

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра информатики
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

А.С. Кузнецов

подпись

инициалы, фамилия

« 13 » 06 20 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Применение метода интерполяции замкнутых кривых в алгоритме обработки
геометрических данных слитков

тема

09.04.04 «Программная инженерия»

код и наименование направления

09.04.04.01 «Программное обеспечение вычислительной техники и
автоматизированных систем»

код и наименование магистерской программы

Научный руководитель

9.06.17 доцент, к.т.н А.С. Кузнецов
подпись, дата должность, ученая степень инициалы, фамилия

Выпускник

9.06.17 А.М.Цирюльников
подпись, дата инициалы, фамилия

Рецензент

9.06.17 профессор, д.т.н С.В. Ченцов
подпись, дата должность, ученая степень инициалы, фамилия

Красноярск 2017

РЕФЕРАТ

Выпускная работа в форме магистерской диссертации по теме «Применение метода интерполяции замкнутых кривых в алгоритме обработки геометрических данных слитков» содержит 53 страницы, 23 рисунка, 11 приведённых использованных источников.

КРИВЫЕ БЕЗЪЕ, ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС, ИНТЕРПОЛЯЦИЯ, ЗАМКНУТЫЕ КРИВЫЕ, КОНТРОЛЬ ГЕОМЕТРИИ

В работе рассматривается проблема автоматизированного контроля геометрии выпускаемой продукции.

Объектом исследования является применение методов интерполяции замкнутых кривых.

Цель диссертационного исследования состоит в применении методов интерполяции замкнутых кривых для контроля геометрических данных выпускаемой продукции и разработке программно-аппаратного комплекса .

Поставленная цель достигается путем решения следующих задач:

- Исследование методов контроля геометрии,
- Исследование методов интерполяции.
- Разработка программно-аппаратного комплекса

Основная идея работы заключается в разработке программного обеспечения контролирующего геометрические данные выпускаемой продукции.

Новизна работы заключается в применении методов интерполяции замкнутых кривых для контроля геометрии.

В результате проведённой работы был разработан прототип программно-аппаратного комплекса.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. Обзор существующих методов контроля геометрии	6
1.1 Стереофотограмметрия	6
1.1.1 Основные этапы обработки визуальной информации в системах, основанных на методах стереофотограмметрии	6
1.1.2 Способы маркировки исследуемых поверхностей	7
1.2 Трехмерное лазерное сканирование	8
1.3 Контактное измерение	9
1.4 Способ измерения геометрии слитков	10
1.4.1 Преимущества предложенного метода измерений	12
2. Программное обеспечение	13
2.1. Программное обеспечение нижнего уровня	13
2.2 Программное обеспечение верхнего уровня	15
2.2.1 Используемые технологии	15
2.2.2 Технология OLE for Process Control (OPC)	15
2.2.3 Технология Windows Communication Foundation (WCF)	16
2.2.4 Схема взаимодействия компонентов программного и аппаратного обеспечения	17
2.2.5 Модуль сервера	19
2.2.5.1 Настройки серверной части	19
2.2.5.2 Структура серверной части	19
2.2.6 Краткое описание алгоритмов обработки данных	20
2.2.7 Хранение данных	22
3. Алгоритмы интерполяции замкнутых кривых.	23
3.1 Периодические кривые Безье	23
3.2 Сплайны Катмалла-Рома (Catmull-Rom splines)	25
3.2.1 Исследование алгоритма построения Сплайнов Катмалла-Рома	29
3.3 Дискретные периодические сплайны с векторными коэффициентами	36

3.3 Исследование алгоритма построения дискретных периодических сплайнов с векторным коэффициентом.	37
3.4 Сравнение сплайнов Катмалла-Рома и Дискретных периодических сплайнов с векторным коэффициентом.	45
ЗАКЛЮЧЕНИЕ	47
СПИСОК СОКРАЩЕНИЙ	49
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	50

ВВЕДЕНИЕ

В настоящий момент на предприятии по производству алюминиевых слитков есть нерешенные задачи, связанные с контролем качества выпускаемой продукции. Наиболее проблематичной является задача по контролю геометрии и поверхности выпускаемой продукции:

- слитков цилиндрической формы
- слитков прямоугольной формы

По сей день, замеры диаметра, поперечного изгиба и продольного изгиба, дефекты поверхности снимаются вручную. Данный метод измерения параметров неудобен и с его помощью, в условиях производства огромных партий продукции, невозможно выдержать нужный темп производства и сохранить высокое качество выпускаемой продукции. В связи с вышеперечисленным, напрашивается автоматизированное решение данной задачи. В силу особенностей и ограничений существующих на производстве программно-аппаратный комплекс разрабатываемый для решения данной проблемы должен отвечать следующим требованиям:

- Бесконтактный способ измерения
- Программно-аппаратный комплекс не должен каким-либо способом изменять существующий технологический процесс(не допускается замедление скорости конвейера, его удлинение)
- не допускается наличие движущихся частей в конструкции программно-аппаратного комплекса(сервоприводов, валиков, и т.д.)
- Максимально допустимая погрешность измерений 0.1 мм

Исходя из требований была спроектирована аппаратная часть в которую входят:

- Лазерные точечные датчики
- Лазерные 2D датчики
- ОПС Контроллеры

программная часть, которая представляет собой разветвленную программную систему, включающую в себя следующие модули:

- Автоматизированное рабочее место Инженера
- Автоматизированное рабочее место Контролера
- Подсистема поставки данных
- Подсистема преобразования данных

Подробнее рассмотрим две последние подсистемы.

Подсистема поставки данных - снимает показания датчиков, и записывает “сырые” данные в базу данных.

Подсистема преобразования данных - создает геометрическую модель слитка из “Сырых данных”. Под сырыми данными подразумеваются показания датчиков, а это в свою очередь расстояния от точки крепления датчика на рамке, до поверхности слитка. Эти длины переводятся в точки, далее на основе полученных точек строится замкнутая кривая с помощью алгоритма дискретных сплайнов, построенного на основе метода интерполяции замкнутых кривых. На выходе получается набор “Срезом слитка”(набор замкнутых кривых описывающих его форму). И уже у полученной модели измеряются все параметры.

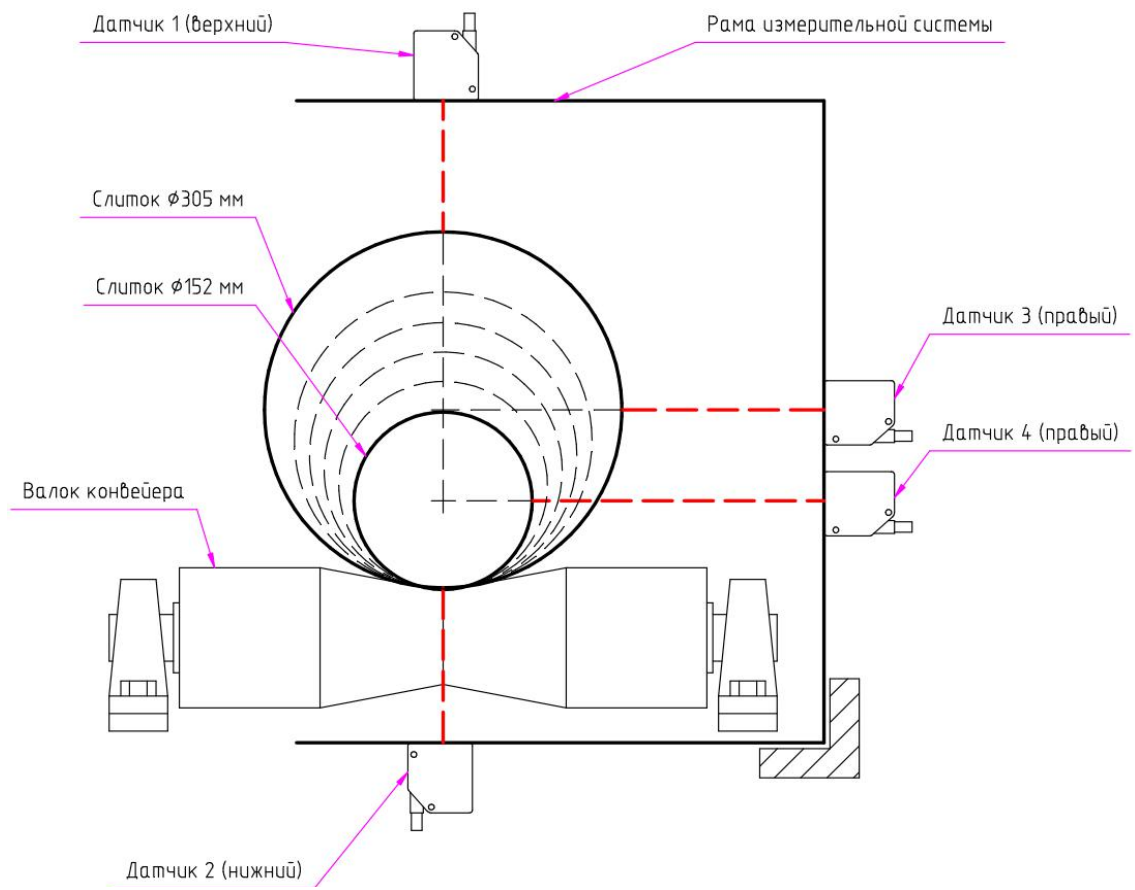


Рисунок 1 - чертеж измерительной рамки.

1. Обзор существующих методов контроля геометрии

Получение результатов измерений с известной погрешностью или с погрешностью, не превышающей допустимых пределов (норм погрешности измерений), является важнейшим условием метрологического обеспечения производства. Известен ряд методов контроля геометрии объектов, основанных на различных принципах.

1.1 Стереофотограмметрия

Система, удовлетворяющая высоким требованиям по точности измерений, а также обеспечивающая возможность одномоментного оценивания отклонений формы отражающей поверхности объекта целиком, в режиме реального времени, может быть реализована на основе принципов стереофотограмметрии.

1.1.1 Основные этапы обработки визуальной информации в системах, основанных на методах стереофотограмметрии

В системе измерения координат точек поверхности, основанной на принципах стереофотограмметрии, задача обработки изображений сводится к установлению маркеров на изображениях стереопары и нахождению их координат в системе координат каждого снимка.

Данную задачу можно разбить на подзадачи:

Распознавание маркеров. В общем случае классификация всех объектов на изображении на два класса: маркеры и помехи.

Локализация маркеров. Определение точных координат маркеров на изображениях. От точности локализации напрямую зависит точность всей системы контроля формы.

Идентификация маркеров. Установление соответствия между маркерами на изображениях стереопары.

1.1.2 Способы маркировки исследуемых поверхностей

Распространенным способом маркировки является проецирование на контролируемую поверхность различных изображений маркеров. Некоторые виды маркеров представлены на рисунке 2 (а-д). Часто с целью идентификации или по другим причинам используют кодированные маркеры (coded target), т.е. изображение маркера содержит небольшое количество информации. Существуют варианты с кодированием 10, 32, 76 бит и другие.

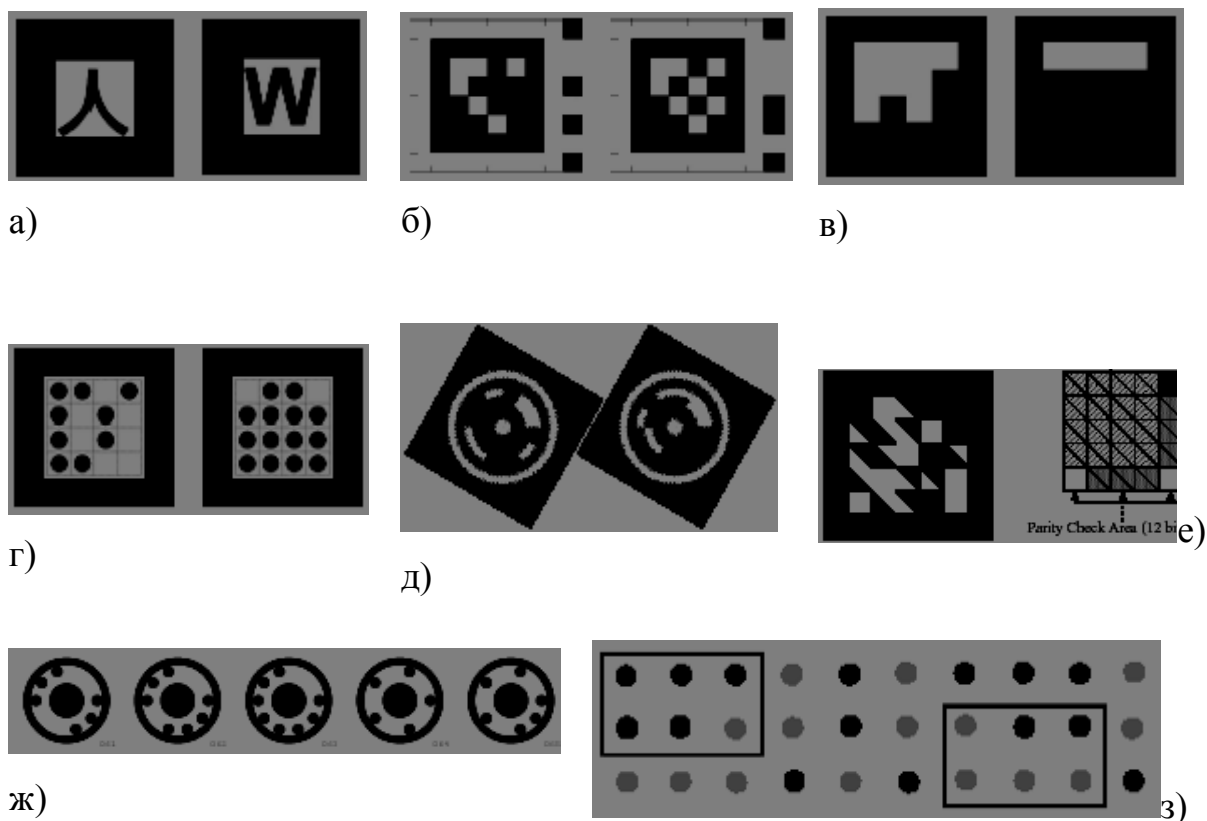


Рисунок 2 - Некоторые виды маркировки.



