Санкт-Петербургский политехнический университет Петра Великого

Институт машиностроения, материалов и транспорта

Высшая школа автоматизации и робототехники

**ОТЧЁТ ПО НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ**

**Разработка системы управления для пятизвенного манипулятора c ременными передачами на шарниры**

Выполнил

студент гр. 3331506/60401 <*подпись*> Д.Д. Сидоренко

Руководитель

старший преподаватель <*подпись*> А.С. Габриель

Научный консультант <*подпись*> И.А. Васильев

доцент, к.т.н.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 г.

Санкт-Петербург

2020

**Реферат**

На 33 с., 9 рисунков, 3 таблицы, 3 приложения

РОБОТОТЕХНИКА, МАНИПУЛЯТОР, ОБРАТНАЯ ЗАДАЧИ КИНЕМАТИКИ. БОРТОВАЯ СИСТЕМА УПРАВЛЕНИЯ, СИСТЕМА УПРАВЛЕНИЯ МАНИПУЛЯТОРОМ, ФОРМАЛИЗМ ДЕНАВИТА – ХАРТЕНБЕРГА, ОБОБЩЕННЫЕ КООРДИНАТЫ.

В настоящей работе проводиться описание разработки бортовой программы управления для подвесного пятиосевого манипулятора с ременными приводами на шарнирах. В рамках исследование произведено описание кинематики манипулятора с использованием формализма Денавита – Хартенберга и с использованием геометрического представления. Так же в рамках работы получены решения обратной задачи кинематики с использованием методов, описанных выше. Сравнение полученных методов производилось по времени решения ОЗК для одного набора входных координат.

Программа управления на вход получает вектор скоростей захватного устройства в декартовом пространстве координат, а на выходе формирует управляющие сигналы приводов манипулятора. Так же в рамках работы разработана структура системы управления манипулятором, включающая в себя защиту от столкновений манипулятора со средой.

Полученные результаты решений и сравнения ОЗК, структура бортовой системы управления, описание возможных положений манипулятора при столкновении с телеуправляемым катамараном исследовательским, на котором базируется манипулятор, будут применимы для реализации программы управления непосредственно на борту робота.

**Содержание**

[Введение 5](#_Toc30411698)

[1 Постановка задачи 6](#_Toc30411699)

[1.1 Кинематическая схема манипулятора 7](#_Toc30411700)

[2 Обзор готовых решений 9](#_Toc30411701)

[2.1 Matlab Robotic System Toolbox 9](#_Toc30411702)

[2.2 ROS MoveIt 10](#_Toc30411703)

[3 Разработка алгоритма управления 11](#_Toc30411704)

[3.1 Решение через ДХ представление 12](#_Toc30411705)

[3.2 ОЗК методом ДХ 16](#_Toc30411706)

[3.3 Геометрический метод решения ОЗК 19](#_Toc30411707)

[3.4 Защита от столкновений 21](#_Toc30411708)

[4 Расчет углов шаговых двигателей и сервоприводов 24](#_Toc30411709)

[4.1 Сравнение решений 27](#_Toc30411710)

[Вывод 28](#_Toc30411711)

[Список литературы 29](#_Toc30411712)

[Приложение 1 31](#_Toc30411713)

[Приложение 2 32](#_Toc30411714)

[Приложение 3 33](#_Toc30411715)

**Перечень сокращений и условных обозначений**

В настоящем отчете по НИР применяются следующие обозначения и сокращения.

ЗУ – Захватное устройство

ДХ – Денавита – Хартенберга

ПЗК – прямая задача кинематики

ОЗК – обратная задача кинематики

ШД – шаговый двигатель

МР – мотор-редуктор

ЗР – задающие рукоятки

Введение

В рамках гранта Фонда содействия инновациям командой резидентов «ФАБЛаб Политех» был разработан телеуправляемый исследовательский катамаран «Кадет-М». Одной из полезных нагрузок катамарана, согласно техническому заданию, является манипулятор, предназначенный для сбора плавающих объектов с поверхности воды. Так же в конструкцию манипулятора была заложена возможность обслуживания и замены аккумуляторов на сигнальных буях. [1]

Манипулятор предусматривает управление только в ручном режиме с пульта корабля сопровождения. Для эргономичного управления движением манипулятора необходимо преобразовывать движение задающих рукояток на пульте управления в изменение декартовых координат рабочего инструмента манипулятора. Что приводит к необходимости преобразования декартовых координат захватного устройства в обобщенные координат манипулятора.

В рамках данной работы разработано математическое описание манипулятора, которое в дальнейшем будет применимо для реализации бортовой системы управления. Так же сравнение нескольких подходов к разработке позволили выбрать наиболее оптимальный по удобству управления оператором вариант.

Разработанная система управления учитывает особенности передачи вращательного момента на звенья манипулятора через ременный привод. Данная особенность не позволяет напрямую преобразовывать угловые координаты звеньев в координаты приводов. [2, 4]

# Постановка задачи

Разработать систему управления пятизвенным манипулятором, установленном на катамаране, с ременными передачами в приводах. Система управления должна давать возможность управление от задающих рукояток, как в декартовых, так и в обобщенных координатах, предотвращать столкновение манипулятора с корпусом катамарана.

Оператор с помощью задающих рукояток, установленных на пульте управления, отправляет сигналы на микроконтроллер, который с помощью алгоритма управления принимает решение об изменении обобщённых координат. Оператор управляет с манипулятором посредством визуального контроля с помощью бортовой камеры, так же ему передаются сообщения об ошибках в управлении и невозможности достижения заданного управления. Оператор имеет возможность изменять декартовые координаты рабочего инструмента манипулятора, углы крена и тангажа кисти, а также управлять одноосевым рабочим устройством.

Для понимания конструкционных особенностей манипулятора ниже приведена кинематическая схема конструкции

## Кинематическая схема манипулятора

На рисунке 1 приведена кинематическая схема манипулятора. [10]



Рисунок 1

На валу мотор-редуктора [8, 7] 1 (МР1) установлен шкив 1. С обратной стороны установлен инкрементальный энкодер. МР1 приводит в движение ось колонны манипулятора. к корпусу колонны присоединены мотор-редуктора 2, 3 и 4.

МР4 через неразъемную муфту со шпоночным пазом передает вращение на вал IV. На валу IV жестко закреплены шкив 5 и подвижная головка инкрементального энкодера. Корпус энкодера в свою очередь жестко соединен с корпусом колонны. Так же вал IV соединён с корпусом колонны через шариковый опорный подшипник. Вращение со шкива 5 передается на шкив 6, который в свою очередь жестко закреплен на валу V с помощью шпоночного соединения. Вал V жестко соединен с корпусом плеча. Вышеописанная кинематическая цепь приводит в движение плече манипулятора.

Конструкция вала II, соединенного с МР2, аналогична валу VI. Вращательное движение со шкива 3 передается на шкив ступичный 7, который закреплен на валу V через пару шариковых опорных подшипников. Далее вращательное движение со ступичного шкива 7 передается на шкив 10, который жестко соединен с валом VI с помощью шпоночного соединения. Вал VI в свою очередь жестко соединен с корпусом предплечья. Вышеописанная кинематическая цепь приводит в движение предплечье манипулятора. Особенность поворота шарнира предплечья состоит в том, что обобщенные координаты поворота предплечья зависят не только от координат МР2, но и от обобщенных координат плеча.

Конструкция вала III аналогична ранее рассмотренным. Передача момента на кисть манипулятора осуществляется аналогично с передачей момента на предплечье, за одним исключение, обобщенные координаты крена кисти зависят не от одной родительской координаты, а от двух.

Согласованное изменением координат МР1, МР2, МР3 обеспечивают изменение декартовых координат рабочего инструмента, а изменение координат МР4 и Д1 изменение ориентации устройства.

Данная кинематическая схема создает особенности расчета управляющих сигналов мотор-редукторов в зависимости от положения ременных приводов в пространстве. Обобщенные координаты углов между плечом и предплечьем, а также между предплечьем и кистью зависят не только от координат управляющих приводов, но и то обобщенных координат родительских звеньев. Для корректного расчета управляющих сигналов МР3 и МР4 необходимо добавить блок расчета новых управляющих сигналов.

## Электрическая функциональная схема манипулятора

Для понимания особенностей проектирования системы управления манипулятором на рисунке 2 приведена электрическая функциональная схема управляющего контроллера манипулятора. На схеме опущены преобразователи уровней управляющих сигналов приводов и датчиков. Также на схеме опущены преобразователи уровней UART и RS485. Для удобочитаемости схема питание узлов и гальванические развязки также не показаны.

