**Описание программного обеспечения**

1. Используемые технологии.
   1. Технология OLE for Process Control (OPC).
   2. Технология Windows Communication Foundation (WCF).
2. Схема взаимодействия компонентов программного и аппаратного обеспечения.
3. Модуль сервера.
   1. Настройки серверной части.
   2. Структура серверной части.
   3. Краткое описание алгоритмов обработки данных.
   4. Хранение данных.
4. Программное обеспечение оператора.
   1. Описание ПО оператора.
   2. Отображаемая информация.
   3. Настройка системы через ПО оператора.
   4. Анализ результатов сканирования.

**1 Используемые технологии.**

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

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 MsSQL.

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) .

**1.1 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».

**1.2 Технология Windows Communication Foundation (WCF).**

Windows Communication Foundation (WCF) - программный фреймворк, используемый для обмена данными между распределенными приложениями, входящий в состав Microsoft [.NET Framework](https://ru.wikipedia.org/wiki/.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 упрощённую 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](https://ru.wikipedia.org/w/index.php?title=.NET_Enterprise_Services&action=edit&redlink=1) и [System.Messaging](https://ru.wikipedia.org/w/index.php?title=System.Messaging&action=edit&redlink=1" \o "System.Messaging (страница отсутствует))), WCF предоставляет единую инфраструктуру разработки, повышающую производительность и снижающую затраты на создание безопасных, надёжных 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.

**2 Схема взаимодействия компонентов программного и аппаратного обеспечения.**

Аппаратный комплекс состоит из следующих частей:

1. Датчики расстояния, закрепленные на рамке.
2. Контроллер.
3. Сервер.
4. Подключаемые клиенты.

Схема взаимодействия компонентов аппаратного комплекса изображена на рисунке 1.

Датчики расстояния

Контроллер

Сервер

Клиент

Рисунок 1 – Схема взаимодействия аппаратного комплекса

Программное обеспечение серверной и клиентской части комплекса имеет схему взаимодействия, изображенную на рисунке 2.

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.

Сервер.

БД

Модуль связи с контрол-лером.

Рассчетный модуль.

Модуль связи с клиентам.

WCF контракт.

Клиент.

WCF контракт.

Модуль визуализации.

Модуль калибровки.

Модуль настроек.

…

Рисунок 2 – Схема взаимодействия серверного и клиентского программного обеспечения

**3 Модуль сервера.**

**3.1 Настройки серверной части.**

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

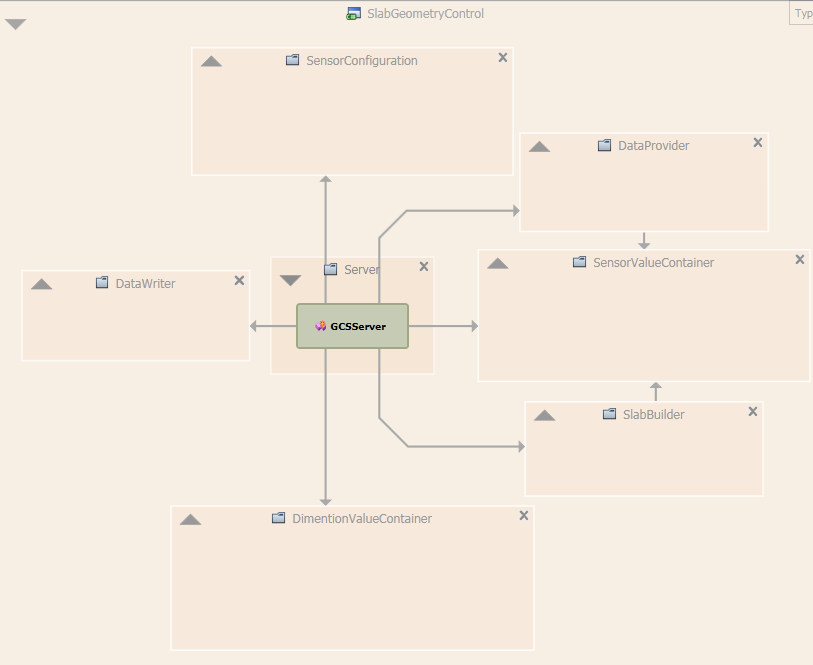
Перед запуском сервера выполняется конфигурация системы для ее корректной работы. Настройки записаны в следующих файлах формата xml.

1. 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.
2. OPCConfiguration.xml – настройки для подключения к OPC-серверу. Здесь задаются имена OPC-тегов для каждого датчика, а также для управляющего блока.
3. CalibratorConfiguration.xml – настройки модуля калибровки для всех датчиков. Калибровка датчиков выполняется переда началом работы сканирующей системы.
4. Hibernate.cfg.xml – настройки для работы с базой данных.
5. Logging.xml – настройки записей лог-файлов во время работы сервера.

**3.2 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 обеспечение, простота и скорость внесения изменений и дополнений.