Рисунок 3 - Установка стереофотограмметрии V-STARS.

1.2 Трехмерное лазерное сканирование

Способ измерения основан на использовании лазерных трехмерных сканеров (рисунок 4), обеспечивающих снятие координат произвольных объектов в пространстве с высокой точностью



Рисунок 4 - Лазерный трехмерный сканер Surphaser 50HS.

Два или более трехмерных сканеров размещаются вокруг измеряемого объекта. Данные со сканеров поступают в систему управления, обеспечивающую «сшивание» данных в единую трехмерную модель объекта.

Преимущества метода: отсутствие движущихся частей; возможность проведения измерений для произвольного объекта, находящегося в любом месте литейного отделения; высокая точность.

Недостатки: исключительно высокая стоимость комплектующих изделий; низкая скорость измерений; потенциальная уязвимость основного узла – трехмерного сканера; сложные и неясные на данный момент алгоритмы обработки («сшивания») данных; потребность в пространстве для размещения системы.

1.3 Контактное измерение

Способ основан на применении т.н. лазерного трекера . Оператор при помощи специального щупа дотрагивается до определенных точек на поверхности слитка. Количество и расположение точек выбирается в соответствии с регламентом измерений. Щуп измеряет координаты каждой точки в пространстве с высокой точностью (около 100 мкм) и передает их по беспроводному каналу на трекер на расстоянии до 40 м. Координаты с трекера передаются в систему управления для анализа.



Рисунок 5 - Координатное контактное измерение при помощи лазерного трекера I-PROBE Wireless.

Преимущества: возможность проведения измерений произвольного слитка в зоне действия координатного поля (любое место литейного отделения); не требуется перемещение слитка; отсутствие движущихся частей.

Недостатки: ручное выполнение измерений; низкая скорость выполнения измерений; отсутствует возможность снятия полного профиля поверхности

слитка; не вполне ясные алгоритмы обработки результатов измерений, существует вероятность ошибок оператора; высокая стоимость комплектующих изделий.

1.4 Способ измерения геометрии слитков

Описанные выше известные способы измерения геометрии объектов обладают недостатками, не позволяющими эффективно использовать их в условиях действующего литейного производства. К ним относятся:

Невысокая скорость измерения, не гарантируется измерение всех без исключения выпускаемых слитков;

-Потребность в дополнительном пространстве для выполнения измерений,

-Присутствует ручной труд;

-Высокая стоимость комплектующих;

-Сложность алгоритмов управления и обработки данных;

-Низкая устойчивость к неблагоприятным внешним воздействиям.

В результате анализа имеющихся способов, был предложен новый способ измерения. Способ измерения заключается в продольном бесконтактном «сканировании» поверхности слитка при его перемещении по рольгангу. Между роликами рольганга устанавливается измерительная установка, представляющая собой прямоугольную раму.

На раме размещены лазерные бесконтактные лазерные триангуляционные датчики расстояния Banner LE 550 рисунок 6. Датчики подобного рода используются в большом количестве в прочих установках литейного отделения.

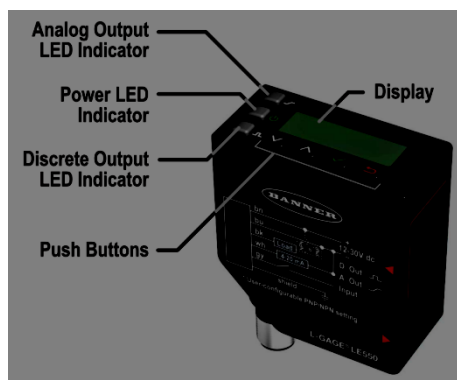


Рисунок 6 - Бесконтактный датчик Banner LE 550.

Выбор датчика данного типа обусловлен необходимой точностью и диапазоном измерения. Датчик имеет следующие параметры:

диапазон измерения: 100-400 мм;

точность измерения: 0,1 мм;

время измерения: 2 мс;

выходной сигнал: аналоговый 4-20 мА.

Для выполнения измерений в соответствии с регламентом на измерительной раме должно быть установлено 16 датчиков. Измерение нижней поверхности слитка осуществляется через пространство между роликами рольганга. Измерение верхней поверхности выполняется датчиками на верхней горизонтальной части рамы. Размеры рамы выбраны таким образом, чтобы обеспечивалось измерение слитков всех имеющихся типоразмеров.

При перемещении слитка по рольгангу датчики измеряют расстояние до поверхности слитка и передают измеренные значения в систему управления. При этом формируется чертеж поперечного сечения слитка. Благодаря передвижению слитка по рольгангу, обеспечивается снятие сечения по всей длине слитка. Таким образом, снимается потребность в сложном и ненадежном в промышленных условиях механическом устройстве, перемещающем измерительную раму вдоль слитка.

Для устранения погрешности, вызванной не идеальным размещением слитка на рольганге (отсутствием опорной точки), применяются парные датчики, направленные на слиток с противоположных сторон. Чтобы

обеспечить измерение слитков различного типоразмера, используется набор датчиков, установленных на раме с различным смещением относительно ее центра.

Для компенсации возможного смещения слитка вверх-вниз и вправо-влево при движении по рольгангу используются специальные алгоритмы обработки данных.

1.4.1 Преимущества предложенного метода измерений

Предложенный метод измерения геометрии слитков свободен от недостатков других описанных выше методов измерения и имеет следующие преимущества:

простая и надежная механическая конструкция, отсутствие движущихся частей;

исключается ручной труд и ошибки, вызванные человеческим фактором;

не требуется дополнительное место для выполнения измерений, узлы системы размещаются в пределах участка пильной установки;

не требуется дополнительное крановое время для перемещения слитка на площадку измерения и обратно;

процесс измерений не требует дополнительного времени, выполняется в ходе существующего технологического процесса;

полученная в ходе измерения информация представляет собой полный продольный профиль слитка (его трехмерную модель), и является существенно более полной, чем полученная при ручных измерениях. Данная информация может быть использована для диагностики состояния литейных агрегатов.

отсутствует вмешательство в существующий технологический процесс;

используются средства электронной техники и средств КИПиА, аналогичные другим участкам литейного отделения, не требуются затраты на обучение обслуживающего персонала и расширение номенклатуры ЗИП;

система легко внедряется на любом литейном отделении, выпускающем слитки;

система имеет наименьшую стоимость по сравнению с аналогами.

2. Программное обеспечение

2.1. Программное обеспечение нижнего уровня

Программное обеспечение контроллера предназначено для сбора первичной информации, обработки и передачи на верхний уровень системы. Программное обеспечение реализовано в среде разработки Step7 для контроллеров фирмы Siemens серии S7-300.

В шкафу сбора данных установлен контроллер, блоки ввода аналоговых и дискретных сигналов. Лазерные датчики подключены к модулям аналогового и дискретного ввода. На модуль аналогового ввода приходит сигнал о расстоянии от датчика до слитка. На модуль дискретного ввода приходит сигнал о наличии измеряемого объекта в поле действия датчика.

К контроллеру может быть подключено до 16 датчиков контроля геометрии слитка и один датчик измерения длины слитка.

Датчик измерения длины слитка может быть подключен к аналоговому входу (лазерный дальномер) или дискретному вводу быстрого счета (инкрементный датчик).

Измерение геометрии слитка начинается в следующих случаях:

по дискретному сигналу от лазерного датчика при обнаружении слитка в поле действия датчика,

по изменению величины аналогового сигнала, если измеренное расстояние до объекта меньше заданной величины.

Измерение геометрии слитка начинается, если хотя бы для одного из датчиков выполнилось одно из выше описанных условий.

Завершение измерения геометрии слитка происходит, если для всех датчиков не выполняются вышеперечисленные условия.

Описанный алгоритм позволяет произвести измерения слитка лежащего под углом относительно оси движения.

Для каждого аналогового датчика программное обеспечение позволяет задать пределы измерения и границу обнаружения слитка, находится данный датчик в работе или нет.

Опрос всех датчиков производится по прерыванию, что обеспечивает высокую частоту опроса и точность временных интервалов между измерениями. Для каждого аналогового датчика создан кольцевой буфер на 1000 значений, что позволяет хранить данные до их передачи на АРМ оператора. Аналогичным образом реализован кольцевой буфер для хранения точного времени измерения параметров. Ячейки с одинаковым номером во всех кольцевых буферах содержат одномоментные измерения, что позволяет легко синхронизировать полученные данные с верхним уровнем системы. При пропадании связи с АРМ оператора кольцевой буфер может переполниться, при этом новые данные будут записаны поверх самых старых.

При нормальной работе установки, после каждого сканирования слитка программное обеспечение верхнего уровня производит обнуление кольцевых буферов и установку указателей начальной и конечной ячейки не считанных данных на первую ячейку кольцевого буфера. Во время сканирования слитка программное обеспечение верхнего уровня производит загрузку данных с контроллера, начиная от указателя на начальную ячейку не считанных данных, и заканчивая указателем на последнюю ячейку не считанных данных, после чего перемещает указатель начальной ячейки не считанных данных на позицию за последней считанной ячейкой. Положение указателя на последнюю ячейку не считанных данных обновляет контроллер, по мере опроса аналоговых датчиков.

2.2 Программное обеспечение верхнего уровня

2.2.1 Используемые технологии

Для реализации программного обеспечения (ПО) системы автоматической диагностики и отбраковки алюминиевых слитков используется следующий набор технологий.

Серверная часть программного комплекса:

1. Фреймворк Microsoft .Net версии 4.0.
2. Среда разработки Visual Studio 2010.
3. Язык программирования C#.
4. Технология OLE for Process Control (OPC).
5. Технология Windows Communication Foundation (WCF).
6. Система управления базами данных Microsoft SQL.

Клиентская часть программного комплекса:

1. Фреймворк Microsoft .Net версии 4.0.
2. Среда разработки Visual Studio 2010.
3. Язык программирования C#.
4. Графическая библиотека Open Graphics Library (OpenGL).

2.2.2 Технология OLE for Process Control (OPC)

OPC – это набор повсеместно принятых спецификаций, предоставляющих универсальный механизм обмена данными в системах контроля и управления. Аббревиатура OPC традиционно расшифровывается как OLE for Process Control. OLE – Object Linking and Embedding (связывание и встраивание объектов). Технология OPC определяет интерфейс между OPC-клиентом и OPC-серверами .

OPC-сервер – программа, получающая данные во внутреннем формате устройства или системы и преобразующая эти данные в формат OPC. OPC-сервер является источником данных для OPC-клиентов. По своей сути OPC-сервер – это некий универсальный драйвер физического оборудования, обеспечивающий взаимодействие с любым OPC-клиентом.

OPC-клиент – программа, принимающая от OPC-серверов данные в формате OPC.

В проекте автоматического контроля и отбраковки алюминиевых слитков используется OPC-сервер компании Siemens, а также OPC-клиент собственной разработки компании ООО «Альвасофт».

2.2.3 Технология Windows Communication Foundation (WCF)

Windows Communication Foundation (WCF) - программный фреймворк, используемый для обмена данными между распределенными приложениями, входящий в состав Microsoft [.NET Framework](#). До своего выпуска в декабре 2006 года в составе .NET Framework 3.0, WCF был известен под кодовым именем Indigo.