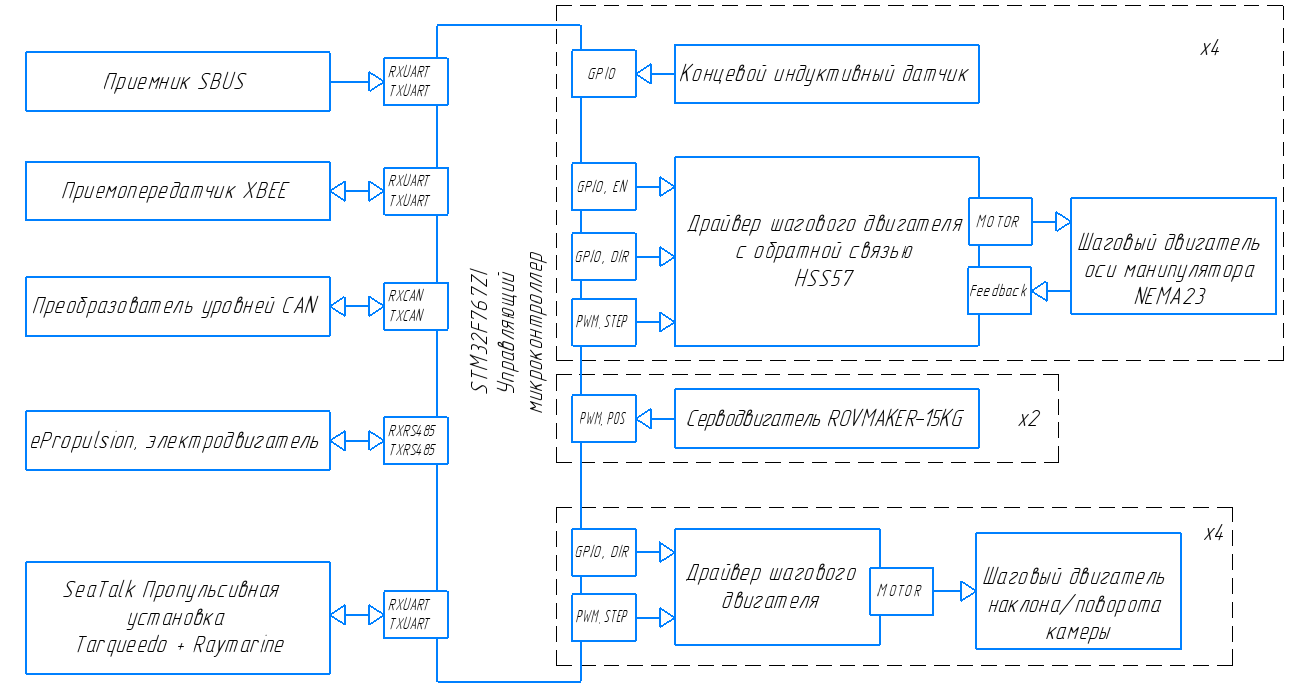


Рисунок 2 - Электронная функциональная схема манипулятора

В данной работе подробно рассматривается управление шаговыми двигателями осей манипулятора и серводвигателями, обеспечивающими крен ЗУ манипулятора и рабочее движение ЗУ манипулятора.

Как видно из рисунка 2 для управления шаговым двигателем необходимо использовать 2 пина ввода-вывода для задания направления вращения и активации драйвера. А также для установки скорости вращения используется аппаратная генерация ШИМ сигнала и отдельный вывод таймера микроконтроллера для каждого двигателя [3, 4]. Требуется использовать отдельный таймер для каждого ШД, так как управление происходит благодаря изменению частоты ШИМ сигнала. Каждый передний фронт ШИМ сигнала сообщает драйверу, что необходимо повернуть ротор ШД на один микрошаг в сторону, соответствующую направлению, заданному пином DIR. При этом для корректной работы микроконтроллера драйвера производитель рекомендует поддерживать скважность сигнала близкой к 50 процентам. Драйвер шагового двигателя HSS57 осуществляет обратную связь с шаговым двигателем с помощью инкрементального энкодера, установленного на оси двигателя. Данная функция драйвера обеспечивает отработку команд управления и дает защиту от пропуска шагов.

Для управления серводвигателем необходим один управляющий сигнал, задающий положение ротора привода. В корпусе серводвигателя установлен абсолютный энкодер, обеспечивающий обратную связь по положению. Управляющий сигнал представляет из себя ШИМ сигнал с частотой 600Гц. Положение ротора задается путем изменения скважности ШИМ сигнала, которая может меняться от 100мкс до 300мкс. Угол поворота серводвигателя ограничен 120 градусами, что соответствует скважности в 300мкс. Так как управление серводвигателя построено на изменении скважности ШИМ сигнала, то целесообразно будет выделить один таймер с двумя внешними каналами для генерации управляющих сигналов [5].

# Обзор готовых решений

## Matlab Robotic System Toolbox

Первая кинематическая модель манипулятора была создана с применением пакета Matlab Robotic System Toolbox [11, 12]. Данный пакет применим не только к манипуляционным роботам, а также и к мобильным. Он имеет встроенные решения SLAM и может интегрироваться с ROS. Инструменты пакета для управления манипуляционными роботами:

* Построение древовидной структуры манипуляционных систем
* Построение кинематической и динамической моделей роботов.
* Решение обратной задачи кинематики итерационными алгоритмами Левенберга – Марквардта и Бройана –Флетчера – Гольдфарба – Шанно.
* Алгоритмы решения

Модули управления манипулятором в Matlab после преобразования скрипта в код на C++ заняли размер 8 МБ, что было недопустимо для размера памяти микроконтроллера всего 256 кбайт. Так же использование методов Robotic System Toolbox не позволяло задать углы крена и тангажа кисти, и при каждом вызове решения необходимо переводить углы Эйлера в матрицу поворота, что отрицательным образом сказывается на быстродействии системы.

Параметры модели, созданной с помощью данного пакета, приведены в приложении 1.

## ROS MoveIt

Данное расширение для ROS разработано специально для управления манипуляторами. Функционал расширения:

* Генератор траекторий, избегающий локальны минимумов
* Анализ взаимодействия захватного устройства со средой
* Решение обратной задачи кинематики, в том числе и для манипуляторов с большим числом степеней свободы, больше шести
* Исполнение команд на контроллерах приводов
* Подключение к датчикам глубины и камерам RGBZ.
* Построение траекторий в изменяющейся среде, благодаря обработки облака точек, поступающего с RGBZ камер.

Среда разработки Arduino, в связи с ограничениями разработчиками ROS и возможностями микроконтроллера поддерживает только урезанную версию данной системы – rosserial\_client, которая не позволяет воспользоваться возможностями расширения.

В дальнейшем планируется поменять контроллер манипулятора на одноплатный компьютер с поддержкой обработки сигналов, что позволит расширить возможности манипулятора и использовать средства технического зрения для захвата объектов с детерминированными параметрами в автоматическом режима, а также позволит стабилизировать положение захватного устройства при волнении. [5]

# Разработка алгоритма управления

На рисунке 2 представлена блок-схема алгоритма управления манипулятора оператором от ЗР.

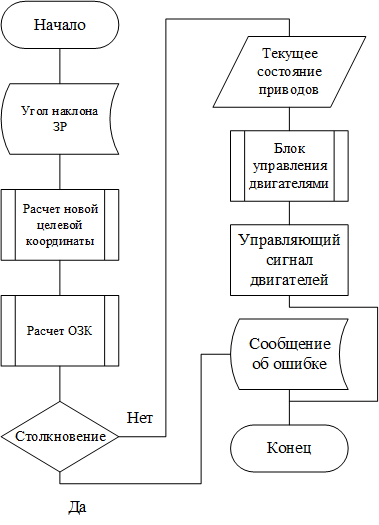


Рисунок 2

В блоке расчета новой целевой координаты происходит перевод заданных оператором с помочью ЗР скоростей в текущее положение целевой точки. В блоке проверки на столкновение происходит проверка не только на столкновение манипулятора с катамараном, но и проверка на самопересечение, а также проверка достижения целевой координаты. В данной работе внимание уделяется блоку расчёта ОЗК, как самому затратному по вычислительной мощности. Существует несколько способов решения ОЗК, рассмотрим один из них далее.

## Решение через ДХ представление

Одним из способов задания системы координат для геометрии манипулятора для решения прямой и обратной задач кинематики является представление Денавита – Хартенберга, которое кинематически связывает сочленения звеньев. Описание связей происходит с помощью матриц 4х4 и однородных преобразований. Начальная система координат связана с основанием манипулятора и имеет нулевой номер. Ниже приведен алгоритм назначения и описание систем координат звеньев, а также нахождения параметров ДХ для каждого сочленения 1 – 5.

Оси следует направить параллельно осям дочернего вращательного сочленения.

Началом i-й системы координат служит точка пересечения осей  и , либо точка пересечения нормали к тем же осям и оси .

Ось задается следующим соотношением , если оси  и  параллельны, то ось задается вдоль общей нормали к этим осям.

Ось дополняет уже заданные оси в каждом сочленении до правосторонней системы координат.

Параметры ДХ:

Расстояние от точки начало координат системы координат до точки пересечения осей и является параметром .

Расстояние от пересечения оси c общей нормалю осей и до начала i-й система координат является параметром .

Угол поворота от оси до оси является параметром .

Угол поворота от оси до оси относительно оси является параметром .

Ниже на рисунке 3 приведена кинематическая схема выбранными системами координат системами координат. Примем данное положение манипулятора за нулевое.

