анкт-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

2012

Министерство образования и науки Российской Федерации

Санкт-Петербургский политехнический университет   
петра великого

Институт машиностроения, материалов и транспорта

Высшая школа автоматизации и робототехники

ЗАДАНИЕ

на выполнение научно-исследовательской работы

Сидоренко Данилу Дмитриевичу

студенту гр. 3331506/60401

1 Тема работы

«Разработка системы управления для пятизвенного манипулятора с

ременными передачами на шарниры»

2 Срок сдачи студентом законченной работы

28 июня 2020 г.

3 Исходные данные к работе

Входное изображение и файл облака точек

4 Содержание расчетно-пояснительной записки

Рассмотрение методов реализации обратной связи для шаговых двигателей с применением драйвера шаговых двигателей с управляющими сигналами типа STEP/DIR. Произвести сравнение нескольких методов реализации обратной связи по положению. Описать применяемые методы построения траекторий звеньев манипуляторов.

5 Перечень графического материала (с точным указанием обязательных чертежей)

Нет

6 Консультанты по работе

Васильев Иван Анатольевич, к.т.н, доцент ВШАиР.

7 Дата выдачи задания 01 февраля 2020 г.

Руководитель Габриель Антон Сергеевич, старший преподаватель

(ФИО, должность, подпись руководителя)

Задание принял к исполнению 01 февраля 2020 г.

(дата)

(подпись студента)

**Реферат**

На 28 с., 8 рисунков, 1 таблицы, 1 приложения

РОБОТОТЕХНИКА, МАНИПУЛЯТОР, БОРТОВАЯ СИСТЕМА УПРАВЛЕНИЯ, СИСТЕМА УПРАВЛЕНИЯ МАНИПУЛЯТОРОМ, ЭНКРЕМЕНТАЛЬНЫЙ ЭНКОДЕР, ОБРАТНАЯ СВЯЗЬ, АППАРТАНЫЙ ТАЙМЕР STM32.

В данной работе представлено сравнение методов детектирования положения ротора шагового двигателя как непосредственно считывая значения с энкодера, так и эмуляция обратной связи внутренними инструментами микроконтроллера STM32F767ZI. Описанный метод эмуляции позволяет сократить необходимое количество периферии. Сравнение методов показывает, что каждый из них имеет как плюсы, так и минусы и может применяться в зависимости от поставленной задачи.

Так же в работе приводится сравнение математического описания траектории движения звеньев манипулятора и табличного метода задания управляющих скоростей Описанные методы построения траектории движения звеньев манипулятора позволяют генерировать необходимое управление на каждом такте дискретного управления.

**Содержание**

[Введение 5](#_Toc41514687)

[1 Постановка задачи 6](#_Toc41514688)

[1.1 Кинематическая схема манипулятора 7](#_Toc41514689)

[1.2 Электрическая функциональная схема манипулятора 9](#_Toc41514690)

[2 Отслеживание положения. Обратная связь по положению 11](#_Toc41514691)

[2.1 Отслеживание положений энкодеров 11](#_Toc41514692)

[2.2 Метод захвата сигнала таймера 13](#_Toc41514693)

[2.3 Интегральный метод вычисление положения 15](#_Toc41514694)

[3 Сравнение методов вычисления положения 18](#_Toc41514695)

[4 Регулятор положения ротора шагового двигателя. 21](#_Toc41514696)

[4.1 Метод генерации массива решений 21](#_Toc41514697)

[4.2 Метод описания траекторий 23](#_Toc41514698)

[Вывод 25](#_Toc41514699)

[Список литературы 26](#_Toc41514700)

[Приложение 1 28](#_Toc41514701)

**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

ШД – шаговый двигатель

ПИД – пропорционально-интегрально-дифференцирующий регулятор

ОУ – объект управления

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. Также в конструкцию манипулятора была заложена возможность обслуживания и замены аккумуляторов на сигнальных буях.

Манипулятор предусматривает управление как в ручном режиме по скоростям, так и в программном по координатам 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 рассмотрены методы обстрой связи по положению для шаговых двигателей, а также методы построения траекторий звеньев.