WCF делает возможным построение безопасных и надёжных транзакционных распределенных систем через упрощенную унифицированную программную модель межплатформенного взаимодействия. Комбинируя функциональность существующих технологий .NET по разработке распределённых приложений (ASP.NET XML Web Services – ASMX, WSE 3.0, .NET Remoting, [.NET Enterprise Services](#) и [System.Messaging](#)), WCF предоставляет единую инфраструктуру разработки, повышающую производительность и снижающую затраты на создание безопасных, надежных и транзакционных распределенных служб. Заложенные в неё принципы позволяют организовать работу с другими платформами, для чего используются технологии взаимодействия, например, WSIT, разрабатываемые на базе открытого исходного кода.[11]

2.2.4 Схема взаимодействия компонентов программного и аппаратного обеспечения

Аппаратный комплекс системы состоит из следующих частей:

1. Датчики расстояния, установленные на измерительной раме.

Программируемый логический контроллер шкафа управления Siemens S7-315.

Сервер.

Подключаемые клиенты.

Схема взаимодействия компонентов аппаратного комплекса изображена на рисунке 7.

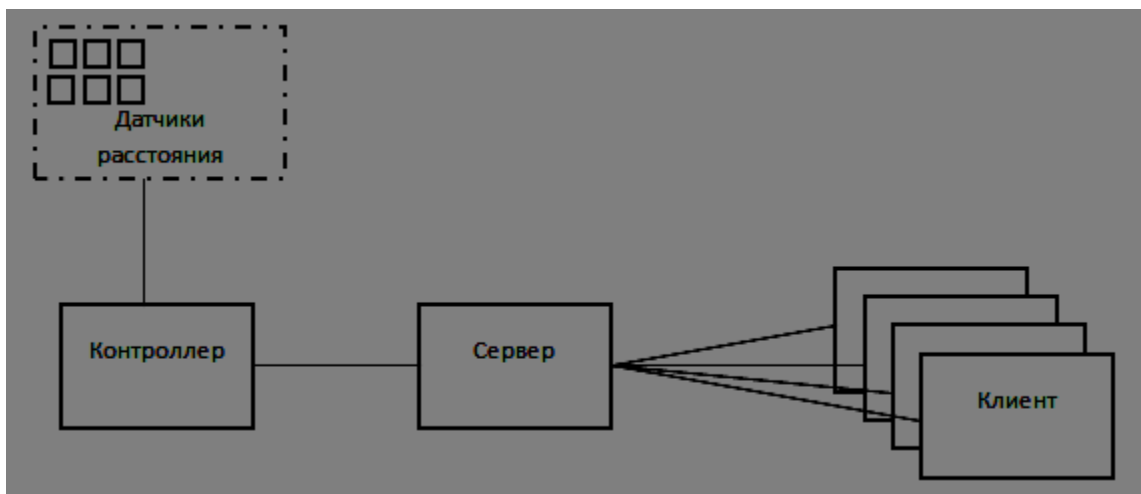


Рисунок 7 - Схема взаимодействия элементов аппаратного комплекса.

Программное обеспечение серверной и клиентской части комплекса имеет схему взаимодействия, изображенную на рисунке .

Взаимодействие сервера с контроллером происходит посредством протокола OPC по TCP-каналу. Клиенты подключаются к серверу по протоколу SOAP, реализованному в рамках технологии WCF. Серверное и клиентское программное обеспечение имеет в своем составе одинаковое определение WCF-контракта, что позволяет им взаимодействовать друг с другом, используя общий API-интерфейс.

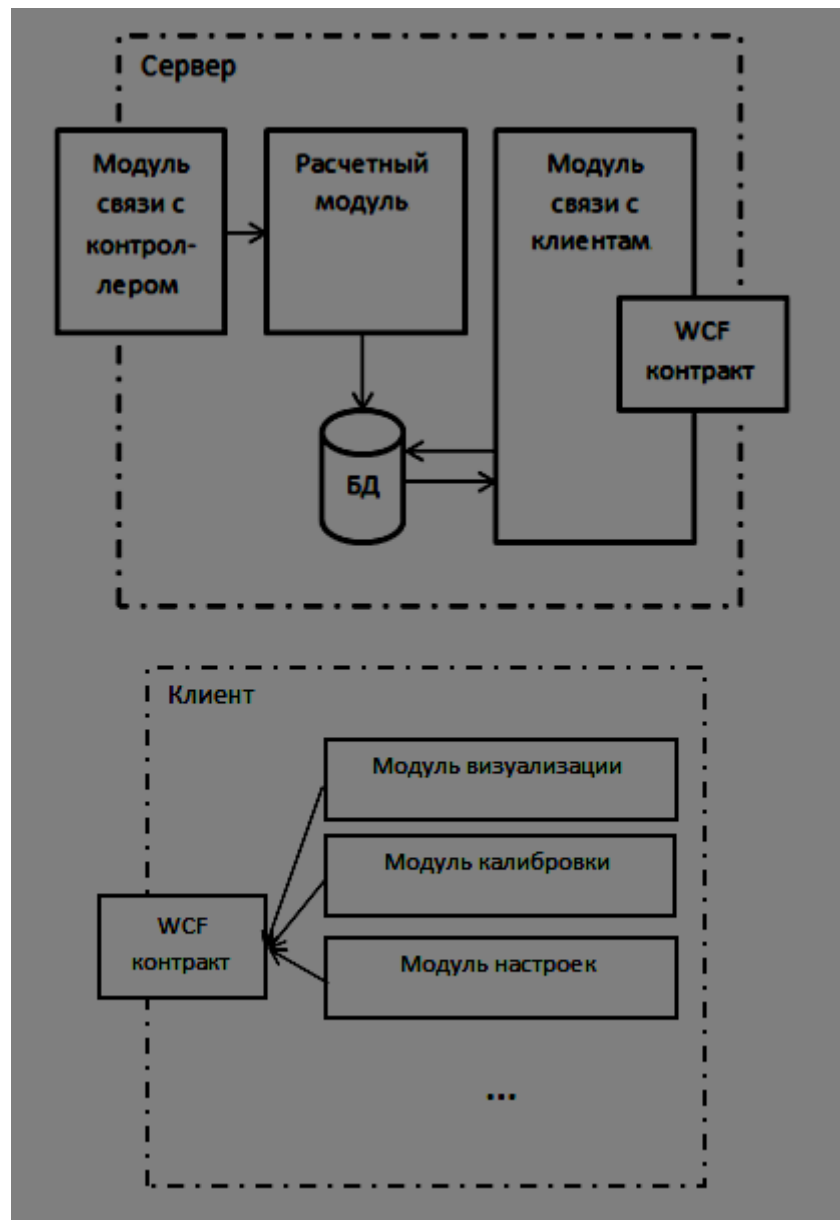


Рисунок 8 - Схема взаимодействия серверного и клиентского программного обеспечения.

2.2.5 Модуль сервера

2.2.5.1 Настройки серверной части

Программное обеспечение серверной части может запускаться как консольное приложение, либо как служба Windows.

Перед запуском сервера выполняется конфигурация системы для ее корректной работы. Настройки записаны в следующих файлах формата xml.

SensorsConfiguration.xml – настройки сканирующей рамки и расположения датчиков на ней. Для каждого датчика необходимо задать его тип,

сторону сканирования, смещение и идентификатор. По этим параметрам расчетный модуль определяет как использовать полученные данные.

OPCConfiguration.xml – настройки для подключения к OPC-серверу. Здесь задаются имена OPC-тегов для каждого датчика, а также для управляющего блока.

CalibratorConfiguration.xml – настройки модуля калибровки для всех датчиков. Калибровка датчиков выполняется перед началом работы сканирующей системы.

Hibernate.cfg.xml – настройки для работы с базой данных.

Logging.xml – настройки записей лог-файлов во время работы сервера.

2.2.5.2 Структура серверной части

Серверная часть состоит из многих элементов, которые не зависят друг от друга, либо зависят слабо. За счет этого архитектурного принципа достигается гибкость программного обеспечения, простота и скорость внесения изменений и дополнений.

Упрощенная UML диаграмма серверной части изображена на рисунке 9.

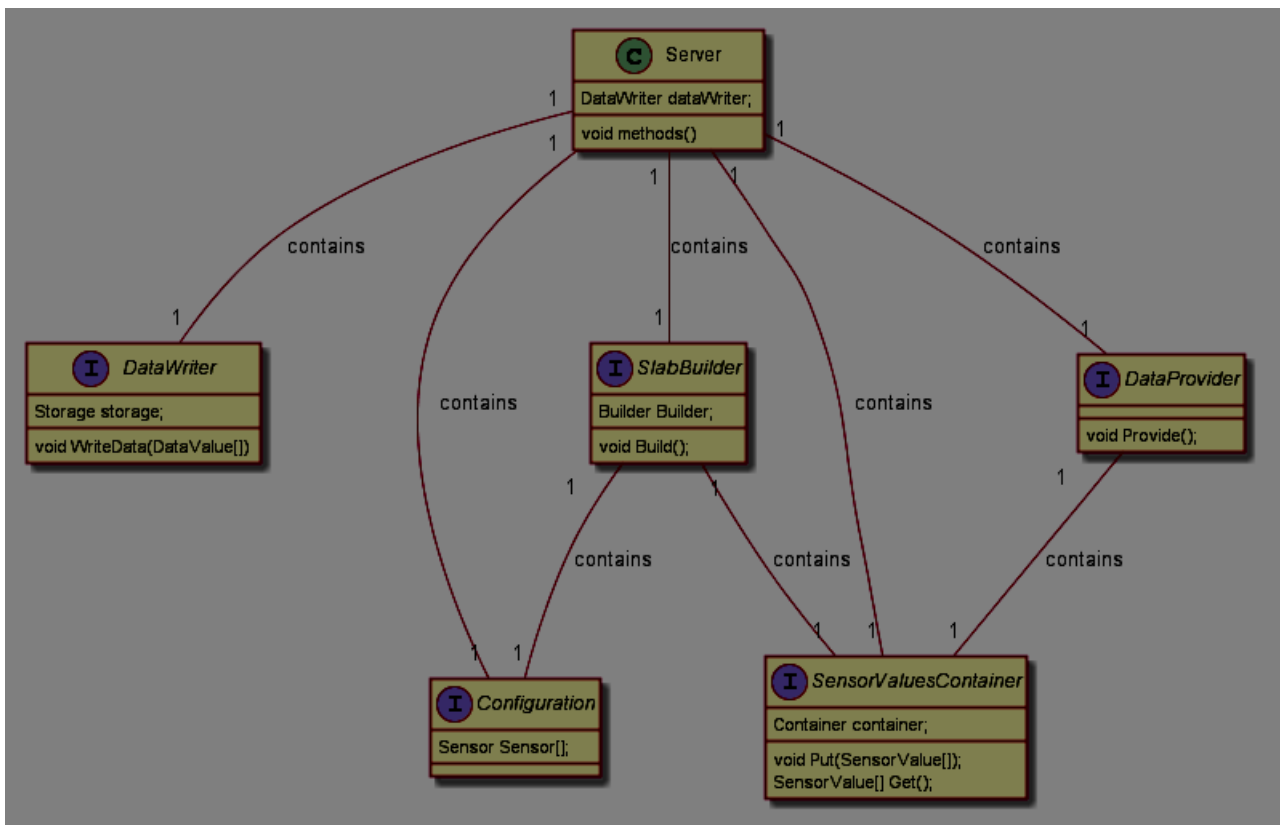


Рисунок 9 - Упрощенная диаграмма ПО сервера.

2.2.6 Краткое описание алгоритмов обработки данных

Данные, которые поступают с контроллера во время сканирования слитка, имеют линейный вид, а также включают много шумов. Для того, чтобы построить модель слитка, необходимо применять фильтры и специальные алгоритмы, которые будут учитывать посторонние движения слитка, его вибрацию, покачивания, подпрыгивания на рольгангах, откаты назад и т.д.

Программные алгоритмы и фильтры – это самая важная часть всего программно-аппаратного комплекса.

К самым простым фильтрам, которые используются при построении модели слитка, относятся:

Модифицированный медианный фильтр – один из видов цифровых фильтров, широко используемый в цифровой обработке сигналов и изображений для уменьшения уровня шума. Медианный фильтр является нелинейным КИХ-фильтром. Значения точек внутри окна фильтрации

сортируются в порядке возрастания и точка, соответствующая середине окна, заменяется другим значением, попадающим в центр отсортированного массива.

Этот метод влияет на пики в наименьшей степени, хорошо сглаживает базовую линию, не меняет форму пика на склонах и очень эффективно устраняет отдельные выбросы (в этом случае выброс заменяется на одну из соседних точек).