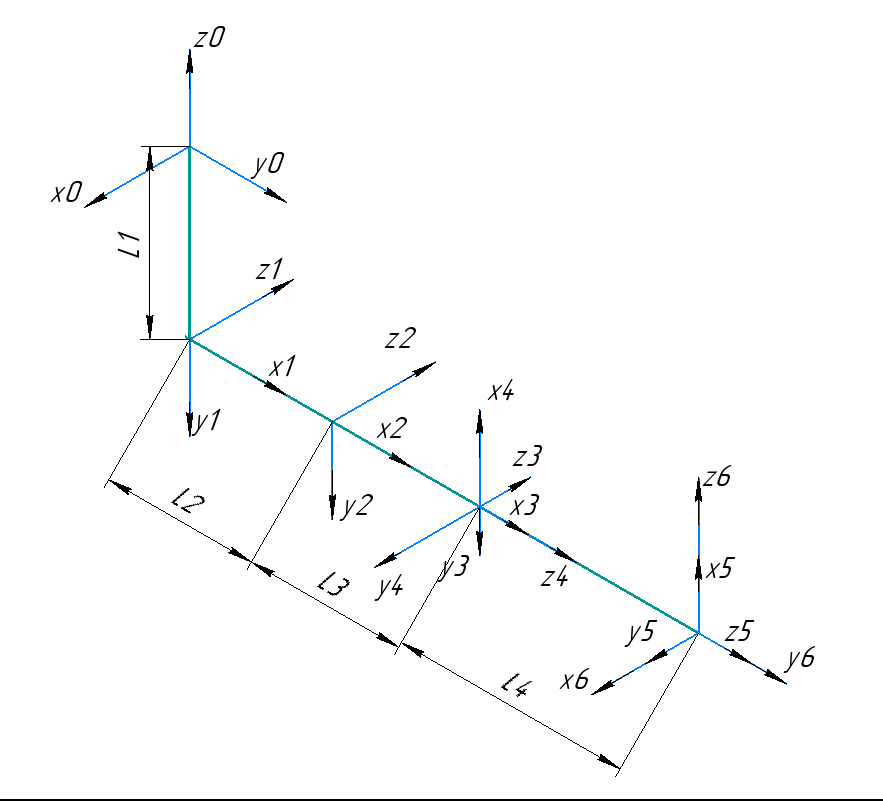


Рисунок 3

В таблице 1 представлены параметры ДХ для данной конструкции манипулятора.

Таблица 1 - Значение параметров ДХ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 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 |

Здесь . выступает в роле присоединительной координаты, а значения, указанные в таблицы, являются нулевыми.

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

Связь системы координат последнего, шестого, звена манипулятора с базовой системой координат осуществляется благодаря матрице однородного преобразования T6, которая имеет вид:

|  |  |
| --- | --- |
|  | (1) |

Последовательное перемножение матриц поворота и переноса, согласно методу ДХ, дает матрицу

|  |  |
| --- | --- |
|  | (2) |

Матрица осуществляет переход от системы координат к i-й

Матрица имеет вид

|  |  |
| --- | --- |
|  | (3) |

где , , – элементы матрицы поворота R, а – вектор положения координат захватного устройства относительно базовой системы координат

Так как 6 сочленение мнимое, то матрица известна заранее

|  |  |
| --- | --- |
|  | (4) |

Получившаяся система уравнений имеет вид:

|  |  |
| --- | --- |
|  | (5) |
|  | (6) |
|  | (7) |
|  | (8) |
|  | (9) |
|  | (10) |
|  | (11) |
|  | (12) |
|  | (13) |
|  | (14) |
|  | (15) |
|  | (16) |

Как видно из системы уравнений выше, параметры шестого сочленения никак не учувствуют в конечном расчете и не усложняют систему.

Данная система уравнение позволяет решать ПЗК – нахождение декартовых координат захватного устройства в зависимости от обобщенных координат манипулятора. Так же нахождение элементов px, py, pz для матриц однородного преобразования T6, T4 и T3 позволяет избегать столкновений с дном катамарана.

Решение системы уравнений относительно обобщенных координат манипулятора позволит решить ОЗК, что является одним из блоков системы управления манипулятором в целом.

## ОЗК используя формализм ДХ [9]

Решим получившуюся систему уравнений и найдем зависимость обобщенных координат от декартовых координат. Для удобства будем обозначать как , косинус суммы углов как . Для синусов введем аналогичные обозначения.

Из уравнения (13) получим

|  |  |
| --- | --- |
|  | (17) |

Далее из формул (14) и (14) получим

|  |  |
| --- | --- |
|  | (18) |

Если , то из формул (7) и (10) следует:

|  |  |
| --- | --- |
|  | (19) |

Если , то и тогда из формул (4) и (8) следует

|  |  |
| --- | --- |
|  | (20) |

Знак появляется из-за периодичности тригонометрических функций. Так как

Из формул (14) и (15) можно показать, что

|  |  |
| --- | --- |
|  | (21) |

Далее из формул (16) и (21) можно увидеть, что

|  |  |
| --- | --- |
|  | (22) |
|  | (23) |

Возведем выражения (22) и (23) в квадрат и сложим друг с другом

|  |  |
| --- | --- |
|  | (24) |

Из выражения (24) найдем угол

|  |  |
| --- | --- |
|  | (25) |

Разложим сумму косинусов и синусов в выражениях (22) и (23)

|  |  |
| --- | --- |
|  | (26) |
|  | (27) |

Далее умножим выражения (26) и (27) на и и сложим их друг с другом

|  |  |
| --- | --- |
|  | (28) |

Далее умножим выражения (26) и (27) на и и сложим их друг с другом

|  |  |
| --- | --- |
|  | (29) |

Умножим выражения (28) и (29) на и и сложим их друг с другом, затем умножим те же выражения на и и так же сложим

|  |  |
| --- | --- |
|  | (30) |
|  | (31) |

Из выражений (30) и (31) найдем угол

|  |  |
| --- | --- |
|  | (32) |

И в конце из выражений (17), (25) и (32) найдем угол

|  |  |
| --- | --- |
|  | (33) |

ОЗК решена. Сводная таблица решения приведена ниже

Таблица 2 - Сводная таблица решения ОЗК

|  |  |
| --- | --- |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| Дополнительные параметры | |
|  |  |
|  |  |

Решение ОЗК с помощью представления ДХ для данной системы управление будет избыточным, так как для ориентации ЗУ в пространстве оператор с помощью рукояток управления задает углы крена и тангажа, а представление ДХ предполагает задание матрицы поворота. Чтобы использовать данное решение необходимо будет переводить вторую систему углов Эйлера (крен, тангаж, рысканье) в матрицу поворота, что увеличит число операций и негативно скажется на времени вычислений. Применим альтернативный метод решения ОЗК с помощью геометрической интерпретации кинематики сочленений манипулятора.

## Геометрический метод решения ОЗК

Для данного метода необходимо ввести несколько новых углов, которые показаны на рисунке 4 ниже. Так же перенесем точку начала базовой системы координат в сочленение первого и второго звеньев, что позволит исключить длину первого звена из расчетов.

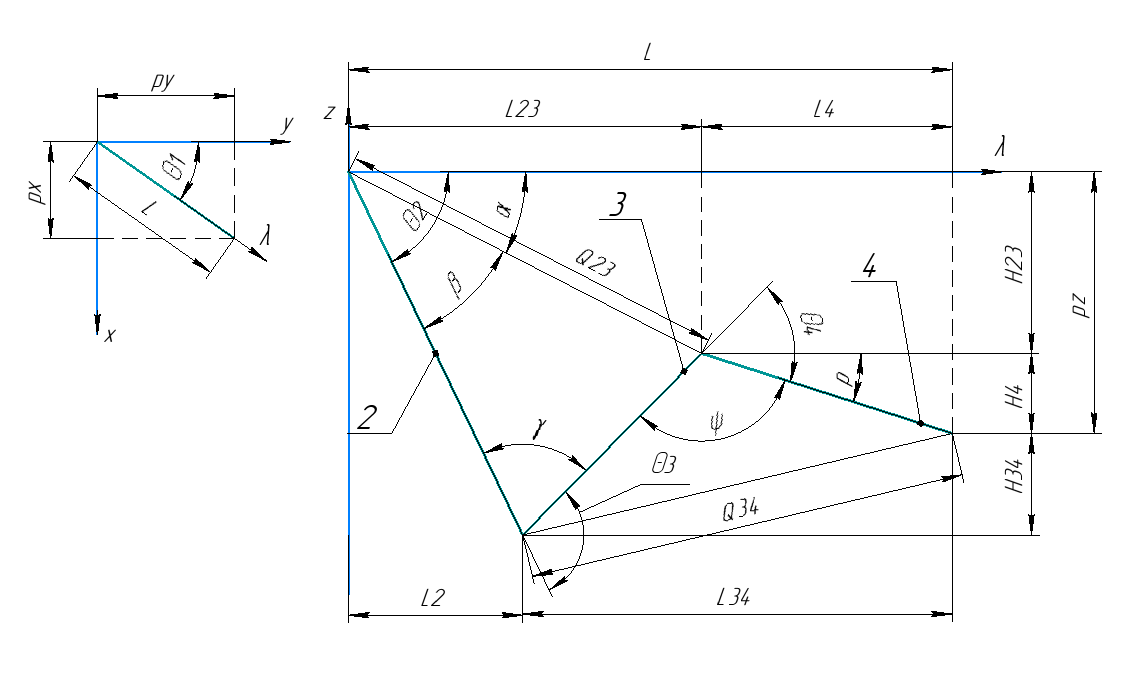


Рисунок 4

На рисунке 4 слева представлена проекция манипулятора на плоскость xOy. Введем новую ось λ так, что звенья манипулятора лежат в плоскости zOλ. На рисунке 4 справа представлено расположение манипулятора в плоскости zOλ. Так же на рисунке 4 обозначены длины отрезков, которые потребуются для дальнейшего решения.