Упрощенная диаграмма серверной части изображена на рисунке 3.

Рисунок 3 – Упрощенная диаграмма ПО сервера

**3.3 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:

1. **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 отсчётов внутри окна фильтра сортируются в порядке возрастания (убывания); и значение, находящееся в середине упорядоченного списка, поступает на выход фильтра. В случае четного числа отсчетов в окне выходное значение фильтра равно среднему значению двух отсчетов в середине упорядоченного списка. Окно перемещается вдоль фильтруемого сигнала и вычисления повторяются.
2. **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 4 изображены показания верхнего и нижнего датчиков, где видно место незначительного подпрыгивания слитка при наезде на рольганг.

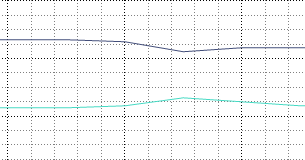


Рисунок 4 – Иллюстрация незначительного подпрыгивания слитка во время движения.

**3.4 Хранение данных.**

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

Microsoft SQL Server - система управления реляционными базами данных (СУРБД), разработанная корпорацией Microsoft. Основной используемый язык 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.

**4 Программное обеспечение оператора.**

**4.1 Описание ПО оператора.**

ПО оператора реализовано в виде оконного Windows-приложения. Взаимодействие с серверной частью происходит через TCP-соединение по протоколу SOAP, реализованном в рамках технологии WCF.

Запущенных экземпляров ПО оператора может быть сколько угодно. За счет этого возможна работа нескольких человек с системой. Кроме того, возможно разделение прав доступа, которое позволит распределять роли при работе и мониторинге состояния программно-аппаратного комплекса.

* 1. **Отображаемая информация.**

Главное окно ПО оператора имеет несколько информационных панелей, где отображается важна информация (см. рисунок 5).

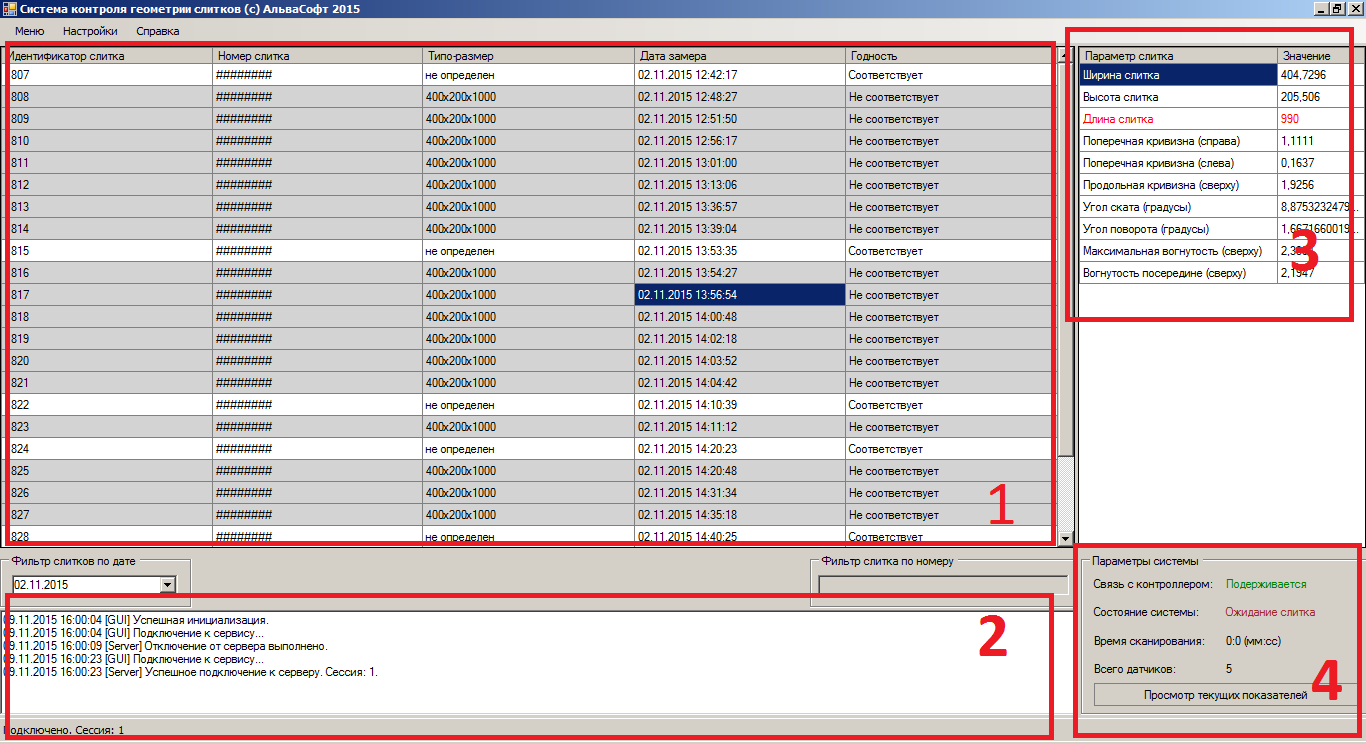


Рисунок 5 – Главное окно ПО оператора.