# Постановка задачи

Система управления манипулятором, базирующаяся на микроконтроллере, подразумевает наличие программного блока низкого уровня управления приводами. Данный блок должен включать в себя подпрограммы регулятора скорости, регулятора положения, систему обратной связи приводов по скорости и положению. Для реализации систем обратной связи необходимо оптимизировать обработку сигналов обратной связи так, чтобы ресурсы микроконтроллера расходовались наиболее оптимально по времени. Так же не стоит забывать и про точность обратной связи. Готовое решение должно приближаться к максимально возможной степени дискретизации. Так же для выбора метода организации обратной связи требуется провести сравнение методов по затрачиваемым периферийным ресурсам микроконтроллера.

Для реализации регуляторов скорости и положения необходимо математически описать траекторию, по которой будет двигаться каждое сочленение манипулятора.

Управление объектом, имеющим несколько степеней подвижности сама по себе сложная задача, так же дополнительный трудности в ее решения вносит влияние степеней подвижности друг на друга. [1, 2]

Для лучшего понимания задачи далее приведены кинематическая и электрическая функциональная схемы манипулятора.

## Кинематическая схема манипулятора

На рисунке 1 приведена кинематическая схема манипулятора.



Рисунок 1 – Кинематическая схема манипулятора

На валу мотор-редуктора 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 от 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.

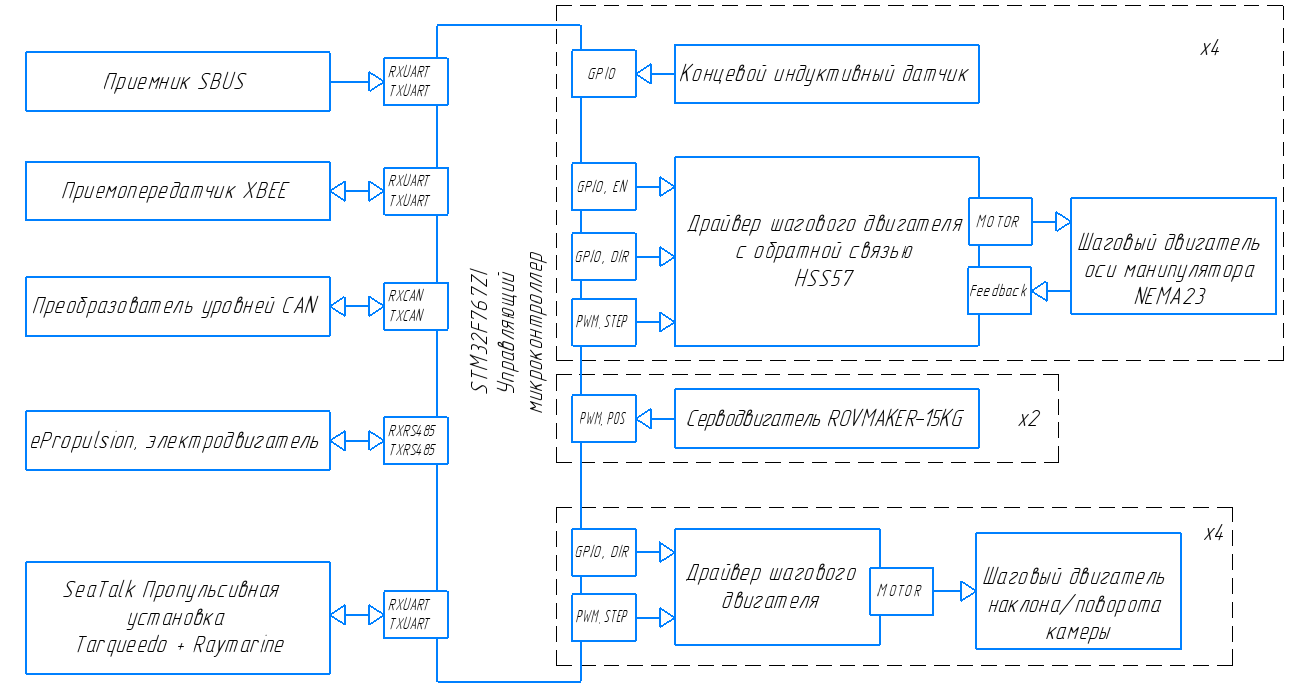


Рисунок - Электронная функциональная схема манипулятора

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].