Использование геометрического подхода позволит находить угол без дополнительных вычислений, сократит количество уравнений и избавит от необходимости перевода углов Эйлера в матрицу поворота.

Оператор задает следующие приращения координат

Таблица 3 - Координаты

|  |  |
| --- | --- |
| Обозначение | Название |
| *x* | Значение координаты x, px на чертеже |
| *y* | Значение координаты y, py на чертеже |
| *z* | Значение координаты z, pz на чертеже |
| *r* | Значение угла крена (roll) |
| *p* | Значение угла тангажа (pitch) |

Так как за расчет скоростей приводов отвечает другой блок системы управления, то для данного раздела примем значения из таблицы за абсолютные. Для расчетов все угловые значения выражаются в радианах. Размерность линейных размеров для данной задачи не важна, так как легко масштабируется.

Значение угла можно найти без каких-либо расчетов.

|  |  |
| --- | --- |
|  | (34) |

Далее найдем длину проекции манипулятора на плоскость *xOy*

|  |  |
| --- | --- |
|  | (35) |

Угол можно найти следующим образом

|  |  |
| --- | --- |
|  | (36) |

Далее найдем дополнительные значения и по теореме косинусов найдем значение угла

|  |  |
| --- | --- |
|  | (37) |
|  | (38) |
|  | (39) |
|  | (40) |
|  | (41) |
|  | (42) |
|  | (43) |

Далее найдем значение углов и по такому же принципу.

|  |  |
| --- | --- |
|  | (44) |
|  | (45) |
|  | (46) |
|  | (47) |
|  | (48) |
|  | (49) |
|  | (50) |

## Защита от столкновений

Как видно из диаграммы управления после расчета новых обобщенных координат необходимо провести проверку на возможность столкновений с элементами конструкции катамарана. Ниже на рисунке 5 приведены схемы столкновения манипулятора с дном катамарана.

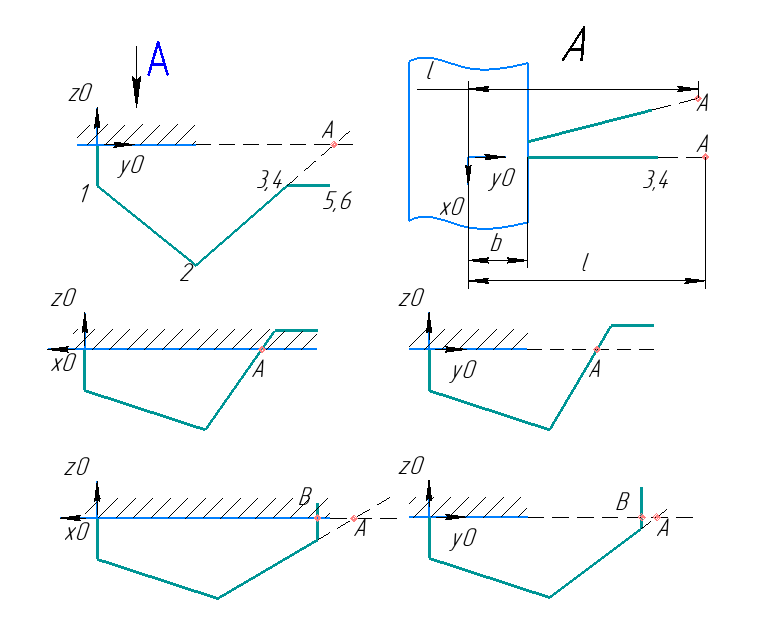


Рисунок 5

Обозначим точкой *А* точку пересечения плоскости *xOy* и прямой, проходящей через начала систем координат 2 и 3, точкой *B* точку пересечения той же плоскости и прямой, проходящей через начала координат систем 3 и 5. Для предотвращения такого рода столкновений необходимо вычислять координаты данных точек и запрещать изменение обобщённых координат, если длина *l* меньше конструктивного размера *b*. Так же необходимо добавить зону безопасности от столкновений, так как все координаты пересечений считаются по центру звеньев, а реальные звенья манипулятора обладают толщиной.

Декартовые координаты сочленения для синтеза уравнений прямых можно взять из матриц , и .

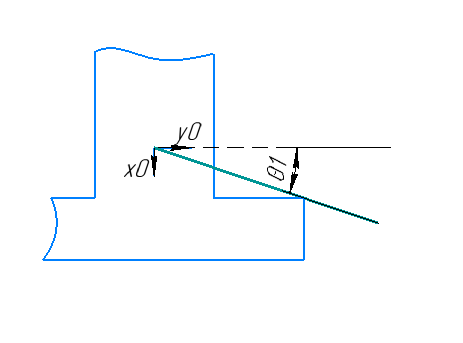


Рисунок 6

На рисунке 6 схематично показан правый поплавок катамарана и столкновение манипулятора с поплавком. Для предотвращения столкновений такого рода необходимо наложить ограничения на угол . Конструкция манипулятор не позволяет ему дотянуться до левого поплавка.

# Расчет углов шаговых двигателей и сервоприводов

Обозначим углы приводов как . Вращение мотор-редукторов по часовой стрелке примем за положительное. Шаговые двигатели имеют дискретность в 1000 шагов на оборот, а сервопривод 360 шагов на оборот. Изначально все обобщенные координаты рассчитывались в радианах. Такая система координат при необходимости позволит сократить число вычислений с плавающей точкой на микроконтроллерах, не обладающих аппаратными возможностями для быстрых операций с плавающей точкой. Примем что нулевое положение всех шаговых двигателей и сервоприводов совпадает с нулевым положением манипулятора. Все шаговые двигатели оснащены редукторами с соотношением 1:5. Будем вести расчет координат шаговых двигателей сразу в шагах, а координат сервопривода в градусах.

Введем функцию перевода радиан в координаты шаговых двигателей.

|  |  |
| --- | --- |
|  | (51) |

Так же введем функцию перевода обобщенных координат в координаты сервоприводов

|  |  |
| --- | --- |
|  | (52) |

На рисунке 7 приведен кинематический узел, обеспечивающий поворот манипулятора вокруг оси z.

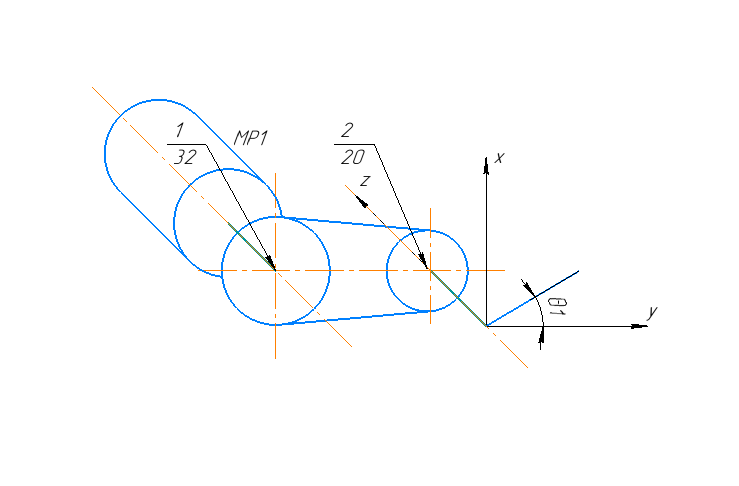


Рисунок 7

Как видно из рисунка 7

|  |  |
| --- | --- |
|  | (53) |

Далее на рисунке 8 приведена пространственная кинематическая схема, которая позволяет найти углы , и .

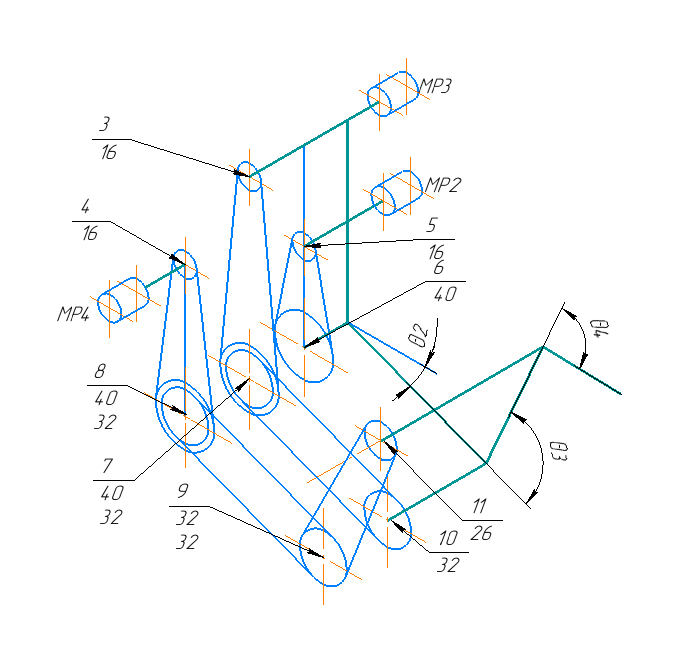


Рисунок 8

|  |  |
| --- | --- |
|  | (54) |

Далее для расчёта углов и следует учесть, что данные координаты зависят не только от обобщенных координат сочленения, но и от обобщенных координат родительских сочленений.

|  |  |
| --- | --- |
|  | (55) |
|  | (56) |
|  | (57) |

## Сравнение решений

Сравнение полученных методов решения ОЗК производилось по времени работы решения. Для сравнения использовался микроконтроллер STM32F103T6 с тактовой частотой 8 МГц. Для более корректного сравнения алгоритмов оптимизацию компилятора примем за O0. Программный код для метода ДХ приведен в приложении 3, программный код для метода геометрического представления структурно эквивалентен приложению 2. На рисунке 9 представлены диаграммы сравнения двух способов решения ОЗК – с помощью геометрического представления звеньев и с помощью формализма ДХ. На диаграммах представлено время расчета одного решения ОЗК в мс. Так же на рисунке 9 приведена диаграмма аналогичного сравнения для микроконтроллера ATMega 2560 на восьмибитной архитектуре. Как видно из графиков, решение с использованием геометрического представления дает результат примерно в 3 раза быстрее, чем решение с использованием формализма ДХ, не зависимо от архитектуры микроконтроллера. Для дальнейшей разработки бортовой программы манипулятора примем решение через геометрическое представление манипулятора как основное.