* Окно 1 – содержит информация о ранее отсканированных слитках: идентификатор слитка, номер слитка, типа-размер (вычисляется автоматически), дата сканирования, годность. В процессе работы системы новые слитки добавляются в конец списка. Выбор даты для отображения списка слитков осуществляется в фильтре слева снизу.
* Окно 2 – Содержит системные сообщения системы, где отображается информация о том, что происходило за время, пока было запущено ПО.
* Окно 3 – информация о конкретном слитке. Содержит значения характеристик слитка, которые были рассчитаны специальными алгоритмами. Параметры слитка устанавливаются согласно регламенту измерения, принятому на заводе. К ним относятся: длина, ширина, высота, поперечные и продольные изгибы и т.д.
* Окно 4 – информация о текущем состоянии системы. Включает в себя стадию процесса сканирования, состояние подключения контроллера, время с начала сканирования.

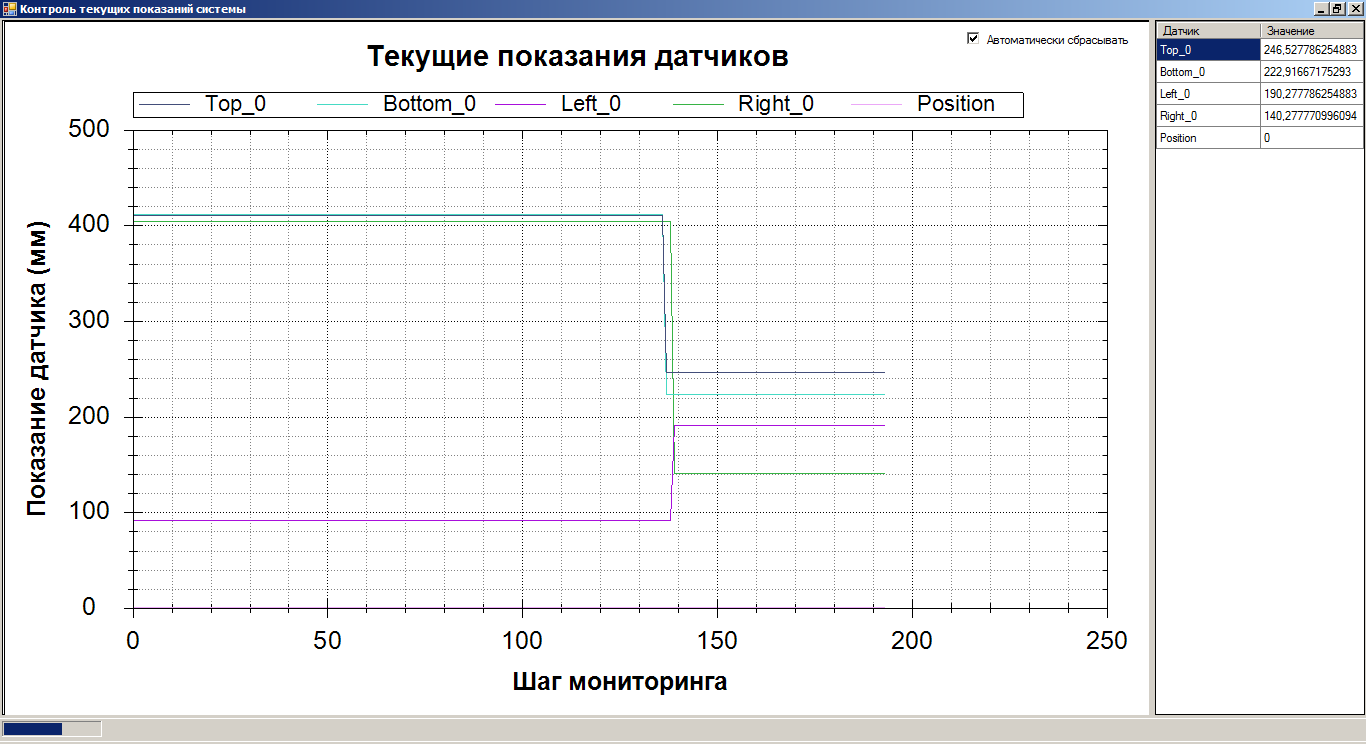


Рисунок 6 – Текущие показания датчиков.

В любое время работы системы доступна для просмотра информация о текущих показаниях датчиков. Кнопка «Просмотр текущих показаний» на главной форме открывает окно, в котором отображаются числовые значения, соответствующие показаниям датчиков в данный момент времени, а также в реальном времени строится график по этим данным (рисунок 6).

