью. К тому же она может пригодиться в бедующем.

1. Список литературы