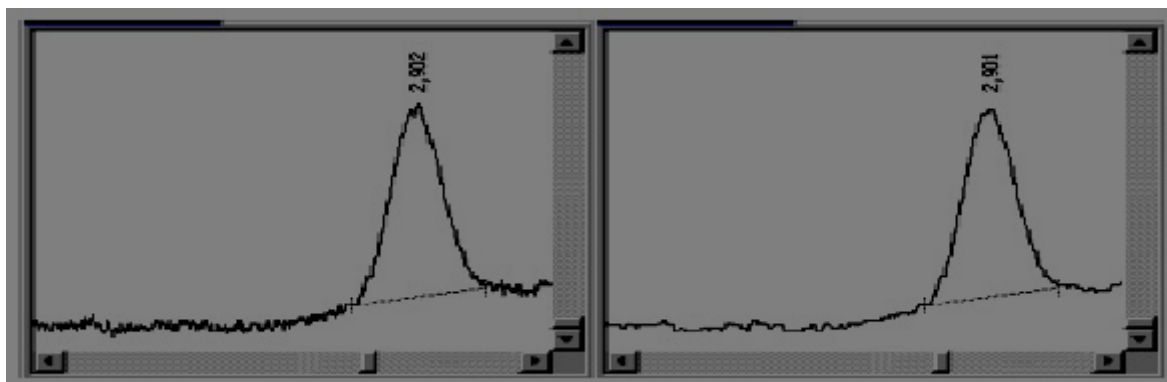


Рисунок 10 - Пример функции до и после медианной фильтрации.

Линейный фильтр – динамическая система, применяющая некий линейный оператор ко входному сигналу для выделения или подавления определённых частот сигнала и других функций по обработке входного сигнала. Линейные фильтры широко применяются в электронике, цифровой обработке сигналов и изображений, в оптике, теории управления и других областях.

Кроме простейших фильтров, применяются более сложные и специфичные алгоритмы, которые позволяют построить модель слитка, даже если во время сканирования происходили подпрыгивания, вибрация и т.д. На рисунке 10 изображены показания верхнего и нижнего датчиков, где видно место незначительного подпрыгивания слитка при наезде на рольганг.

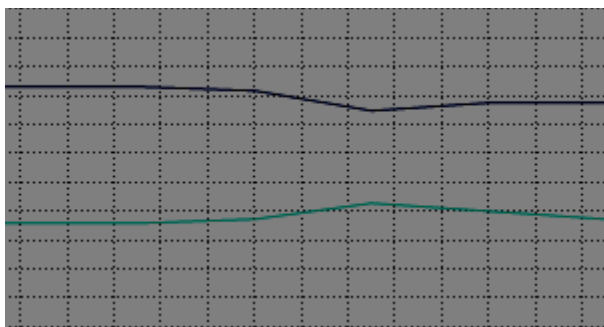


Рисунок 11 - Иллюстрация незначительного подпрыгивания слитка во время движения.

2.2.7 Хранение данных

Все данные, поступающие от контроллера, а также данные расчетов и параметров слитка, сохраняются в локальной базе данных, которая управляется СУБД Microsoft SQL.

Microsoft SQL Server – система управления реляционными базами данных (СУРБД), разработанная корпорацией Microsoft. Основным используемым языком запросов – Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

В среднем, для слитка длиной 5 м, 5 датчиков (4 датчика расстояния на рамке и 1 датчик положения) генерируют 50000 точек.

При сохранении в БД каждая точка характеризуется собственным идентификатором, временем измерения, идентификатором датчика и значением. Таким образом, каждая точка занимает в памяти 24 байта. Соответственно, 50000 точек занимают 1.14 мегабайта.

3. Алгоритмы интерполяции замкнутых кривых.

Интерполяция - построение кривой, проходящей через контрольные точки и обладающей некоторыми дополнительными свойствами (часто гладкостью); [1]

В ходе исследования были рассмотрены следующие алгоритмы для построения замкнутых кривых:

- периодические кривые Безье
- периодические дискретные сплайны
- составные сплайновые кривые Catmull-Rom

3.1 Периодические кривые Безье

Типы кривых, предложенные в 60-х годах XX века независимо друг от друга Пьером Безье из автомобилестроительной компании “Рено” и Полем де Кастельжо из компании “Ситроен”, где применялись для проектирования кузовов автомобилей.

Несмотря на то, что открытие де Кастельжо было сделано несколько ранее Безье (1959), его исследования не публиковались и скрывались компанией как производственная тайна до конца 1960-х.

Кривая Безье является частным случаем многочленов Бернштейна, описанных Сергеем Натановичем Бернштейном в 1912 году.

Впервые кривые были представлены широкой публике в 1962 году французским инженером Пьером Безье, который, разработав их независимо от де Кастельжо, использовал их для компьютерного проектирования автомобильных кузовов. Кривые были названы именем Безье, а именем де Кастельжо назван разработанный им рекурсивный способ определения кривых (алгоритм де Кастельжо).

Впоследствии это открытие стало одним из важнейших инструментов систем автоматизированного проектирования и программ компьютерной графики.[1]

Кривая Безье — параметрическая кривая, задаваемая выражением

$$\mathbf{B}(t) = \sum_{i=0}^n \mathbf{P}_i \mathbf{b}_{i,n}(t), \quad 0 \leq t \leq 1$$

где \mathbf{P}_i — функция компонент векторов опорных вершин, а $\mathbf{b}_{i,n}(t)$ — базисные функции кривой Безье, называемые также полиномами Бернштейна.

$$\mathbf{b}_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

где $\binom{n}{i} = \frac{n!}{i!(n-i)!}$ — число сочетаний из n по i , где n — степень полинома, i — порядковый номер опорной вершины.

При использовании кривых Безье для интерполяции предвидится большое количество вычислений тригонометрических полиномов, что в свою очередь скажется на быстродействии конечного программного продукта, в связи с этим для было принято решение не рассматривать их как дальнейшую перспективу.[2][3]

3.2 Сплайны Катмалла-Рома (Catmull-Rom splines)

Представляют собой семейство кубических интерполяционных сплайнов, сформулированных так, что касательная в каждой точке p_i , вычисляется с использованием предыдущей и следующей точки на сплайне $\tau(p_{i+1} - p_{i-1})$.

Геометрическая матрица задана следующим образом:

$$p(s) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\tau & 0 & \tau & 0 \\ 2\tau & \tau - 3 & 3 - 2\tau & -\tau \\ -\tau & 2 - \tau & \tau - 2 & \tau \end{bmatrix} \begin{bmatrix} p_{i-2} \\ p_{i-1} \\ p_i \\ p_{i+1} \end{bmatrix}$$

Непрерывны в пространстве комплексных чисел \mathbb{C} , интерполируемы, но не лежат на выпуклой оболочке.

Заметим, что касательная в точке p_0 не определена четко, обычно она задается как $\tau(p_1 - p_{-1})$. Но это не обязательно, можно предположить что кривая не интерполируется в конечных точках.

Параметр τ (Натяжение) влияет на сглаживание кривой между контрольными точками (рисунок 12). Обычно значение натяжения берется за 0.5, но его можно варьировать в зависимости от задачи.

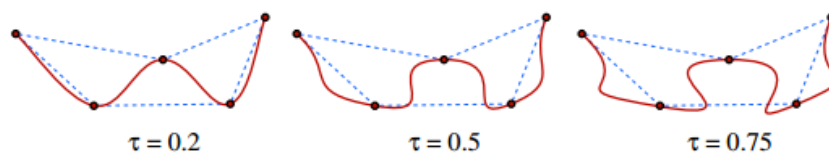


Рисунок 12 - влияния параметра τ (натяжения).

Рассмотрим один сегмент, $p(s)$. Предположим, что он определяется четырьмя контрольными точками, p_{i-2} , p_{i-1} , p_i , p_{i+1}

как на рисунке 13. Мы знаем, что поскольку он кубический, его можно выразить полиномиальной формой:

$$\mathbf{p}(s) = \mathbf{c}_0 + \mathbf{c}_1 u + \mathbf{c}_2 u^2 + \mathbf{c}_3 u^3 \quad (1)$$

$$= \sum_{k=0}^3 \mathbf{c}_k u^k \quad (2)$$

Теперь нам нужно выразить некоторые ограничения. Рассматривая рисунок 13, мы находим следующее:

$$\begin{aligned} \mathbf{p}(0) &= \mathbf{p}_{i-1} \\ \mathbf{p}(1) &= \mathbf{p}_i \\ \mathbf{p}'(0) &= \tau(\mathbf{p}_i - \mathbf{p}_{i-2}) \\ \mathbf{p}'(1) &= \tau(\mathbf{p}_{i+1} - \mathbf{p}_{i-1}) \end{aligned}$$

(Вы должны убедиться, что понимаете, откуда они взялись). Теперь мы можем объединить эти ограничения с (2) получить следующее,

$$\mathbf{c}_0 = \mathbf{p}_{i-1} \quad (3)$$

$$\mathbf{c}_0 + \mathbf{c}_1 + \mathbf{c}_2 + \mathbf{c}_3 = \mathbf{p}_i \quad (4)$$

$$\mathbf{c}_1 = \tau(\mathbf{p}_i - \mathbf{p}_{i-2}) \quad (5)$$

$$\mathbf{c}_1 + 2\mathbf{c}_2 + 3\mathbf{c}_3 = \tau(\mathbf{p}_{i+1} - \mathbf{p}_{i-1}) \quad (6)$$

Теперь мы можем подставить (3) и (5) в (4) и (6), чтобы получить уравнения, в которых \mathbf{c}_2 и \mathbf{c}_3 являются единственными переменными

$$\mathbf{c}_2 + \mathbf{c}_3 = (\mathbf{p}_i - \mathbf{p}_{i-1}) - \tau(\mathbf{p}_i - \mathbf{p}_{i-2})$$

$$2\mathbf{c}_2 + 3\mathbf{c}_3 = \tau(\mathbf{p}_{i+1} - \mathbf{p}_{i-1}) - \tau(\mathbf{p}_i - \mathbf{p}_{i-2})$$

Теперь должно быть ясно, что, вычитая эти два уравнения, мы можем легко решить для \mathbf{c}_2 и \mathbf{c}_3 в терминах из \mathbf{p}_j и τ (который также задан пользователем). Мы получаем

$$\mathbf{c}_0 = \mathbf{p}_{i-1}$$

$$\mathbf{c}_1 = \tau(\mathbf{p}_i - \mathbf{p}_{i-2})$$

$$\mathbf{c}_2 = 3(\mathbf{p}_i - \mathbf{p}_{i-1}) - \tau(\mathbf{p}_{i+1} - \mathbf{p}_{i-1}) - 2\tau(\mathbf{p}_i - \mathbf{p}_{i-2})$$

$$\mathbf{c}_3 = -2(\mathbf{p}_i - \mathbf{p}_{i-1}) + \tau(\mathbf{p}_{i+1} - \mathbf{p}_{i-1}) + \tau(\mathbf{p}_i - \mathbf{p}_{i-2})$$

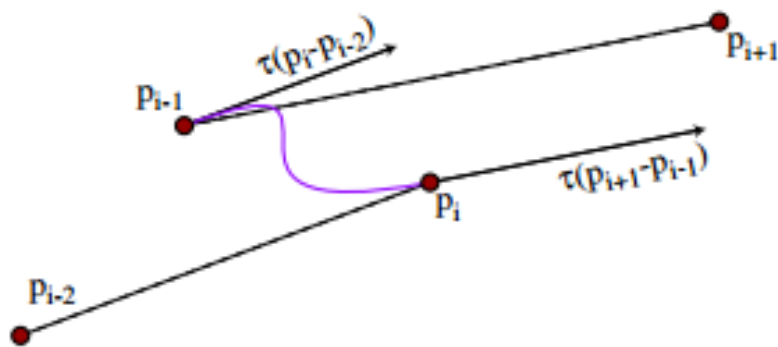


Рисунок 13 - процесс вычисления сплайна.

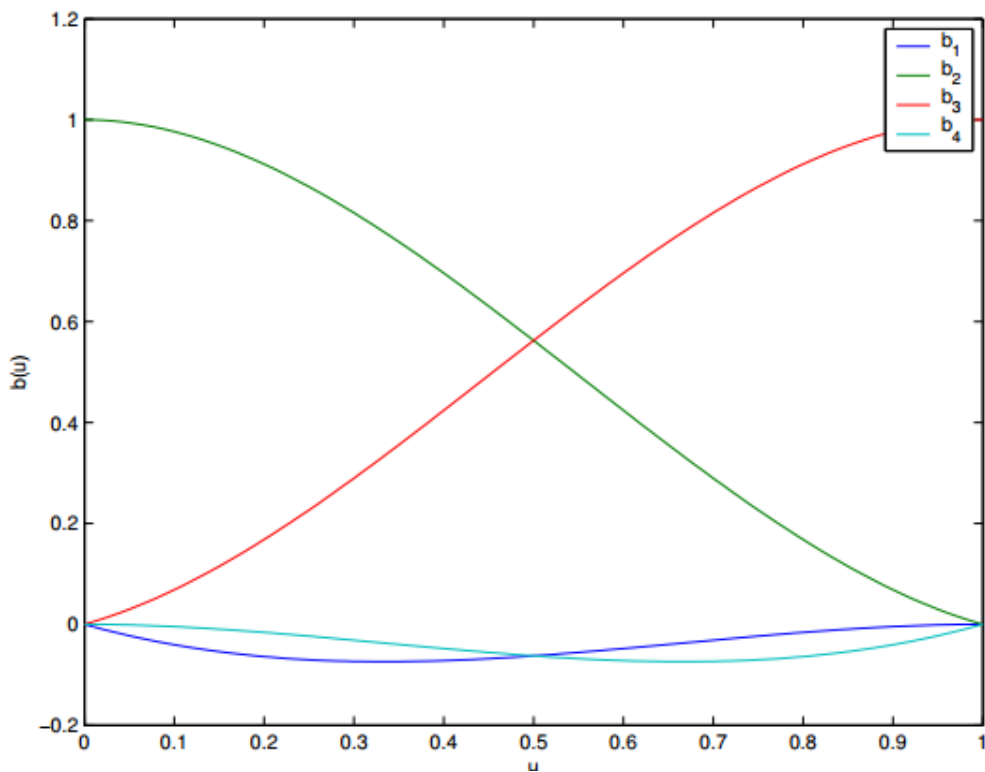


Рисунок 14 - смежные функции для $\tau = 1/2$.