**4.3 Настройка системы через ПО оператора.**

Программное обеспечение оператора предоставляет возможность вносить изменения в конфигурацию всего программно-аппаратного комплекса. Для этих целей предусмотрено специальное меню «Настройки», где доступны следующие пункты: «Типо-размеры», «Регламент измерений», «Датчики», «Подключение», «Калибровка».

**Редактирование типо-размеров.**

Типо-размер – это эталонный размер слитка, который определяется его длиной, шириной и высотой. Для каждого типо-размера существуют свои нормы отклонений параметров слитка, записанных в регламенте измерений завода.

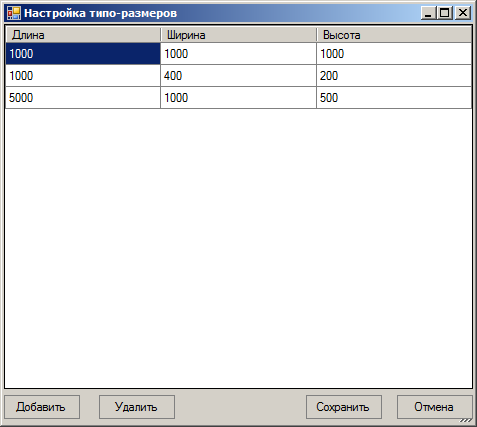


Рисунок 7 – Окно редактирования типо-размеров.

После окончания сканирования типо-размер для слитка выбирается автоматически из списка, заданного в конфигурации (рисунок 7). Важно указать все возможные типо-размеры слитков, чтобы определение годности выполнялось правильно.

**Изменение регламента измерений.**

Регламент измерений задает границы отклонений параметров слитка, при превышении которых считается, что слиток не соответствует достаточной степени годности. Такие границы могут изменяться при изменении стандартов, поэтому существует возможность их редактировать (рисунок 8).

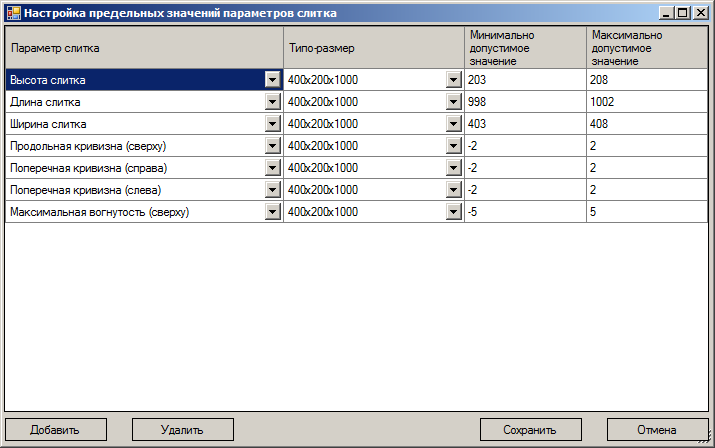


Рисунок 8 – Настройка регламента измерений.

**Изменение настроек датчиков.**

Датчики настраиваются с помощью добавления, удаления или редактирования на форме настройки датчиков (рисунок 9). Для каждого датчика задается его положение на рамке (сторона и смещение от центра), а также тип и имя.

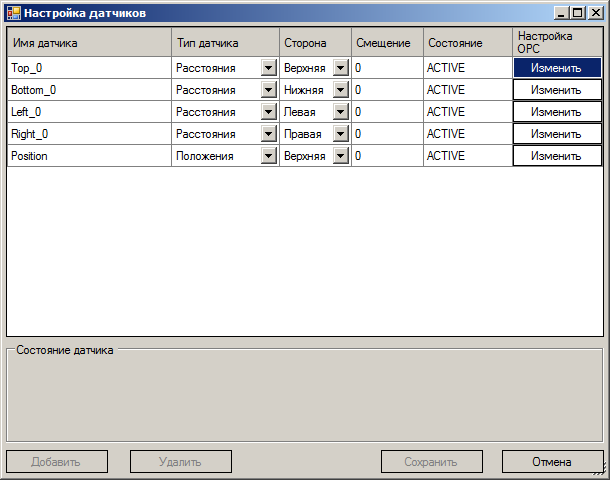


Рисунок 9 – Настройка датчиков.

Для изменения OPC-тегов (указателей на область памяти), необходимых для работы с контроллером, используется кнопка «Изменить» напротив каждого датчика.

Таким образом, редактирование настроек датчиков может выполняться не на стороне ПО сервера, а на удаленной машине, где работает ПО оператора.

**Настройка подключения к серверу.**

Для того, чтобы ПО оператора выполняло подключение к серверу, необходимо внести изменения в конфигурацию соединения (рисунок 10). Для этого задается сетевой адрес, по которому доступен сервер и порт, который им прослушивается.