# 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.

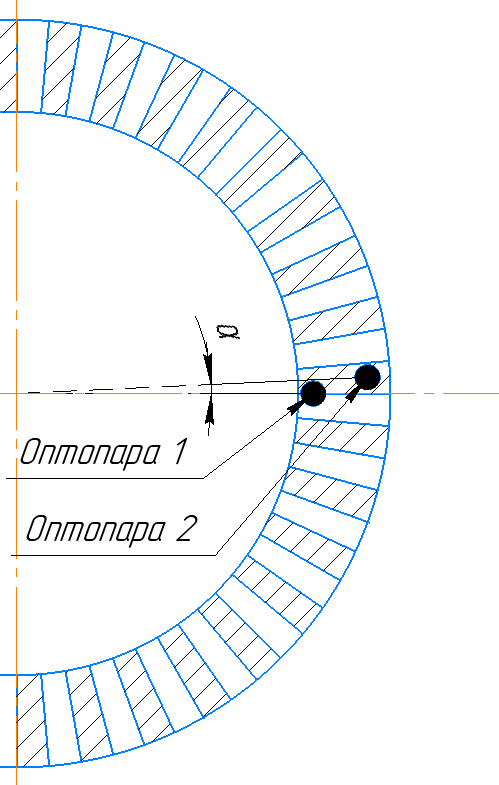


Рисунок – Схема инкрементального энкодера

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.

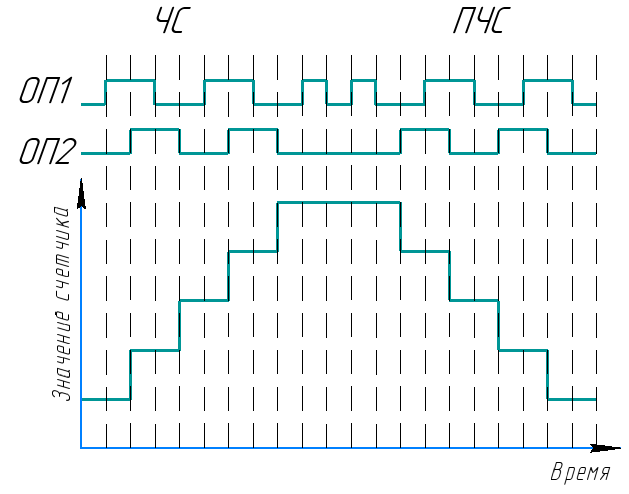


Рисунок – Принцип подсчета положения энкодера

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 – 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.

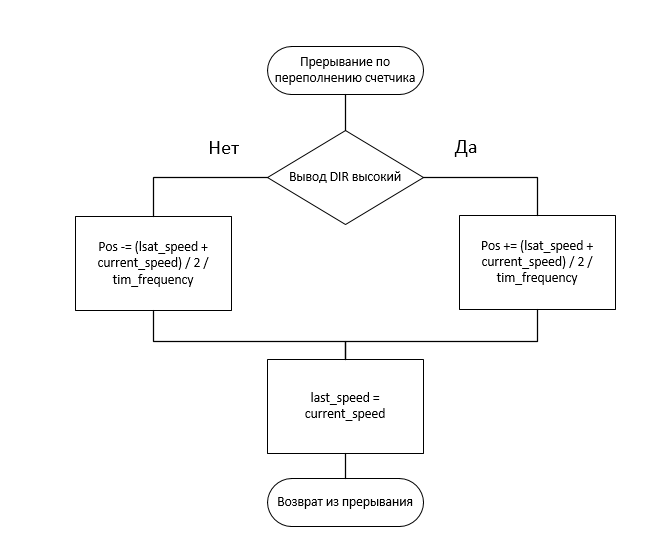


Рисунок – Алгоритм расчета положения

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.