Теперь, если мы хотим сформировать базовую матрицу из этого, мы должны сгруппировать все P_j слагаемые вместе,

$$\mathbf{c}_0 = \mathbf{P}_{i-1}$$

$$\mathbf{c}_1 = (-\tau)\mathbf{P}_{i-2} + (\tau)\mathbf{P}_i$$

$$\mathbf{c}_2 = (2\tau)\mathbf{P}_{i-2} + (\tau - 3)\mathbf{P}_{i-1} + (3 - 2\tau)\mathbf{P}_i + (-\tau)\mathbf{P}_{i+1}$$

$$\mathbf{c}_3 = (-\tau)\mathbf{P}_{i-2} + (2 - \tau)\mathbf{P}_{i-1} + (\tau - 2)\mathbf{P}_i + (\tau)\mathbf{P}_{i+1}$$

Из этого должно быть достаточно ясно, как заполнить геометрическую матрицу,

$$\mathbf{p}(s) = \mathbf{u}^T \mathbf{M} \mathbf{p} = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\tau & 0 & \tau & 0 \\ 2\tau & \tau - 3 & 3 - 2\tau & -\tau \\ -\tau & 2 - \tau & \tau - 2 & \tau \end{bmatrix} \begin{bmatrix} \mathbf{p}_{i-2} \\ \mathbf{p}_{i-1} \\ \mathbf{p}_i \\ \mathbf{p}_{i+1} \end{bmatrix}$$

Отсюда мы также можем получить наши функции смешивания (см. Рис. 14)[4][5][6],

$$\mathbf{M} \mathbf{u} = \begin{bmatrix} -\tau u + 2\tau u^2 - \tau u^3 \\ 1 + (\tau - 3)u^2 + (2 - \tau)u^3 \\ \tau u + (3 - 2\tau)u^2 + (\tau - 2)u^3 \\ -\tau u^2 + \tau u^3 \end{bmatrix}$$

3.2.1 Исследование алгоритма построения Сплайнов Катмалла-Рома

Для исследования было разработано небольшое приложение, с графическим интерфейсом наглядно демонстрирующее работу алгоритма.

Пример реализации алгоритма выглядит следующим образом:

```
public class CatmullRomSpline {
    /// <summary>
    /// Вычисляет узловые CatmullRom-сплайна.
    ///
    https://www.cs.cmu.edu/~462/projects/assn2/assn2/catmullRom.pdf
    /// </summary>
    /// <param name="aPoints">Полюса сплайна (исходные точки). Должно
    быть не менее 2-х полюсов.</param>
    /// <param name="aTension">Натяжение сплайна.</param>
    /// <param name="n">Число узлов между полюсами сплайна.</param>
    /// <param name="aIsClosedCurve">True - сплайн будет замкнут,
    false - иначе.</param>
    /// <returns></returns>
    public static Vector2[] Calculate(Vector2[] aPoints, double
    aTension, int n = 5, bool aIsClosedCurve = true) {
        if (aPoints == null) {
            throw new ArgumentNullException("aPoints");
        }
        if (aPoints.Length <= 2) {
            throw new ArgumentException("Число полюсов должно быть >
2.");
        }
        if (n < 1) {
            throw new ArgumentException("Число узлов между полюсами
сплайна должно быть >= 1.");
        }
        var N = aPoints.Length;

        var vectors = BuildVectors(aPoints, aIsClosedCurve);
        var resultSpline = new List<Vector2>();
        for (var i = 1; i < vectors.Length - 2; ++i) {
            var singleSpline = CalculateSpline(aTension, n, vectors[i -
1], vectors[i], vectors[i + 1], vectors[i + 2]);
            resultSpline.AddRange(singleSpline);
        }

        return resultSpline.ToArray();
    }
}
```



```

/// <summary>
/// Рассчитывает вектора для одного участка сплайна, ограниченного
p0, p1, p2 и p3.
/// </summary>
/// <param name="aTension">Натяжение.</param>
/// <param name="n">Количество узлов между полюсами.</param>
/// <param name="p0">Полюс 1.</param>
/// <param name="p1">Полюс 2.</param>
/// <param name="p2">Полюс 3.</param>
/// <param name="p3">Полюс 4.</param>
/// <returns>Вектора сплайна.</returns>
private static Vector2[] CalculateSpline(double aTension, int n,
Vector2 p0, Vector2 p1, Vector2 p2, Vector2 p3) {
    var resultVectors = new Vector2[n + 1];
    var vectorIndex = 0;
    var step = 1.0 / n;
    for (var t = 0; t < n; ++t) {
        var vector = ReturnCarmullRomVector(t * step, aTension,
p0, p1, p2, p3);
        resultVectors[vectorIndex++] = vector;
    }
    resultVectors[vectorIndex] = new Vector2(p2);

    return resultVectors;
}

/// <summary>
/// Рассчитывает конкретный вектор для заданного шага.
///
https://www.cs.cmu.edu/~462/projects/assn2/assn2/catmullRom.pdf
(Страница 2).
/// </summary>
/// <param name="t">Шаг.</param>
/// <param name="p0">Полюс 1.</param>
/// <param name="p1">Полюс 2.</param>
/// <param name="p2">Полюс 3.</param>
/// <param name="p3">Полюс 4.</param>
/// <returns>Вектор.</returns>
private static Vector2 ReturnCarmullRomVector(double dt, double
aTension, Vector2 p0, Vector2 p1, Vector2 p2, Vector2 p3) {
    Vector2 a = p1;
    Vector2 b = aTension * (p2 - p0);
    Vector2 c = 3f * (p2 - p1) - aTension * (p3 - p1) - 2f *
aTension * (p2 - p0);
    Vector2 d = -2f * (p2 - p1) + aTension * (p3 - p1) + aTension
* (p2 - p0);

```

```
    Vector2 result = a + (b * dt) + (c * dt * dt) + (d * dt * dt
* dt);
    return result;
    }
}
```

В ходе исследования проверялось поведение алгоритма для объединения восьми точек в одну замкнутую кривую, изначально параметр натяжения τ , был выбран величиной в 0,5, а количество узлов(количество точек между целевыми) 8.

На рисунке 15 продемонстрирована его работа.

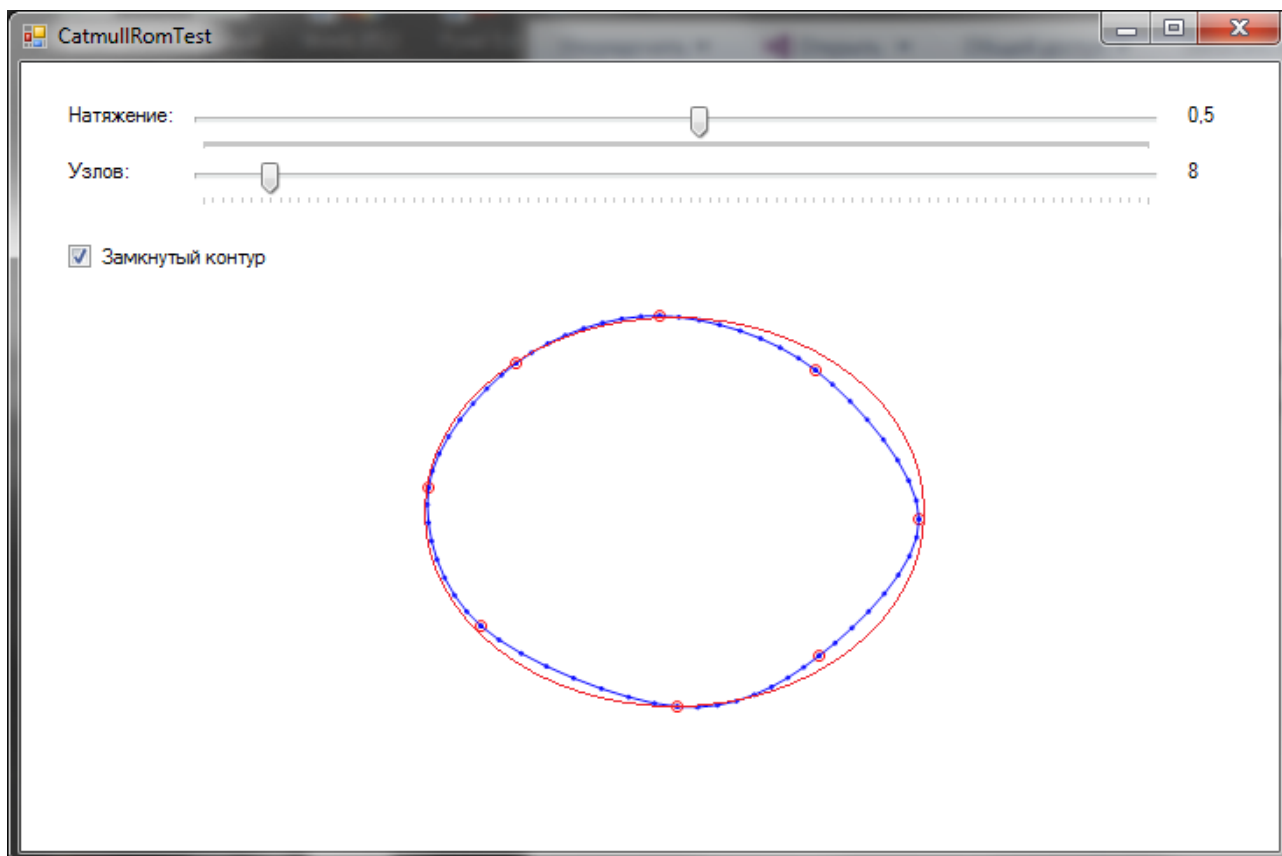


рисунок 15 - исследование алгоритма.

Результаты не удовлетворительны, поскольку полученная замкнутая кривая, не имеет необходимой округлой формы. При увеличении или уменьшении параметра натяжения, результаты остаются такими же. (рисунок 16, 17, 18)