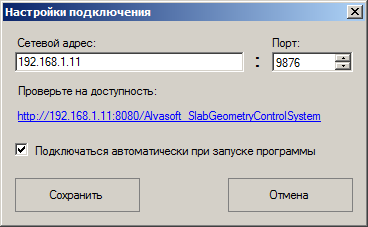


Рисунок 10 – Настройки подключения.

Проверить на корректность указанные настройки можно, кликнув по специально сформированной ссылке. В случае, если браузер успешно загрузил xml-схему WCF-контракта, это означает, что соединение будет успешно установлено.

**Калибровка датчиков.**

Важным этапом настройки всего программно-аппаратного комплекса является калибровка датчиков (рисунок 11). От качества заданных калибровочных значений зависит точность будущих измерений.

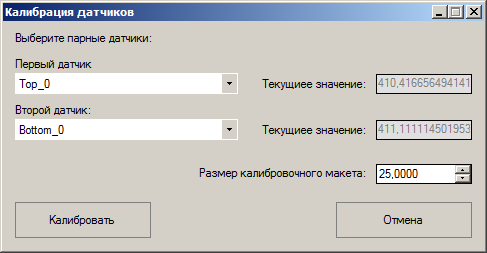


Рисунок 11 – Калибровка датчиков.

Процесс калибровки заключается в том, что задаются расстояния между парами датчиков таким образом, чтобы вычисления размеров калибровочного макета происходили максимально точно. При этом характеристики калибровочного макета заранее известны и задаются вручную.

Калибровка происходит автоматически при нажатии кнопки «Калибровать». При этом макет с заданными размерами должен быть помещен между парными датчиками, выбранными для калибровки.

**4.4 Анализ результатов сканирования.**

Результаты сканирования для каждого слитка доступны при просмотре графиков показаний датчиков, срезов слитка, графиков отклонения от среднего, а также визуальной 3-D модели слитка.

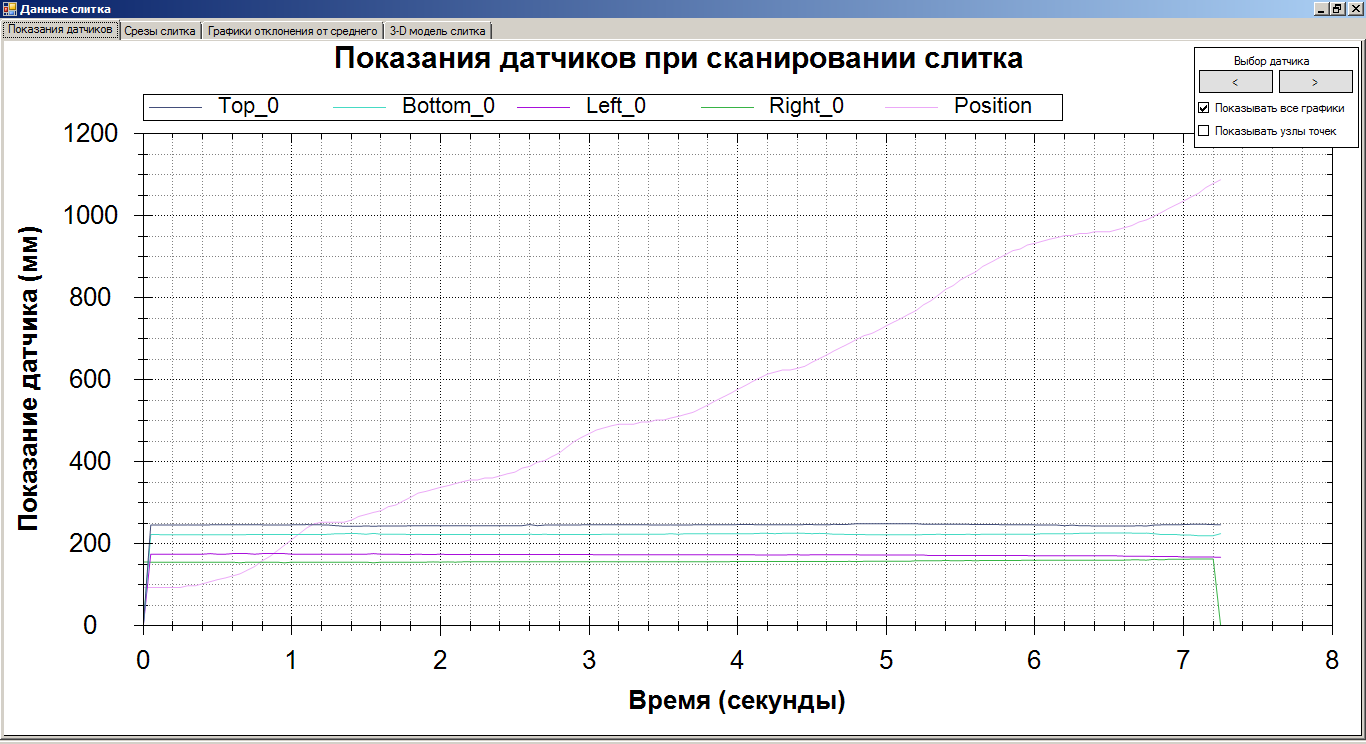


Рисунок 12 – Графики показаний датчиков.