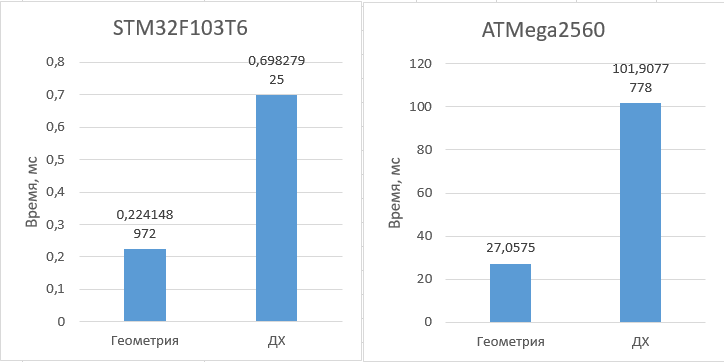


Рисунок 9

# Отслеживание положения. Обратная связь по положению

## Отслеживание положений энкодеров

Для осуществления обратной связи по положению между микроконтроллером и шаговым двигателей применяются различные методы. Самый простой и надежный метод, это отслеживание сигналов энкодеров, подключенных к драйверу шаговых двигателей. Для реализации данного метода в микроконтроллерах STM32 предусмотрена опция таймеров для аппаратной обработки сигналов инкрементальных энкодеров. Для реализации данной опции необходимо для каждого двигателя выделить таймер, способный работать в энкодерном режиме, и дополнительно два вывода микроконтроллера, связанные с двумя каналами таймера. Обрабатывать сигнал энкодеров способны таймеры 1-5 и 8. На рисунке 3 представлена схема работы инкрементального энкодера, основанного на оптопарах.

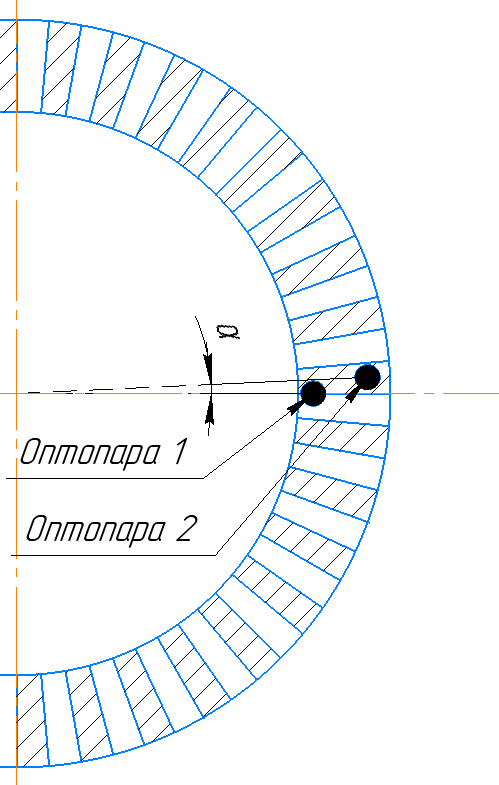


Рисунок 3 – Схема инкрементального энкодера

ШД nema 23 компонуются энкодерами, изготовленными из прозрачных дисков. По диаметру данных дисков через одинаковое расстояние нанесены штрихи, которые не пропускают свет, на рисунке обозначены штриховкой. Количество штрихов определяет дискретизацию измерения положения энкодера такого типа. В работе рассматриваются энкодеры, дискретизация которых составляет 1000 шагов на оборот, или 0,36 градуса. Смещение одной оптопары на угол α, равный половине угла дискретизации позволяет устанавливать направление выражения вала двигателя. Принцип детектирования положения ротора показан на рисунке 4.

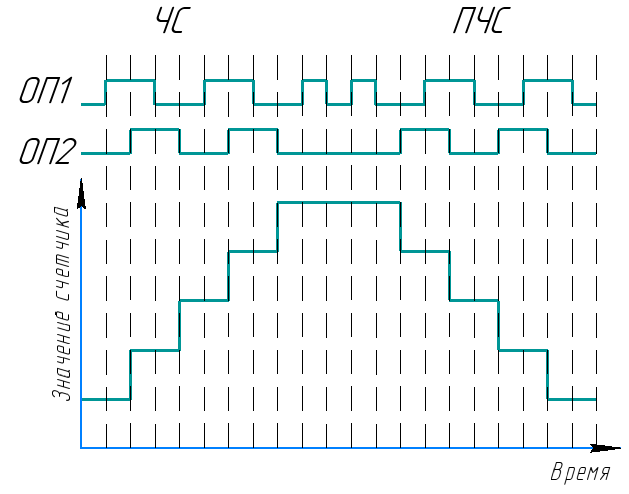


Рисунок 4 – Принцип подсчета положения энкодера

На рисунке 4 ЧС и ПЧС обозначает вращение по часовой стрелке и против часовой стрелки соответственно. ОП1 и ОП2 обозначают соответственные оптопары. Благодаря запаздыванию либо опережению фронта сигнала ОП2 относительно сигнала ОП1 определяется направление вращения. Так как для изменения значения положения таймеру микроконтроллера необходимо детектировать 2 последовательных сигнала на двух выводах, то увеличение или уменьшение счетчика таймера происходит не на один, а на два. То есть один оборот шагового двигателя все равно соответствует 1000 шагам энкодера, дискретизация так же составит 0,36 градуса, но при полном обороте в регистре счетчика таймера будет находится значение 2000.

Данный метод имеет значительный минус в количестве дополнительно используемых выводов и таймеров микроконтроллера. При использовании данного метода для работы одного двигателя необходимо использовать 5 выводов микроконтроллера, 3 для управления: STEP, DIR, EN, а также 2 для обеспечения обратной связи. При этом необходимо задействовать 2 таймера, один для генерации ШИМ сигнала, и один для отслеживания положения инкрементального энкодера.

## Метод захвата сигнала таймера

Второй метод отслеживания положения представляет из себя уже не прямую, а косвенную обратную связь двигателя с микроконтроллером. Так как драйвер шагового двигателя обеспечивает своевременное коммутирование обмоток, и засчёт обратной связи драйвера и двигателя с энкодером обеспечивается отсутствие пропуска шагов и обработка управляющих команд в реальном времени, можно разделить линию обратной связи. Таким образом драйвер обрабатывает информацию, поступающую с энкодеров, а микроконтроллер обрабатывает ШИМ сигнал, генерируемый им же. Таймеры в микроконтроллерах STM32 могут работать не только в режиме генерации ШИМ сигнала, но и в режиме захвата сигнала. Таким образом можно отслеживать какое количество импульсов было послано на драйвер двигателя, соответственно на какое количество шагов повернулся вал двигателя. Для данного метода так же необходимо при каждом срабатывании прерывания на следящем таймере учитывать какой уровень задан на управляющем пине DIR, и в зависимости от него инкрементировать, либо декрементировать текущее положение двигателя. Для реализации представленного метода так же необходимо задействовать дополнительный таймер и дополнительный вход микроконтроллера. Стоит отметить, что некоторые таймеры 1-5 и 8 могут работать в подчинённом режиме, что позволяет назначить одному таймеру роль ведущего, а второму ведомого. Данная опция позволяет отслеживать сигналы внутри микроконтроллера не соединяя выводы.

Описанные выше два метода подразумевают использование дополнительного таймера для каждого шагового двигателя. Ниже в таблице 1 приведены возможности таймеров микроконтроллера STM32F767ZI [6].

Таблица 4 – Возможности таймеров микроконтроллера STM32F767ZI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер таймера | Энкодерный режим работы | Возможность работать в режиме ведущего/ведомого | Возможность захвата внешнего ШИМ | Количество внешних ШИМ каналов |
| 1 | + | + | + | 6 |
| 2 | + | + | + | 4 |
| 3 | + | + | + | 4 |
| 4 | + | + | + | 4 |
| 5 | + | + | + | 4 |
| 6 | - | - | - | 0 |
| 7 | - | - | - | 0 |
| 8 | + | + | + | 2 |
| 9 | - | - | + | 1 |
| 10 | - | - | - | 1 |
| 11 | - | - | - | 1 |
| 12 | - | - | - | 1 |
| 13 | - | - | - | 1 |
| 14 | - | - | - | 1 |

Как видно из таблицы 1, при использовании энкодерного метода для манипулятора, имеющего 6 степеней свободы, соответственно 6 шаговых двигателей необходимо будет задействовать все таймеры, имеющие внешние каналы, 1-5 и 8 для обработки энкодеров и 9-14 для генерации ШИМ сигнала. Что делает невозможным, так как необходимо также управлять захватным устройством.