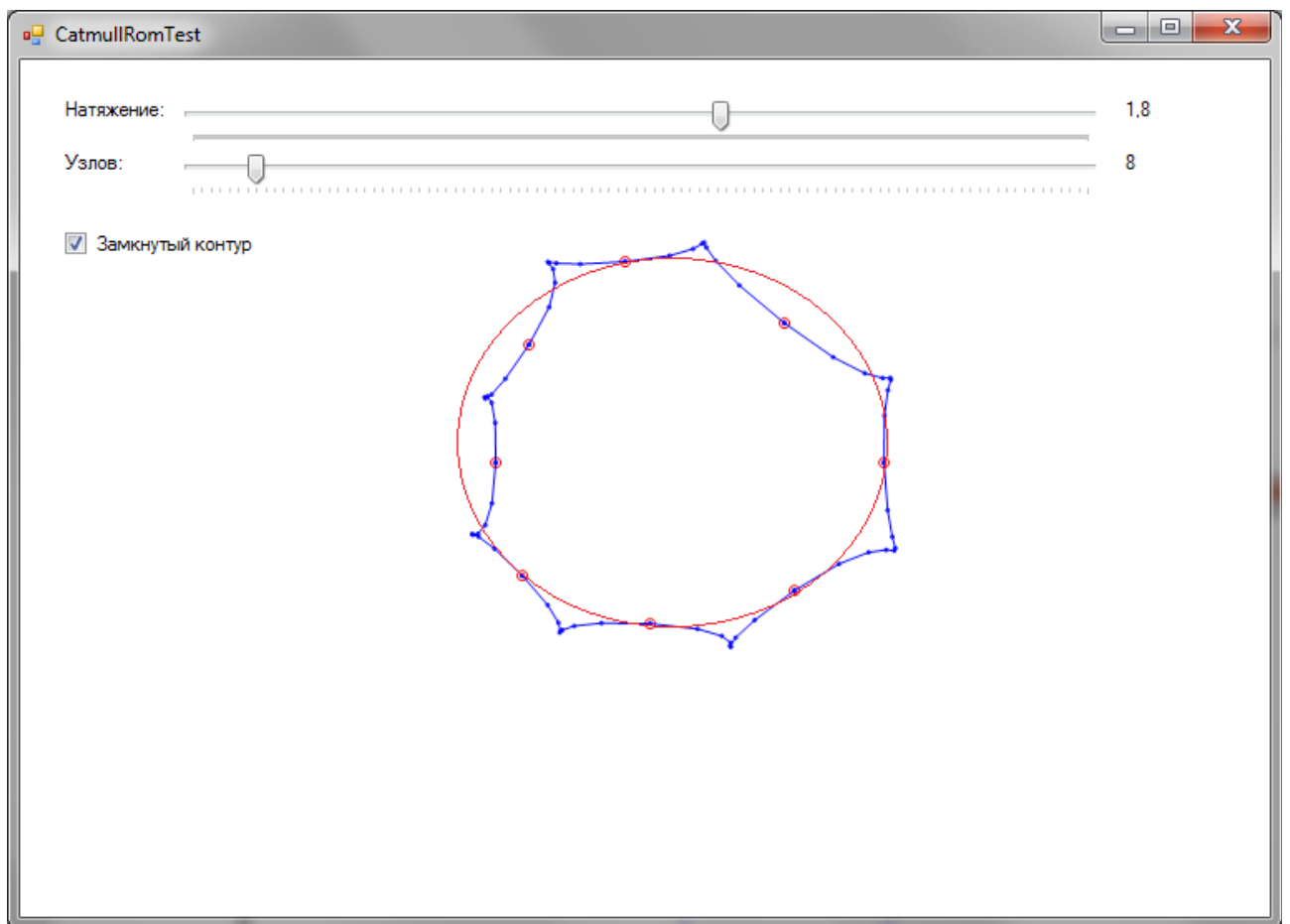


рисунок 16- Увеличение параметра τ .

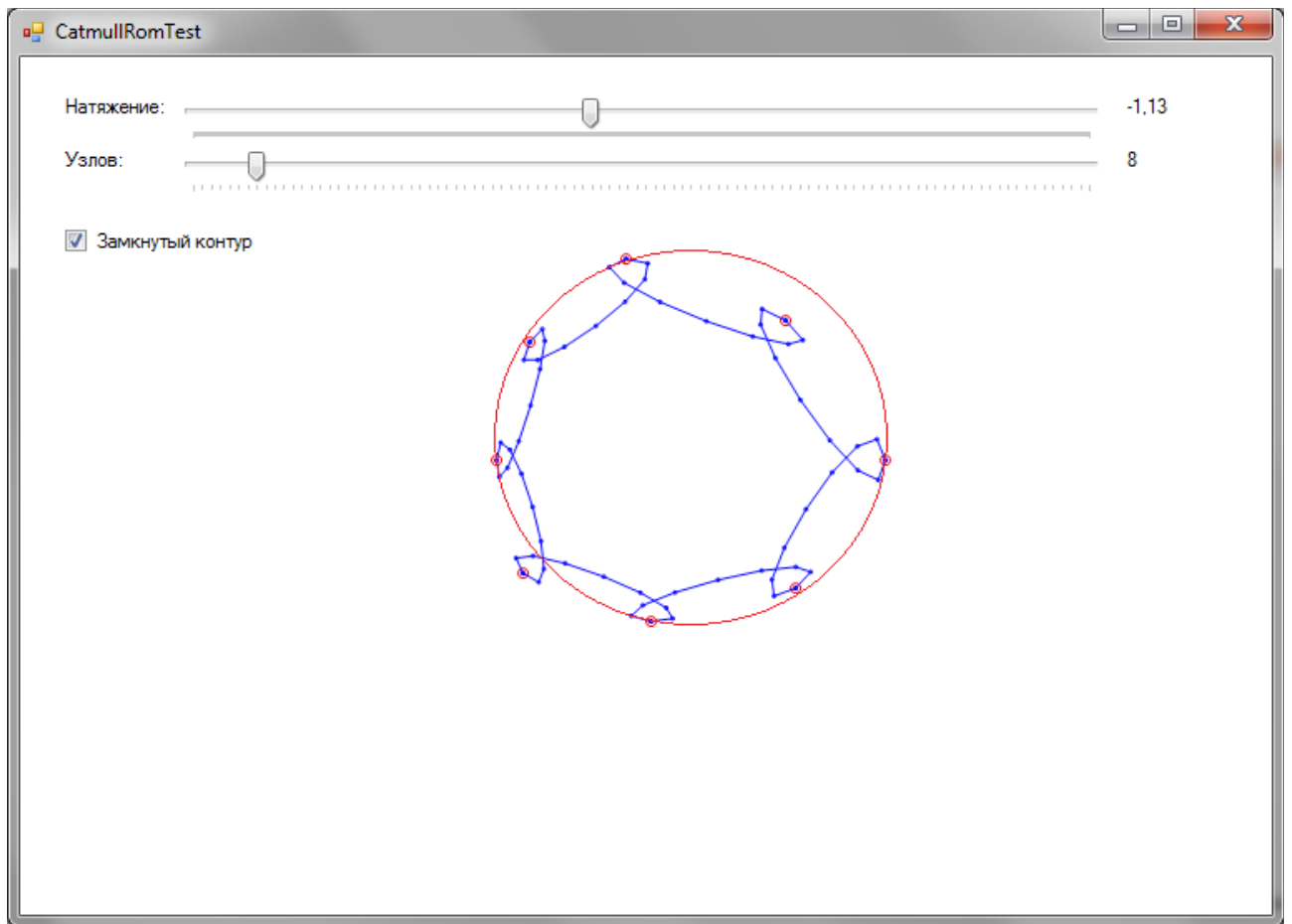


рисунок 17 - уменьшение параметра τ .

Увеличение количества узлов между контрольными точками, также не дает желаемого результата.

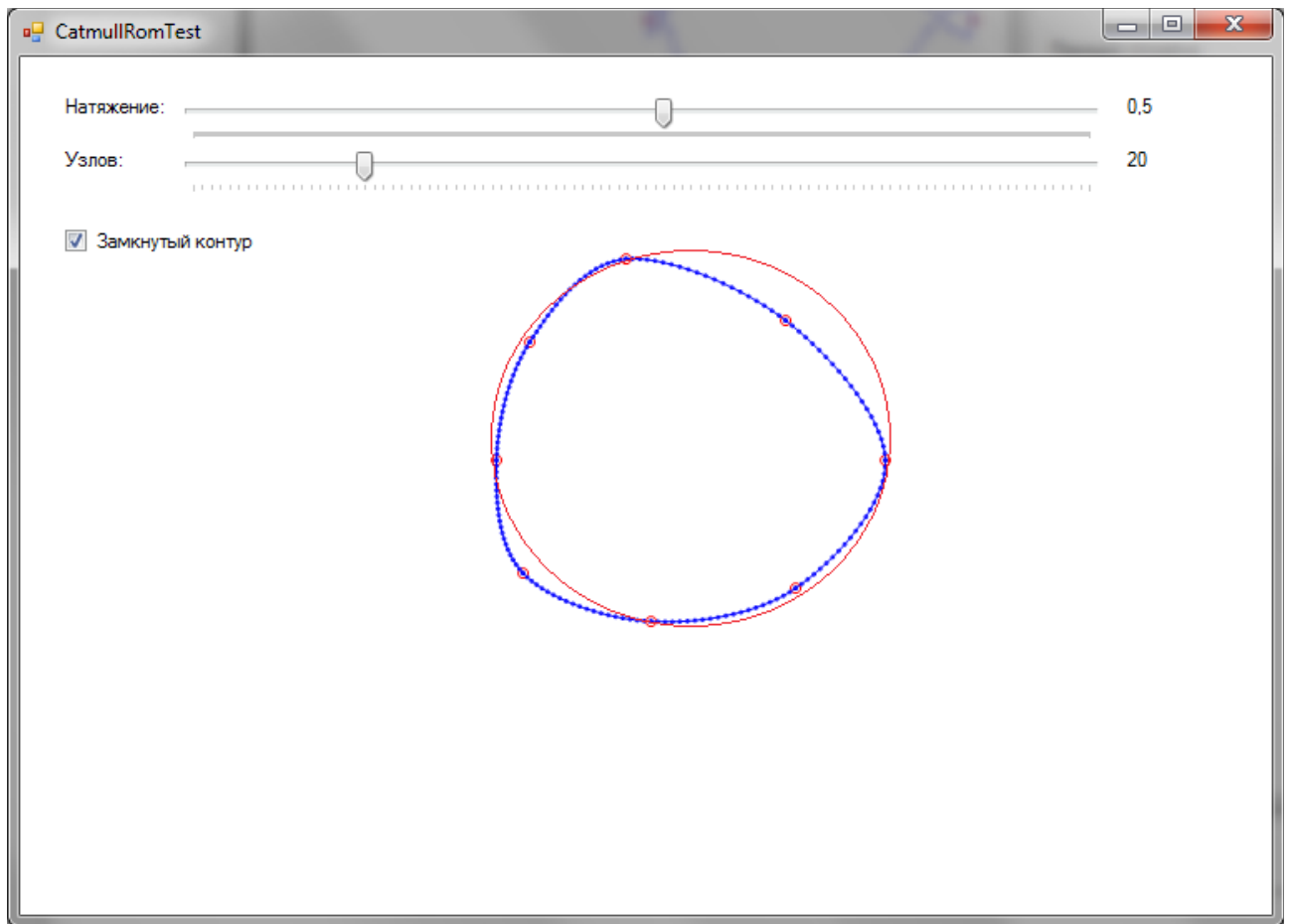


рисунок 18 - увеличение количества узлов.

3.3 Дискретные периодические сплайны с векторными коэффициентами

Пусть $N = mn$, где m, n — натуральные числа, отличные от единицы. В-сплайн первого порядка на основном периоде $0 : N - 1$ задается явно:

$$Q_1(j) = \begin{cases} n - j & \text{при } j \in 0 : n - 1, \\ 0 & \text{при } j \in n : N - n, \\ j - N + n & \text{при } j \in N - n + 1 : N - 1. \end{cases} \quad (7)$$

В-сплайны более высоких порядков определяются с помощью циклической свертки.

$$Q_\nu = Q_1 * Q_{\nu-1}, \quad \nu = 2, 3, \dots \quad (8)$$

Отметим, что В-сплайн $Q_r(j)$ принимает только целые неотрицательные значения. Более глубоким является следующее свойство:

$$\sum_{p=0}^{m-1} Q_r(j - pn) \equiv n^{2r-1}. \quad (9)$$

Введем нормализованные В-сплайны

$$\tilde{Q}_r(j) = \frac{1}{n^{2r-1}} Q_r(j), \quad j \in \mathbb{Z}. \quad (10)$$

Для них формулы (7) – (9) примут вид

$$\tilde{Q}_1(j) = \begin{cases} 1 - j/n & \text{при } j \in 0 : n - 1, \\ 0 & \text{при } j \in n : N - n, \\ j/n - m + 1 & \text{при } j \in N - n + 1 : N - 1; \end{cases} \quad (11)$$

$$\tilde{Q}_\nu = \frac{1}{n} (\tilde{Q}_1 * \tilde{Q}_{\nu-1}), \quad \nu = 2, 3, \dots; \quad (12)$$

$$\sum_{p=0}^{m-1} \tilde{Q}_r(j - pn) \equiv 1. \quad (13)$$

Результирующий сплайн вычисляется следующим способом:[7][8][9][10]

$$\mathbf{S}_1(j) = \sum_{p=0}^{m-1} \mathbf{a}_p \tilde{Q}_{1,n}(j - pn),$$

$$\mathbf{S}_\nu(j) = \frac{1}{n} \sum_{k=0}^{N-1} \tilde{Q}_{1,n}(k) \mathbf{S}_{\nu-1}(j - k), \quad \nu = 2, 3, \dots$$

3.3 Исследование алгоритма построения дискретных периодических сплайнов с векторным коэффициентом.

Исследование заключается в изучении поведения алгоритма, для этого было разработано программное обеспечение реализующее данный алгоритм.