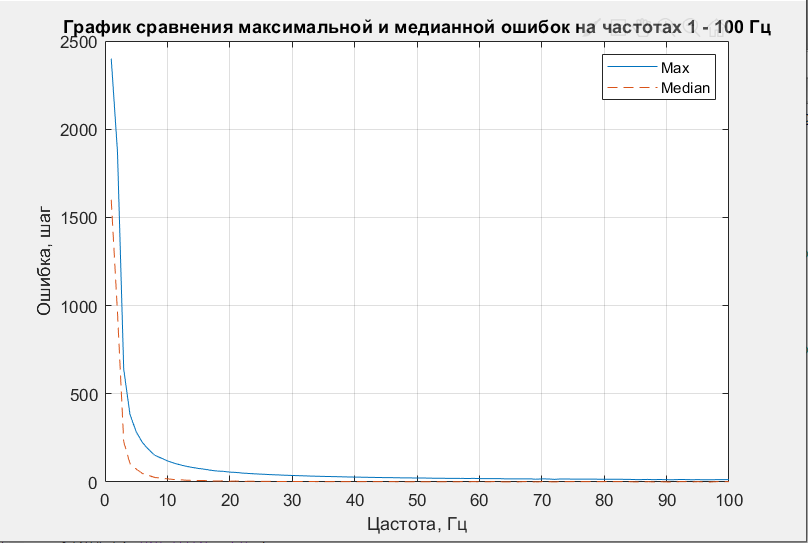


Рисунок – Зависимость ошибок в диапазоне 1 – 100 Гц

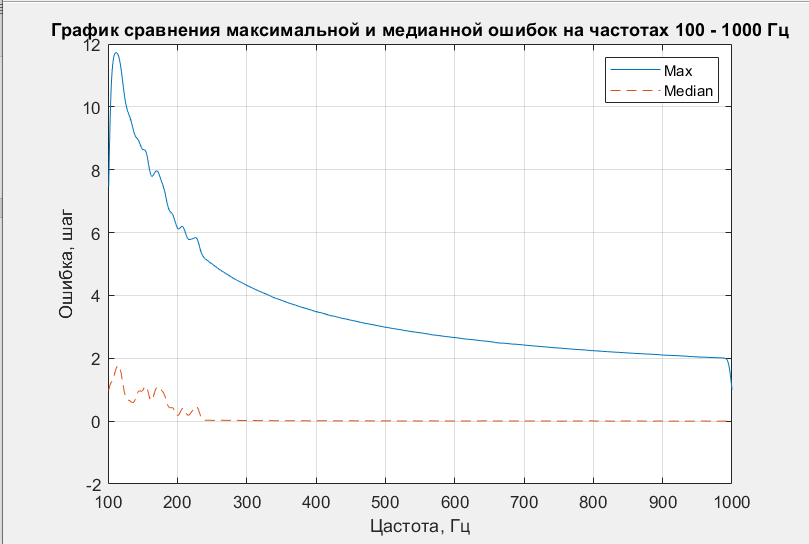


Рисунок - Зависимость ошибок в диапазоне 100 – 1000 Гц

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.

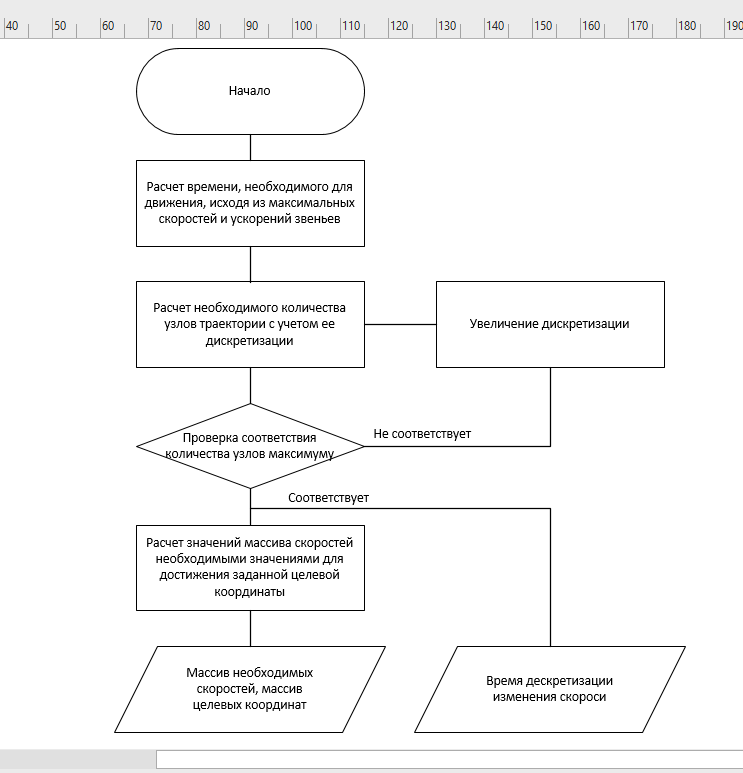


Рисунок – Формирование таблицы решений

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 ускорение, 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 тригономметрических 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.