Использование ведущих и ведомых таймеров невозможно, потому что только 6 таймеров обладают данной опцией, что дает возможность управления только тремя двигателями. Использование режима захвата приводит к тому же, что и энкодерный режим – необходимо задействовать все таймеры, имеющие внешние каналы.

## Интегральный метод вычисление положения

Вышеописанное приводит к необходимости использовать меньшее количество таймеров для отслеживания положения вала шагового двигателя. Предлагается применить один таймер, который по периодическому прерыванию будет вычислять текущее положение ротора двигателя путем дискретного интегрирования текущей скорости [7]. Для реализации данного метода можно использовать один таймер, который даже может не иметь внешних выводов, что позволит значительно сократить использованную периферию. Схема алгоритма работы интегрального метода представлена на рисунке 5, программная реализация алгоритма на микроконтроллере STM32F767ZI представлена в приложении 1.

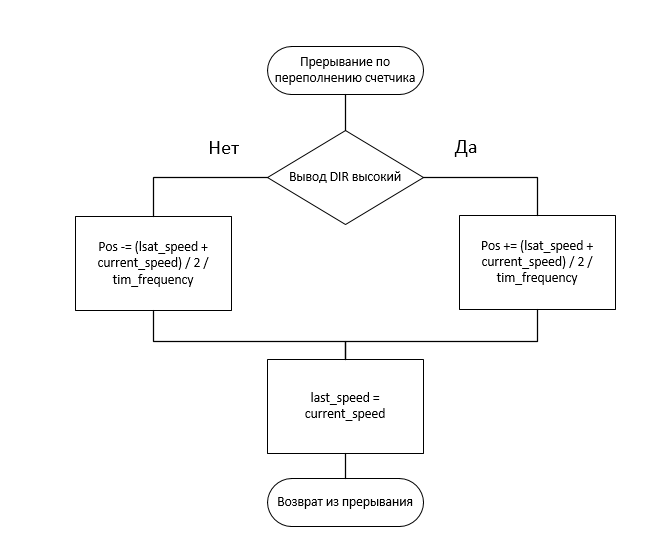


Рисунок 5 – Алгоритм расчета положения

На рисунке 5 Pos – позиция ротора, last\_speed – скорость на прошлом срабатывании прерывания, current\_speed – текущая скорость, tim\_frequency – частота таймера, использующегося для вычисления положения. Вычисление положения ротора осуществляется по методу трапеций.

Точность определения положения ротора шагового двигателя интегрального метода зависит от частоты вызова интегрального прерывания, а также от характера изменения скорости. Для компенсации ухода расчетных значений от реальных в конструкции рассматриваемого манипулятора присутствую концевые датчики.

Управление серводвигателями происходит путем задания скважности ШИМ сигнала. Период ШИМ для всех серводвигателей должен быть одинаковым, а скважность задает угол поворота вала двигателя. Таким образом, для управления серводвигателями используется один аппаратный таймер с двумя каналами, позволяющий задавать разную скважность на двух выводах, при одинаковом периоде сигнала.

Система управления расположена на микроконтроллере STM32F767ZI. Управляющие сигналы для шаговых двигателей задаются ШИМ сигналом и направлением вращения.

# Сравнение методов вычисления положения

В рамках проделанной работы был проведен эксперимент для сравнения трех методов определения положения ротора шагового двигателя. Во время эксперимента движение ротора двигателя имело равномерный и ускоренный характер, что позволяет оценить погрешность вычисления положения по интегральному методу. В рамках эксперимента драйвер двигателя настроен на 2000 микрошагов на оборот ротора, что соответствует количеству шагов энкодера, считываемых таймером микроконтроллера за один оборот. Из-за особенностей аппаратной обработки энкодеров на STM32 каждый импульс считается 2 раза, соответственно дискретизация (d) положения ротора, считанная с энкодера, также составляет 2 шага, или 0,36 градуса, что видно из формулы 1.

|  |  |
| --- | --- |
|  | (1) |

Для оценки необходимой частоты срабатывания прерывания для вычисления положения по интегральному методу необходимо сравнить вычисленное значение положения ротора со значением, полученным путем обработки энекдеров, при разных частотах срабатывания прерывания.

Для оценки точности определения положения ротора на разных скоростных режимах, заданная скорость во время проведения эксперимента изменялась по закону синуса, так как исходя из общего решения обратной задачи кинематики для манипуляторов с шарнирными сочленениями при прямолинейном движении ЗУ скорость сочленения меняется также по закону синуса. Полученные в ходе эксперимента зависимости максимальной и медианной ошибки положения ротора от частоты срабатывания прерываний позволяют оценить необходимую частоту таймера, для генерации прерываний интегрального метода вычисления положения ротора шагового двигателя.

На рисунках 6 и 7 представлены график зависимости максимальной и медианной ошибок положения ротора от частоты срабатывания прерывания.

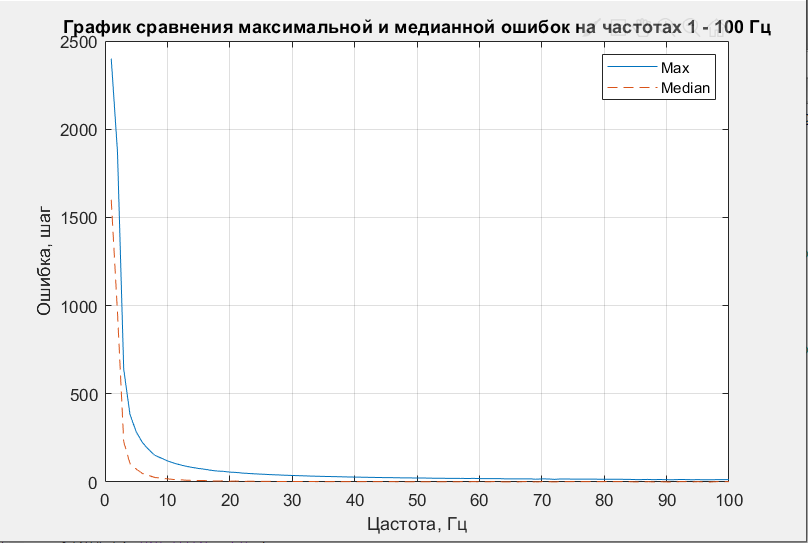


Рисунок 6 – Зависимость ошибок в диапазоне 1 – 100 Гц

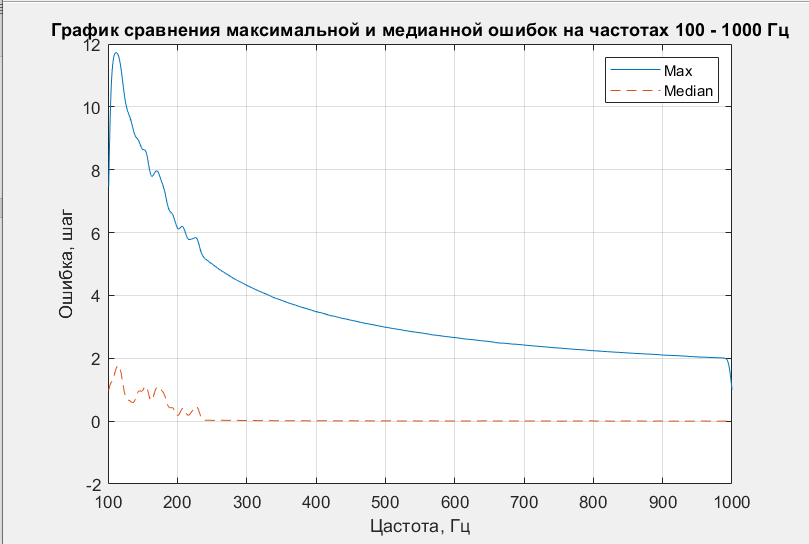


Рисунок 7 - Зависимость ошибок в диапазоне 100 – 1000 Гц

Как видно из рисунка 6 при частоте 100 Гц медианная ошибка падает ниже размера дискретизации положения, а при частоте около 1000 Гц максимальная ошибка положения ротора также падает ниже размера дискретизации. Из чего можно сделать вывод, что минимально необходимая частота срабатывания прерывания составляет 100 Гц, но при такой частоте необходимо применять фильтры сглаживания выбросов. При использовании частоты в 1000 Гц полученные значения положения ротора будут максимально соответствовать реальным и необходимость в фильтрации пропадает.