Просмотр графиков показаний датчиков возможен как для всех датчиков одновременно (рисунок 12), так и для каждого датчика отдельно (рисунок 13). При этом существует возможность включать и отключать отображение конкретных точек, увеличивать или отдалять выбранные места графика.



Рисунок 13 – График показаний верхнего датчика.

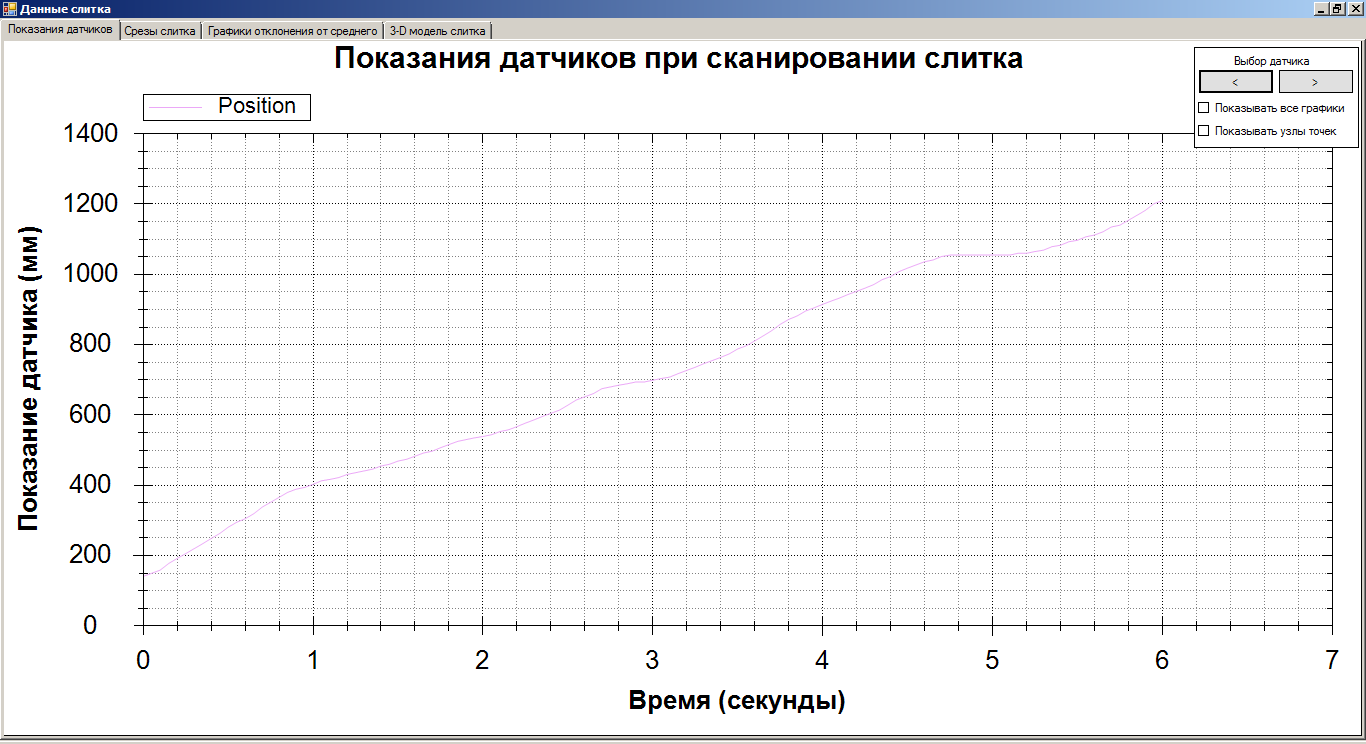


Рисунок 14 – График показаний датчика положения.

Срезы слитка сверху и слева показаны на вкладке «Срезы слитка» (рисунок 15). Масштабируя полученную картинку, можно визуально отследить отклонения сторон слитка от идеального состояния.