Вывод

Предложенный в работе метод имитации обратной связи по положению шаговых двигателе, оснащенных инкрементальным энкодером, с микроконтроллером обеспечивает точность расчета положения соизмеримую с прямыми измерениями сигналов энкодера при частотах срабатывания обработчика прерывания более 300 Гц. Предложенное решение позволяет сократить использование периферии микроконтроллера, что в некоторых случаях может быть критично, так как энкодеры в робототехника применяются не только в связи с валом двигателя, для детектирования его положения, но и отдельно от двигателей для создания сигналов управления. Такие энкодеры необходимо обрабатывать непосредственно с помощью аппаратных таймеров. Предложенный метод позволяет расширить возможность количество подключаемых периферийных устройств.

Описанные методы построения траекторий звеньев манипулятора позволяют реализовывать регулирование скоростей на каждом такте управления манипулятора.

Список литературы

1. Design, Modeling and Control of a 5-DoF Light-Weight Robot Arm for Aerial Manipulation /Carmine Dario Bellicoso, Luca Rosario Buonocore, Vincenzo Lippiello, and Bruno Siciliano. — 2015.
2. On the Design, Modeling and Control of a Novel Compact Aerial Manipulator /D. Wuthier, D. Kominiak, C. Kanellakis, и др. — 2016.
3. 2HSS57-A Hybrid Stepper Servo Drive Manual [Электронный ресурс] /. — Электрон. текстовые дан. — Режим доступа: https://www.jmc-motor.com/file/1806082314.pdf, свободный (дата обращения: 10.04.20).
4. 57J1880EC-1000 Stepper Servo Motor Manual [Электронный ресурс] /. — Электрон. текстовые дан. — Режим доступа: https://www.jmc-motor.com/product/890.html, свободный (дата обращения: 10.04.20).
5. Rov underwater servo user manual [Электронный ресурс] /. — Электрон. текстовые дан. — Режим доступа: http://rovmaker.org/products/Underwater\_Servo/attachment/MANUAL\_Underwater\_Servo.pdf, свободный (дата обращения: 10.04.20).
6. Arm® Cortex®-M7 32b MCU+FPU, 462DMIPS, up to 2MB Flash/512+16+4KB RAM, USB OTG HS/FS, 28 com IF, LCD, DSI [Электронный ресурс] /. — Электрон. текстовые дан. — Режим доступа: https://www.st.com/en/microcontrollers-microprocessors/stm32f767zi.html#resource, свободный (дата обращения: 05.04.20).
7. 3-axis UART Stepper Hub [Электронный ресурс] / Alex Muzychko. — Электрон. текстовые дан. — Режим доступа: https://github.com/omuzychko/StepperHub, свободный, (дата обращения: 10.04.20).
8. Trajectory planning for automatic machines and robots /Luigi Biagiotti & Claudio Melchiorri. — Springer Science & Business Media, 2008.
9. From motion planning to trajectory control with bounded jerk for service manipulator robots /X. Broqu`ere & D. Sidobre. — IEEE Int. Conf. Robot. And Autom., 2010.
10. Soft motion trajectory planner for service manipulator robot /Xavier Broquere, Daniel Sidobre & Ignacio Herrera-Aguilar. — Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, pages 2808– 2813, Sept. 2008.
11. Path planning for redundant manipulator without explicit inverse kinematics solution /Wang, W., & Li, Y. — IEEE International Conference on Robotics and Biomimetics (ROBIO) 2009.
12. Trajectory planning and control for robot manipulations /Ran Zhao. — 2015
13. STM32F7 Series and STM32H7 Series Cortex®-M7 processor programming manual [Электронный ресурс] /. — Электрон. текстовые дан. — Режим доступа: https://www.st.com/en/microcontrollers-microprocessors/stm32f767zi.html#resource, свободный (дата обращения: 05.04.20).
14. Efficient Algorithms for the Kinematics and Path Planning of Manipulator /Dequan Guo, Hui Ju, Yuqin Yao, Feng Ling and Tianxiang Li. — 2009.

Приложение 1

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | **void** **HAL\_TIM\_PeriodElapsedCallback**(TIM\_HandleTypeDef \*htim)  {  **if**(htim->Instance == TIM5)  {  **int** speed = stepper1.current\_speed\_sps;  **if**(st1\_flag && count != **0**)  {  **if**(st1\_flag == **1**)  integ\_position -= (last\_integ\_position + speed)  / **2** / tim5\_freqHz;  **else** **if** (st1\_flag == **2**)  integ\_position -= (last\_integ\_position + speed)  / **2** / tim5\_freqHz;  last\_integ\_position = speed;  }  }  } |