Так же для оценки необходимой частоты срабатывания прерываний необходимо учесть время обработки одного прерывания, так как оно имеет наибольший приоритет, значит другие прерывания от других событий не смогут вступить в работу, до тех пор, пока не завершится интегральное прерывание. Также стоит учесть, что микроконтроллер с системой управления манипулятором на борте помимо вычисления положения должен выполнять большое число действий, таких как отправка и прием сообщений по UART и CAN, расчет целевых координат, предупреждение столкновений, изменение скорости вращения звеньев и так далее, что также требует задействовать ресурсы ядра микроконтроллера. Следовательно, если время обработки одного прерывания интегрального таймера будет велико, то для остальных задач у микроконтроллера не останется свободного вычислительного времени ядра. Согласно проведенным измерением время, необходимое на выполнение одного прерывания для интегрального таймера для расчёта положения четырех двигателей составляет 64мкс, при системной частоте ядра равной 96 МГц. Значит при частоте прерывания 1000 Гц время, затрачиваемое на выполнения интегрального прерывания, составляет 6,4% от общего вычислительного времени микроконтроллера. При необходимости частоту срабатывания интегральных прерываний допустимо снижать со 300 Гц, тогда медианная ошибка останется близкой к нулю, а выбросы текущей ошибки положения ротора до 4х шагов необходимо фильтровать применяя фильтр высоких частот, так как исходя из результатов эксперимента значение ошибки положения ротора, при низких частотах срабатывания прерывания, имеют высокочастотных характер изменения, причем на частотах более 40 Гц частота резкого изменения ошибки положения ротора совпадает с частотой срабатывания интегрального прерывания.

# Регулятор положения ротора шагового двигателя.

Классическая задача управление подразумевает под собой сведение ошибки между заданными значениями положения скорости и ускорения к минимально возможным значениям, при этом не выходя за рамки наложенных ограничений на управление, таких как максимальные скорости и ускорения шарниров и невозможности достижения конфигураций звеньев.

## Метод генерации массива решений

Для перемещения ЗУ между точками в рабочей области манипулятора применим метод построения траекторий [8], основанный на массиве требуемых скоростей, последовательное применение которых, позволяет достичь заданного положения. Данный метод позволяет повысить точность достижения целевой позиции, гарантирует отсутствие нежелательных положений во время перемещения и позволяет реализовать плавный старт и остановку ЗУ. Суть метода заключается в разбиении траектории между двумя точками на множество точек в пространстве и дальнейшее вычисление необходимых скоростей для достижения последующей точки в минимально возможное время. Чаще всего траекторией между текущей точкой ЗУ и целевой является отрезок прямой. Возможны два варианта заполнения массива скоростей. Первый вариант применяется, когда выделенный размер массива больше либо равен требуемому объему скоростей, согласно степени дискретизации изменения скорости. Тогда каждое следующее значение из массива скоростей будет применяться через равные промежутки времени. Второй вариант применяется, когда выделенный размер массива скоростей недостаточен для хранения всего объема необходимых данный, тогда применяется метод увеличения дискретизации изменения скорости. Кроме того, при использовании программных решений, позволяющих динамически выделять память для массива управляющих скоростей пропадает необходимость увеличивать дискретизацию управления, но появляется ограничения, связанные с максимальным объемом памяти микроконтроллера. Так же следует рассмотреть вариант, когда звенья не успевают выйти на максимальную скорость, что приводит к треугольной диаграмме скоростей, в отличии от трапециевидной в предыдущем случае.

Схема работы метода без динамического выделения памяти для управляющего представлена на рисунке 8.

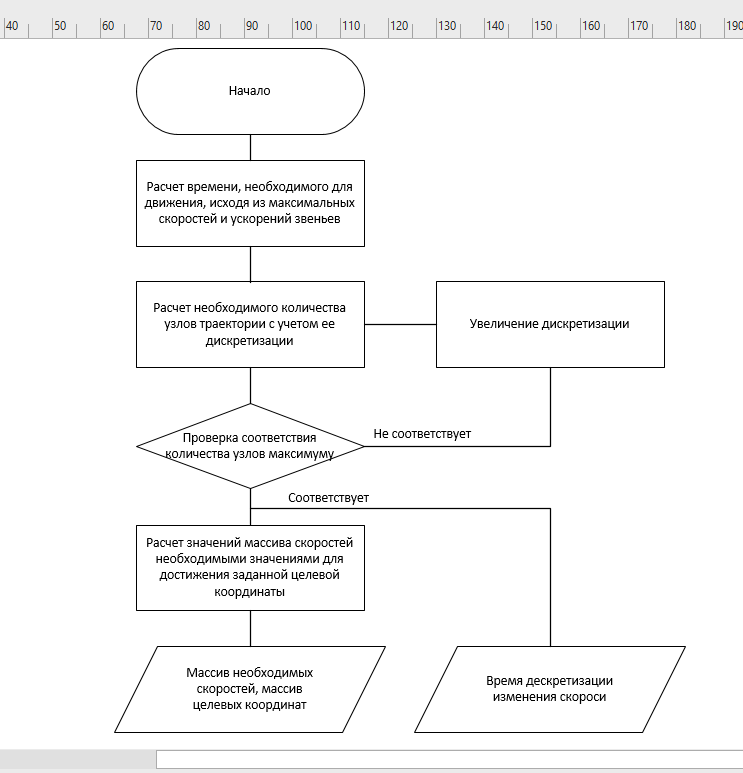


Рисунок 8 – Формирование таблицы решений

Из рисунка 8 видно, что применение данного метода создает необходимость хранения большого количества данных. Так же при необходимости перемещать ЗУ на большие расстояния появляется необходимость увеличение размеров массивов для хранения скоростей и положений ЗУ, либо уменьшение дискретизации изменение скорости, а, следовательно, увеличение частоты процедуры изменения скорости, что неизбежно приведет к более резкому движению звеньев манипулятора и ЗУ.

## Метод описания траекторий

Метод описанный выше требует большой затраты памяти микроконтроллера, что не позволительно для мобильных решений. Значит для описания траектории необходимо применить метод, использование которого не приводило бы к необходимости занимать большой объем памяти. Для этого также применяют разбивку траектории на блоки, только не по расстоянию, а по времени одного дискретного такта управления. Далее благодаря непрерывному описанию траектории между начальной и конечной позиций шарниров, используя разные способы математического описания.

Существуют несколько методов описания траектории в задачах, связанных с движением манипулятора. Основные два метода — это описание траектории с использованием B-сплайнов и с использованием полиномов нечетных степеней [9], так как нечетная степень позволяет добиться не только заданных значений кривой, но и заданных значений производных на концах отрезков. [10]

Приведем описание объекта управления и ограничений, накладываемых на объект использую полиномиальный метод. В пространстве обобщенных координат применимы следующие ограничения, накладываемые на начальные и конечные значения обобщенные координаты и их производные, представленные в формуле 2.

|  |  |
| --- | --- |
|  | (2) |

где – значение обобщенной координаты и ее скорости, и – начально и конечное значения угловой координаты, – время, необходимое для достижения заданного управления.

На практике, начальную и конечную скорости требуется приравнивать к 0. Только четыре ограничения удовлетворяются полиномом по крайней мере третьей степени. Эти ограничения однозначно определяют конкретным кубическим полиномом, представленным в формуле 3 [11, 12].

|  |  |
| --- | --- |
|  | (3) |

Используя полином, соединяющий начальное и конечное положение обобщенных координат, приведенный в формуле 3, можно сделать так что обобщенные скорости в начальный и конечный момент времени будут равны нулю. Построение траектории с помощью полинома обеспечивает плавный разгон и торможение звеньев. При увеличении степени описывающего полинома неизбежно увеличивается процессорное время, необходимое для вычисление обобщенных координат на каждом такте управления. Вычисление управляющих скоростей обобщенных координат в узлах траектории происходит путем решения обратной задачи кинематики в каждой точки траектории. В связи с этим данный метод применим только на микроконтроллерах с достаточной вычислительной мощностью, способных обрабатывать ОЗК в реальном времени. Применяемый в данной работе микроконтроллер STM32F767ZI имеет аппаратную поддержку плавающей точки, [13] что позволяет ускорить обработку тригонометрических функций, используемых в описании ОЗК [14]. При невозможности обработки траектории в реальном времени применяют метод управляющих массивов, описанный выше.

Существует несколько способов задания желаемой скорости в точках траектории:

-пользовательский,

-эвристический.

-непрерывности ускорений в точках перехода.

При использовании пользовательского способа оператор сам задает необходимые значения управляющих скоростей в каждой точке массива. Эвристический метод подразумевает применение походящей эвристики в каждой точки управляющего массива. Использование автоматического метода подразумевает введение ограничения на непрерывность ускорения в точках управляющего массива.