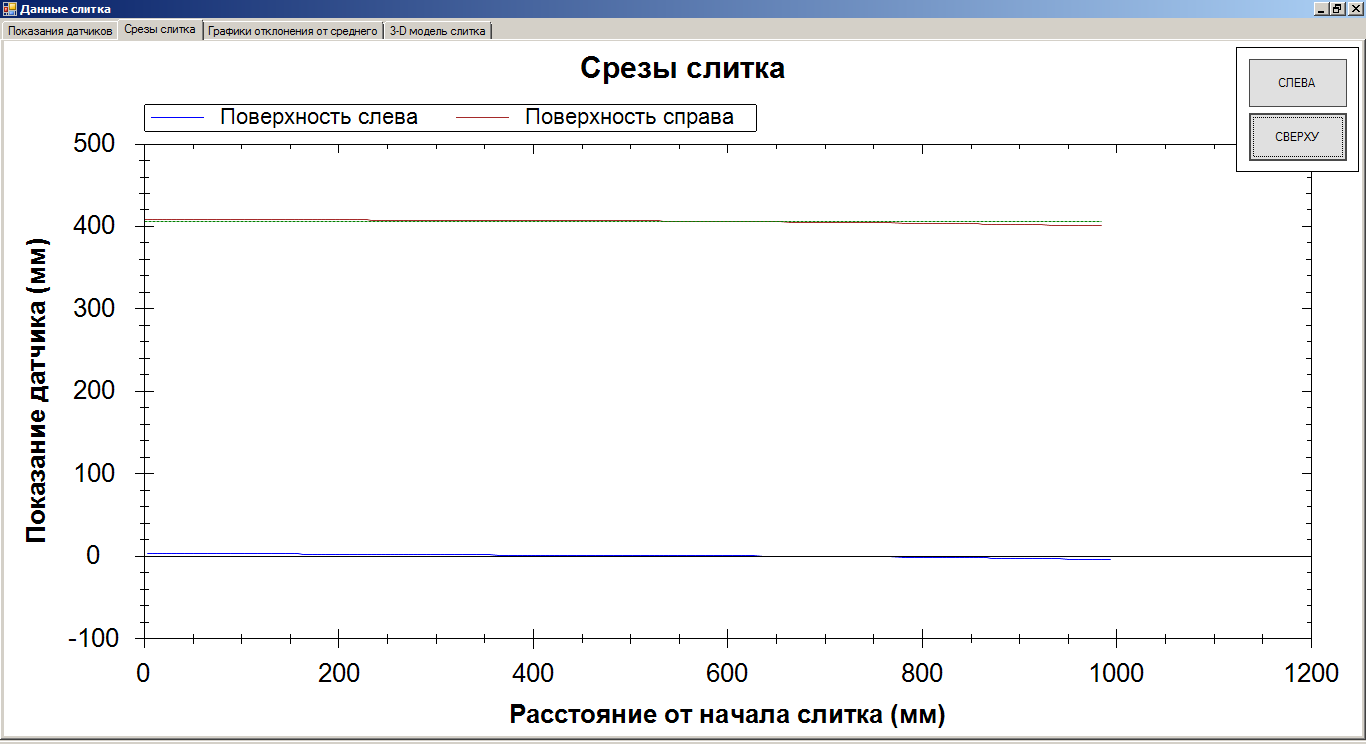


Рисунок 15 – Срез слитка сверху.

На вкладке «Графики отклонения от среднего» для каждой стороны слитка отображается ее поверхность и виртуальная натянутая леска, которая позволяет проследить отклонения и неровности (рисунок 16).