Листинг программного модуля реализующего алгоритм:

```
/// <summary>
/// Дискретный периодический сплайн с векторными коэффициентами.
/// </summary>
public static class DpSpline {
    // Для кеширования q-сплайна.
    private static int previousPoleCount = -1;
    private static int previousPointsBeetwinPoleCount = -1;
    private static double[] qSpline;
    /// <summary>
    /// Вычисляет узловые точки дискретного N-периодического сплайна с
    векторными коэффициентами.
    /// </summary>
    /// <param name="aPoints">Полюса сплайна (исходные точки). Должно
    быть не менее 2-х полюсов.</param>
    /// <param name="r">Порядок сплайна.</param>
    /// <param name="n">Число узлов между полюсами сплайна.</param>
    /// <param name="aIsIncludeOriginalPoints">True - сплайн будет
    проходить через полюса, false - сплайн не будет проходить через
    полюса.</param>
    /// <returns></returns>
    public static Vector2[] Calculate(Vector2[] aPoints, int r, int n
    = 5, bool aIsIncludeOriginalPoints = true) {
        if (aPoints == null) {
            throw new ArgumentNullException("aPoints");
        }
        if (aPoints.Length <= 2) {
            throw new ArgumentException("Число полюсов должно быть >
2.");
        }

        if (r <= 0) {
            throw new ArgumentException("Порядок сплайна должен быть >
0.");
        }

        if (n < 1) {
            throw new ArgumentException("Число узлов между полюсами сплайна
должно быть >= 1.");
        }
        var m = aPoints.Length;
        var N = n * m;
        Vector2[] vectors;
        if (aIsIncludeOriginalPoints) {
            vectors = RecalculateVectors(aPoints, r, n, m);
        }
    }
}
```

```

    } else {
        vectors = new Vector2[m];
        aPoints.CopyTo(vectors, 0);
    }
    if (n != previousPointsBeetwinPoleCount || m !=
previousPoleCount) {
        previousPointsBeetwinPoleCount = n;
        previousPoleCount = m;
        qSpline = CalculateQSpline(n, m);
    }

    var resultPoints = CalculateSSpline(vectors, qSpline, r, n, m);

    return resultPoints;
}
/// <summary>
/// Вычисляет вектора дискретного периодического сплайна с
векторными коэффициентами, согласно
/// формулам
http://www.math.spbu.ru/ru/mneh/AspDok/pub/2010/Chashnikov.pdf
(страница 7).
/// </summary>
/// <param name="vectors"></param>
/// <param name="qSpline"></param>
/// <param name="r"></param>
/// <param name="n"></param>
/// <param name="m"></param>
/// <returns></returns>

```

Ход исследования такой же как и ход исследования алгоритма построения сплайнов Катмалла-Рома. Количество узлов между контрольными точками 8, а изначальный порядок сплайна равен 2.

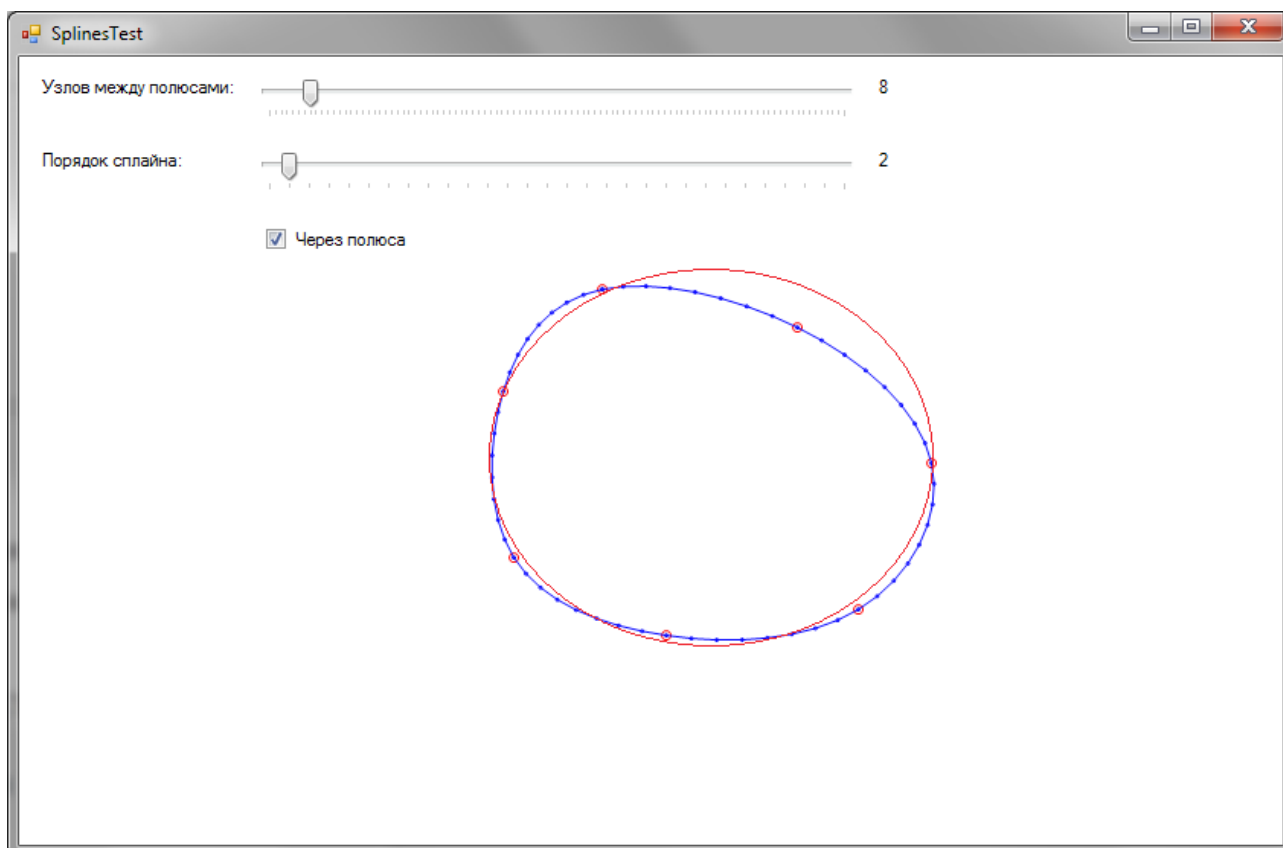


рисунок 19 - исследования алгоритма построения дискретных периодических сплайнов.

Из рисунка 19 видно, что данный алгоритм дает отличный результат, построенная с помощью него кривая округлая, данная кривая будет достаточно точно описывать срез слитка.

Изменим входные параметры алгоритма для полноты картины.

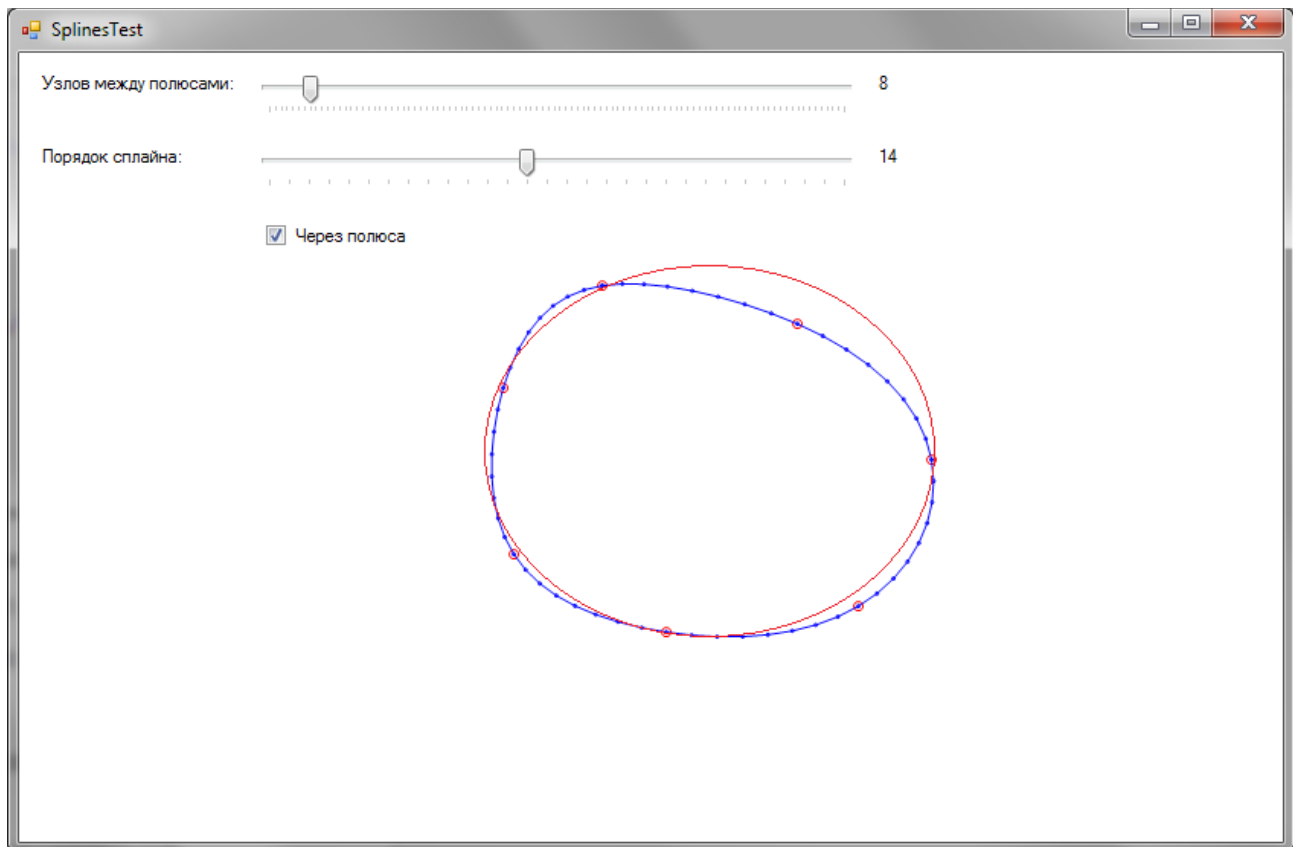


рисунок 20 - увеличение порядка сплайна.

Исходя из рисунка 20 следует, что увеличение порядка сплайна не дает улучшения, а даже наоборот портит конечную замкнутую кривую. Она становится менее округлой, что в свою очередь не позволит описать срез слитка достаточно точно.

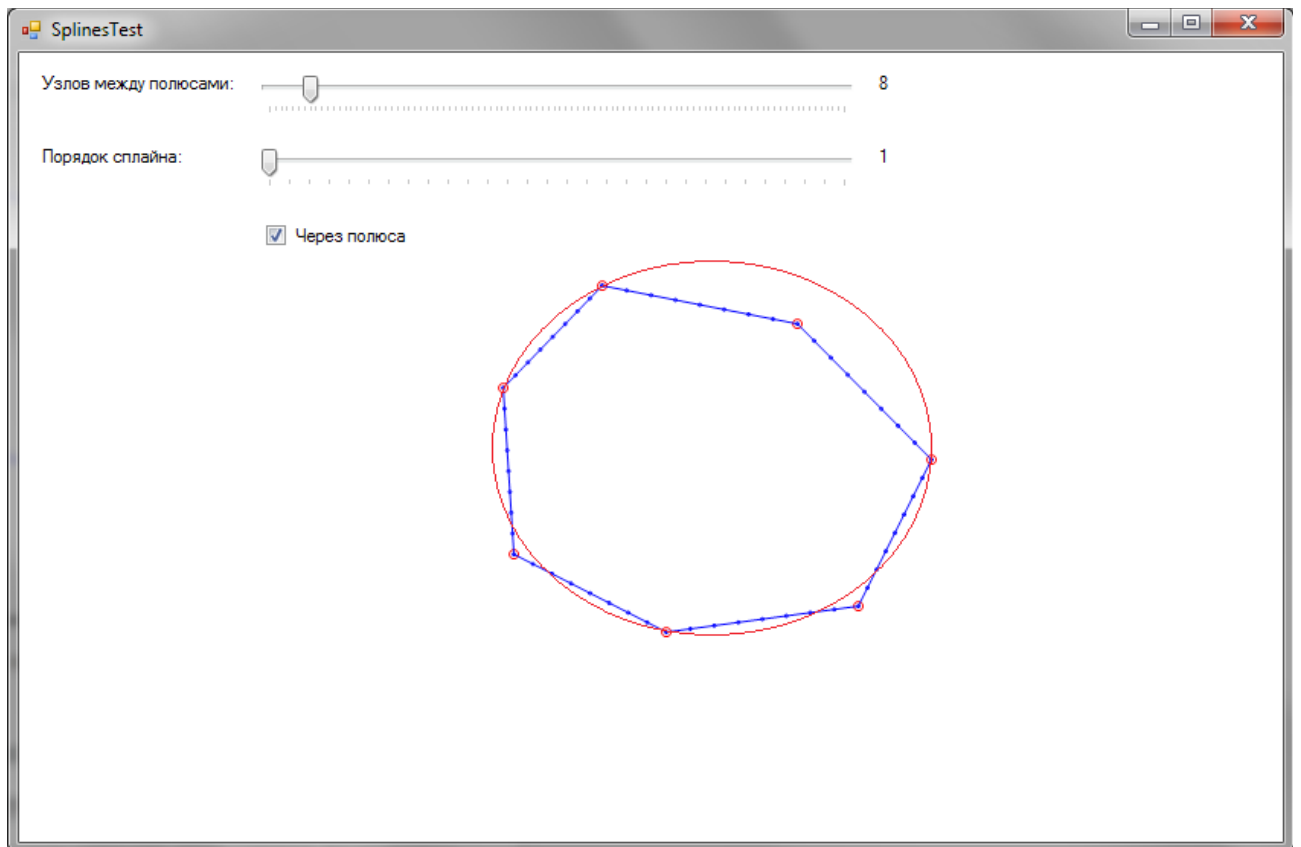


рисунок 21 - уменьшение порядка сплайна.