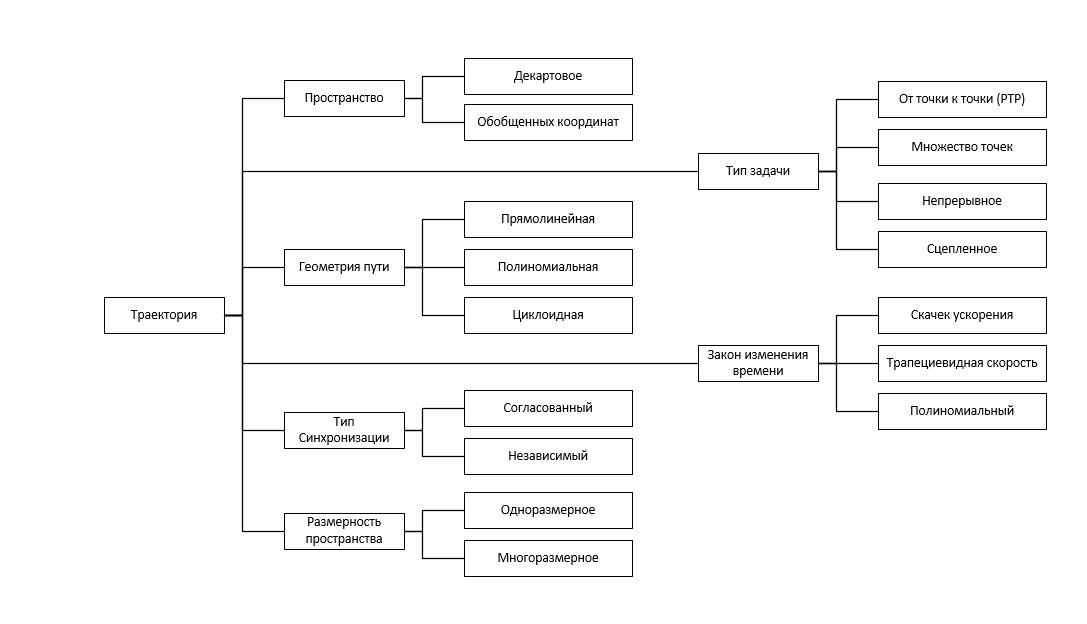
Планирование траектории

Для перемещения ЗУ из начальной точки в конечную за время *t* робототехнической системе необходимо согласованно изменять значения обобщенных координат звеньев. Для описания движения ЗУ в пространстве используются не только начальная и конечная точки, но и промежуточные точки. Такие точки выбираются в соответствии с доступной для конкретного МК дискретизацией. Так же для более подробного описания прибегают к построению зависимостей скоростей и ускорений звеньев. Для получения изменяющегося во времени положения ЗУ необходимо применить изменяющийся во времени вектор угол , вычисляемый по формуле 123.

|  |  |
| --- | --- |
| *,* | (1) |

где – зависящее от времени решение ОЗК с начальным и конечным положением ЗУ.

В робототехнике применяется несколько методов построения вектора . Все методы построения траекторий можно разделить по пространственному признаку: методы, работающие в пространстве обобщенных координат и методы, работающие в декартовом пространстве. На рисунке 123 приведена подробная схема разделения методов построения траекторий согласно [Siciliano 08b].



Сосредоточим внимание на выборе рабочего пространства для построения траекторий. При работе в декартовом пространстве на каждом шаге управления необходимо решать ОЗК для каждой промежуточной точки. При этом в большинстве случаев используется прямолинейная геометрия пути. Это позволяет построить наиболее короткую траекторию между двумя точками. При отсутствии препятствий такая траектория представляет собой отрезок прямой. Когда-же в рабочей зоне манипулятора присутствуют препятствия, то в декартовом пространстве их легко обрабатывать. При использовании декартового пространства интерполяция траектории предсказуема. Работа же в пространстве обобщенных координат позволяет значительно сократить количество расчёта ОЗК, так как при работе с данным пространством используют методы сглаживания траектории. Такие методы позволяют производить расчет ОЗК только в начальной и конечной точках пути. Так как начальная точка следующего пути- это конечная точка предыдущего, то для построения траектории необходимо сделать всего один расчет ОЗК, что значительно снижает затраты вычислительных ресурсов микроконтроллера. В таблице 5 приведено сравнение плюсов и минусов каждого пространства построения траекторий.

Таблица 5 - Сравнение пространств построения траекторий

|  |  |  |
| --- | --- | --- |
|  | Декартовое пространство | Пространство обобщенных координат |
| Плюсы | Предсказуемое движение  Лучшая обработка препятствий и столкновений | Быстрое вычисление  Движение приводов плавнее и проще для проверки |
| Минусы | Медленное вычисление  Менее плавное движение привода | Промежуточные точки не гарантируют соблюдение ограничений или столкновений |

Для сглаживания траекторий применяются методы описания полиномом нечетных степеней, трапецией и Б-сплайном. Нечетные степени в полиномах используются из-за того, что кривая нечётной степени и ее производная принимает необходимые значения,

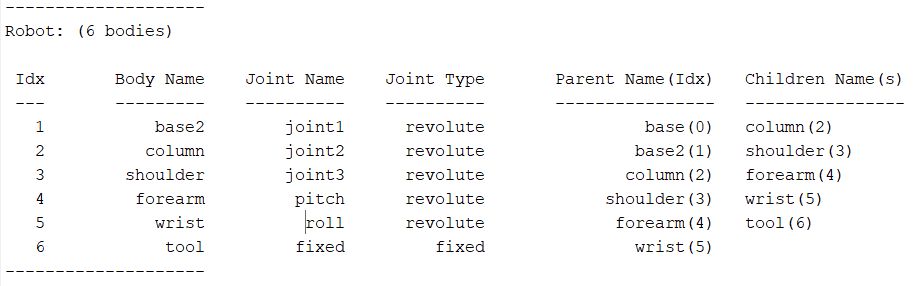
Пакет моделирования MatLAB Robotic System Toolbox позволяет сравнить

Список литературы

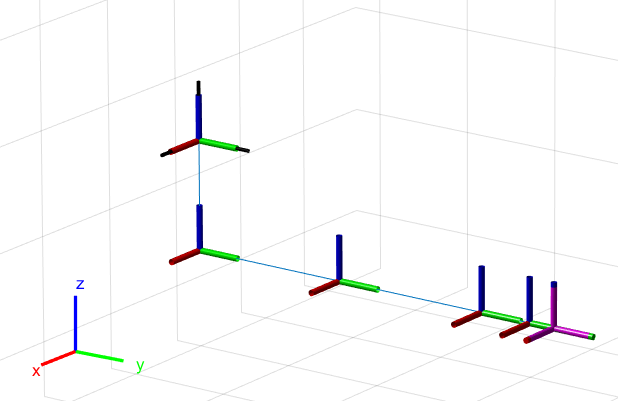
1. Guo D. et al. Efficient algorithms for the kinematics and path planning of manipulator // 2009 International Conference on Artificial Intelligence and Computational Intelligence, AICI 2009. 2009. Vol. 2. P. 282–287.

Приложение 1

Параметры модели манипулятора в Matlab Robotic System Toolbox



Так же на рисунке ниже приведена получившаяся модель.



Приложение 2

Функция в MATLab для проверки правильности расчета ОЗК.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | **function** [theta] = **IK**(x, y, z, r, p)  theta = [pi/**2** **0** **0** -pi/**2** **0** pi/**2**];  s = [**0** **0** **0** **0** **0**];  s(**5**) = r;  L = sqrt(x^**2** + y^**2**);  s(**1**) = atan2(x, y);  L23 = L - l4 \* cos(p);  H23 = z - l4 \* sin(p);  Q23 = sqrt(H23^**2** + L23^**2**);  alpha = acos((L23^**2** + Q23^**2** - H23^**2**)/(**2** \* L23 \* Q23));  betta = acos((Q23^**2** + l2^**2** - l3^**2**)/(**2** \* Q23 \* l2));  **if** H23 <= **0**  s(**2**) = -alpha - betta;  **else**  s(**2**) = -betta + alpha;  **end**  H34 = l2 \* sin(s(**2**)) - z;  L34 = L - l2 \* cos(s(**2**));  Q34 = sqrt(L34^**2** + H34^**2**);  psi = acos((l3^**2** + l4^**2** - Q34^**2**)/(**2** \* l3 \* l4));  s(**4**) = pi - psi;  gamma = acos((l2^**2** + l3^**2** - Q23^**2**)/(**2** \* l2 \* l3));  s(**3**) = pi - gamma;  theta(**1**) = s(**1**) + pi/**2**;  theta(**2**) = -s(**2**);  theta(**3**) = -s(**3**);  theta(**4**) = s(**4**) - pi / **2**;  theta(**5**) = s(**5**);  theta(**6**) = pi/**2**;  **end** |

Приложение 3

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | **float**\* **IKDH** (**float** x, **float** y, **float** z ,**float** p ,**float** r)  {  **float** s [**5**] = {**0**};  **float** nx = cos(s[**0**] \* cos(r) \* cos(p) - cos(s[**0**] \* sin(p)));  **float** ny = sin(s[**0**] \* cos(r) \* cos(p) + cos(s[**0**] \* sin(p)));  **float** nz = -sin(r) \* cos(p);  **float** ox = cos(s[**0**] \* cos(r) \* sin(p) - sin(s[**0**] \* cos(p)));  **float** oy = -sin(s[**0**] \* cos(r) \* sin(p) + cos(s[**0**] \* sin(p)));  **float** oz = sin(r) \* sin(p);  **float** ax = sin(r) \* cos(p);  **float** ay = sin(r) \* sin(p);  **float** az = cos(r);  **float** s234 = acos(-az);  s[**0**] = atan(-y / x);  **if** (s234 != **0**)  s[**4**] = atan(-oz / nz);  **else**  s[**4**] = s[**0**] - atan(ox / nx);  **int** al = -l1 - l4\*cos(s234) - **0** \* sin(s234) - z;  **int** b = cos(s[**0**]) \* x + sin(s[**0**]) \* y + l4 \* sin((s234)) - **0** \* cos(s234);  s[**2**] = acos((pow(al, **2**) + pow(b, **2**) - pow(l2, **2**) - pow(l3, **2**))/  (**2** \* l2 \* l3));  s[**1**] = atan((al \* (l3 \* sin(s[**2**])+ l2) - b \* l3 \* sin(s[**2**]))/  b \* (l3 \* cos(s[**3**]) + l2) + al \* l3 \* sin(s[**2**]));  s[**3**] = acos(-az) - s[**2**] - s[**1**];  **float** \*s2 = s;  **return** s2;  } |