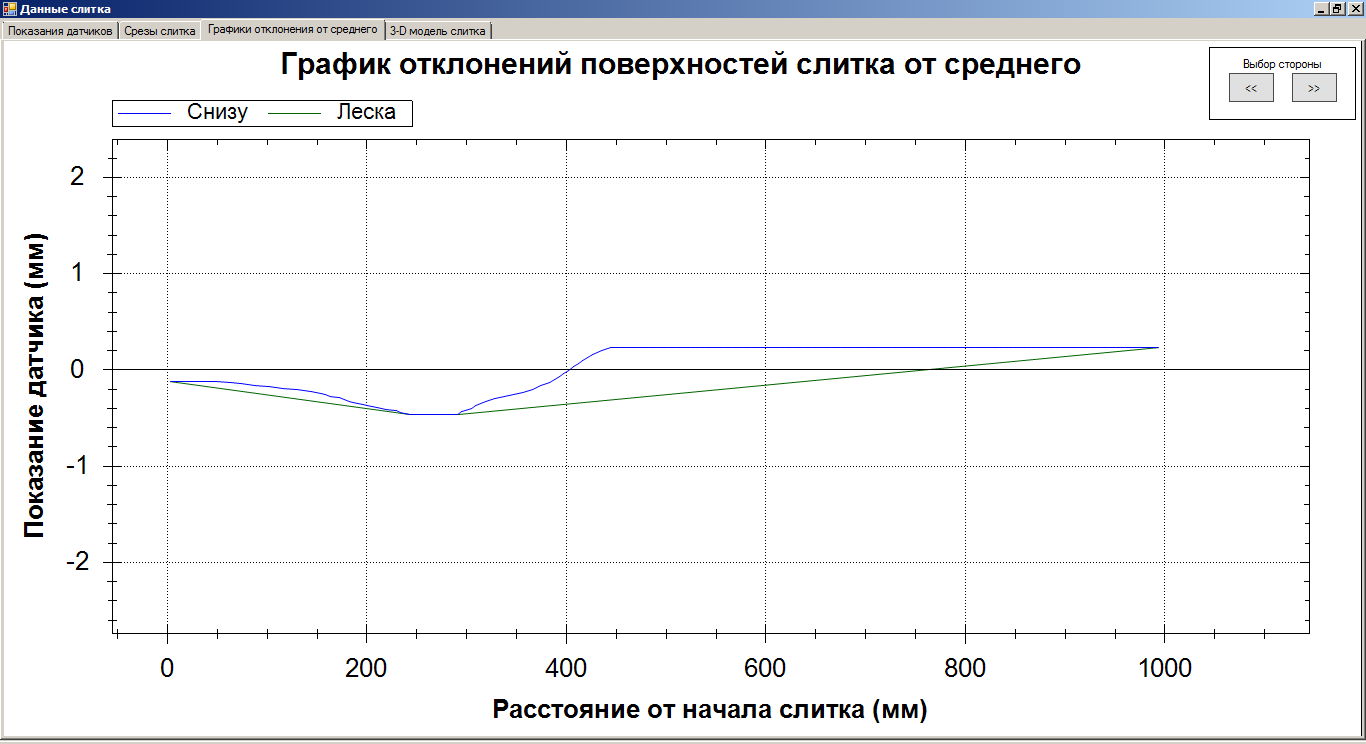


Рисунок 16 – Поверхность слитка снизу.

Изменяя масштаб по оси ординат, можно сделать выводы о качестве поверхности анализируемого слитка (рисунок 17).

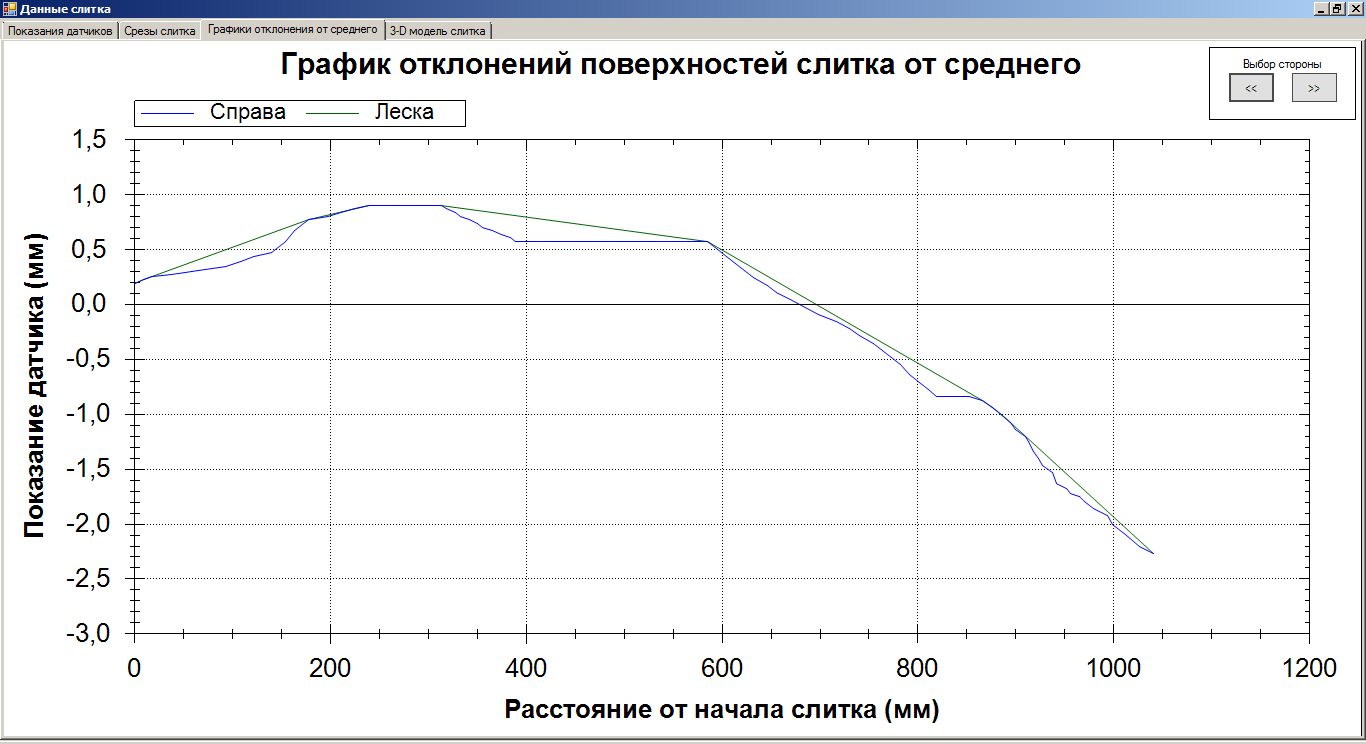


Рисунок 17 – Поверхность слитка справа.

Выбор стороны для отображения осуществляется путем переключения кнопками «<<» и «>>».