Уменьшение порядка сплайна приводит к наглядному несоответствию округлой форме.

В итоге экспериментальным путем выявлено, что наилучшее значение порядка сплайна для данной задачи является 2.

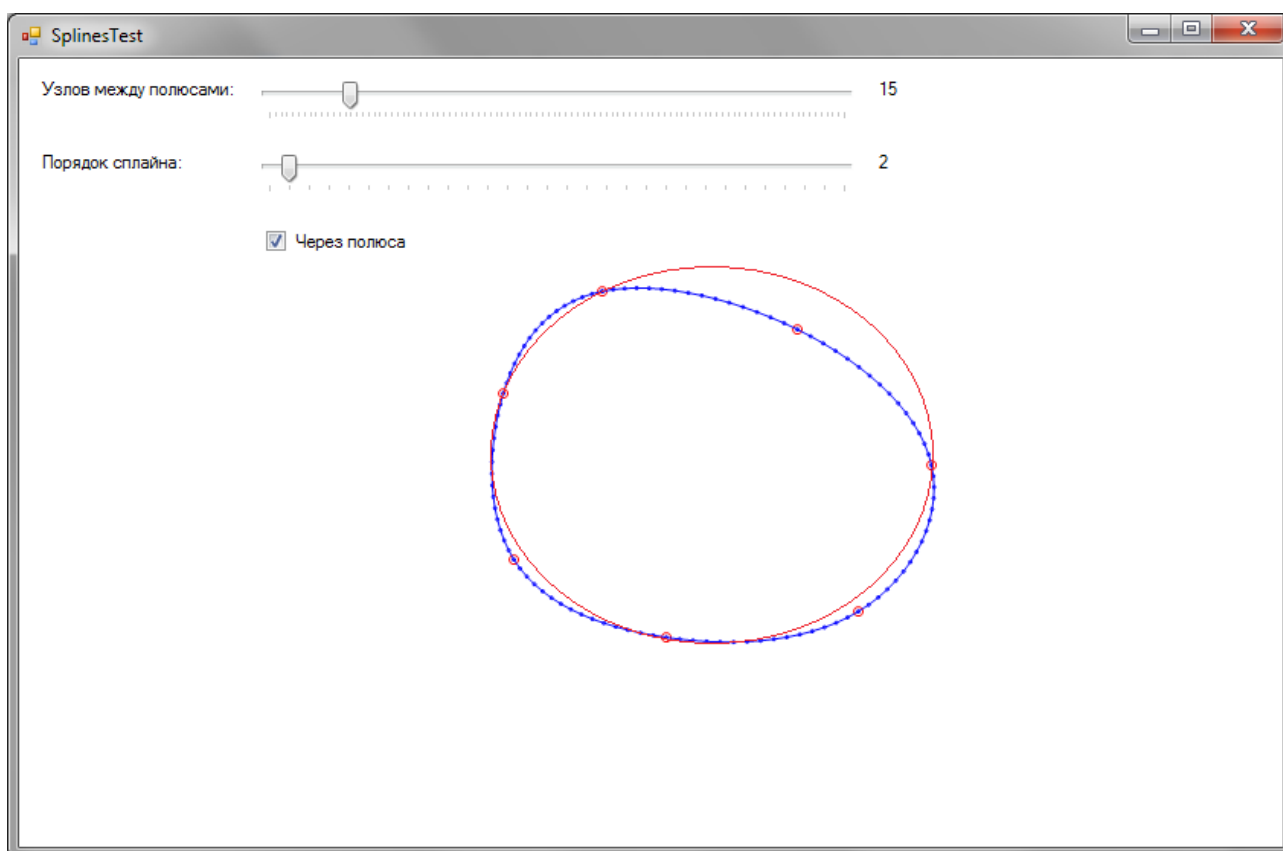


рисунок 22 - увеличение числа узлов.

Увеличение числа узлов, также не несет существенных изменений(см. рисунок 22).

3.4 Сравнение сплайнов Катмалла-Рома и Дискретных периодических сплайнов с векторным коэффициентом.

Для сравнения работы алгоритмов, рассмотрим их результаты.

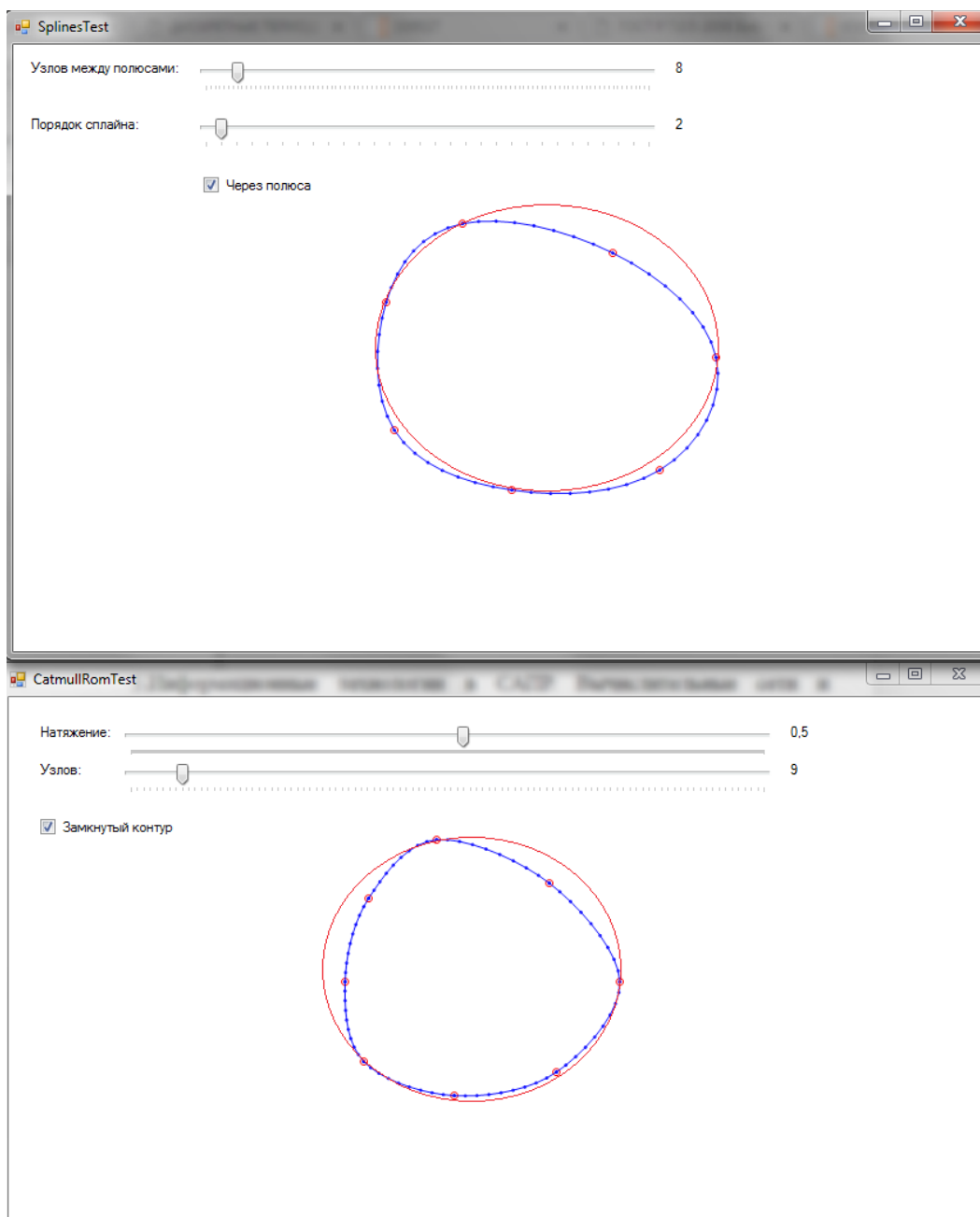


рисунок 23 - сравнение работы алгоритмов.

На рисунке 21 видно, что сплайны Катмалла-Рома(Нижняя часть рисунка 23) дают менее округлую форму замкнутой кривой в сравнении с Дискретными периодическими сплайнами(верхняя часть рисунка 23).Замкнутая кривая построенная с помощью сплайнов Катмалла-Рома имеет небольшие угловатости, что в свою очередь не позволяет описать окружность слитка достаточно точно.

Из вышеперечисленного следует, что наиболее удачным являются Дискретные периодические сплайны с векторным коэффициентом.

ЗАКЛЮЧЕНИЕ

В результате исследования, были рассмотрены следующие алгоритмы для интерполяции замкнутых кривых:

- периодические кривые Безье
- сплайны Катмалла-Рома
- Дискретные периодические сплайны с векторным коэффициентом

В ходе исследования вышеприведенных алгоритмов, экспериментальным путем было выявлено, что наиболее подходящими для решения задачи являются Дискретные периодические сплайны с векторным коэффициентом. Они и были использованы в модуле преобразования геометрических данных.

Разработанный программно-аппаратный комплекс позволил увеличить точность, в сравнении с предыдущим способом измерения, точность выросла в 1,5 раза.

Также были выявлены некоторые недочеты:

-при существенных отклонениях слитка от траектории движения Линейный и Модифицированный медианный фильтры не справляются с задачей.

-количество лазерных датчиков на рамке недостаточно для построения максимально точной геометрической модели слитка

В дальнейшем планируется исправление недочетов, поиск алгоритмов подавления шумов в данных удовлетворяющих условиям и ограничениям производства, доработка аппаратной части в плане увеличения количества точечных лазерных датчиков, а также добавлением на измерительную рамку 2D лазерных датчиков, для снятия поверхности слитка.

Также для объединения данных с точечных и 2D лазерных датчиков, будет проводится работа по проектированию алгоритма наложения данных, что в свою очередь позволит получить максимально полную геометрическую модель слитка.

СПИСОК СОКРАЩЕНИЙ

- фильтр с конечной импульсной характеристикой(КИХ фильтр)
- контрольно-измерительные приборы и автоматика (КИПиА)
- запасные изделия прилагаемые (ЗИП)
- автоматизированное рабочее место (АРМ)
- база данных (БД)
- программируемый логический контроллер (ПЛК)
- программное обеспечение (ПО)
- система управления базой данных (СУБД)
- application programming interface (API)
- OLE for process control (OPC)
- simple object access protocol (SOAP)
- Windows communication foundation (WCF)

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Лекции ФИСТ УлГТУ “Компьютерная Графика” URL:
<https://fex.gongled.me/>

2. Роджерс Д., Адамс Дж. Математические основы машинной графики. — М.: Мир, 2001.

3. Информационные технологии в САПР. Вычислительные сети и компьютерная графика : учебное пособие / С.А. Васильев, В.Е. Подольский, И.В. Милованов, В.И. Лоскутов. – Тамбов : Изд-во Тамб. гос. техн. ун-та, 2008. – 80 с.

4. Сплайны Катмалла-Рома/ Кристофер Твигг URL:
<https://www.cs.cmu.edu/~462/projects/assn2/assn2/catmullRom.pdf>

5. Catmull, E., and Rom, R. A class of local interpolating splines. In Computer Aided Geometric Design, R. E. Barnhill and R. F. Reisenfeld, Eds. Academic Press, New York, 1974, pp. 317–326.

6. Watt, A., and Watt, M. Advanced Animation and Rendering Techniques: Theory and Practice. ACM Press, 1992.

7. Дискретные периодические сплайны с векторными коэффициентами и поверхности кунса / В. Н. Малозёмов, Д. А. Хорохонов, Н. В. Чашников URL:
<http://www.math.spbu.ru/ru/mmeh/AspDok/pub/2010/Chashnikov.pdf>

8. Дискретные периодические сплайны с векторными коэффициентами и геометрическое моделирование / Н. В. Чашников URL:
<http://dha.spb.ru/PDF/discreteSplines.pdf>

9. Дискретные периодические сплайны и их вычислительные применения /Малозёмов В. Н., Певный А. Б. / Журн. вычисл. мат. и матем. физ. 1998. Т. 38. № 8. С. 1235–1246.

10. Основы дискретного гармонического анализа. Часть третья. / Малозёмов В. Н., Машарский С. М. / СПб.: НИИММ, 2003. 88 с.

11. Professional WCF 4: Windows Communication Foundation with .NET 4 / Pablo Cibraro, Kurt Claeys, Fabio Cozzolino, Johan Grabner