МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**Факультет прикладной математики, информатики и механики**

Кафедра механики и компьютерного моделирования

РЕШЕНИЕ ЗАДАЧИ КИНЕМАТИКИ ДВИЖЕНИЯ РАБОЧЕГО ИНСТРУМЕНТА ЛАЗЕРНОГО РЕЗАКА С ПОМОЩЬЮ ГЛАДКИХ КРИВЫХ

**БАКАЛАВРСКАЯ РАБОТА**

Направление 01.03.03 Механика и математическое моделирование

Профиль Компьютерный инжиниринг в механике сплошных сред

|  |  |  |  |
| --- | --- | --- | --- |
| Зав. кафедрой | \_\_\_\_\_\_\_\_\_\_ | д.ф. -м.н. | Ковалёв А.В. |
| Обучающийся | \_\_\_\_\_\_\_\_\_\_ | Боев А.Ю. |  |
| Руководитель | \_\_\_\_\_\_\_\_\_\_ | к.ф.-м.н., доц. | Яковлев А.Ю. |

Воронеж 2021

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| СОДЕРЖАНИЕ   |  |  | | --- | --- | | Введение | 3 | | Постановка задачи | 5 | | Актуальность | 6 | | ЧАСТЬ 1. Методы перемещения по дуге | 7 | | 1.1 Способ многоугольников в G-code | 7 | | 1.2 Интерполяционные методы | 8 | | 1.3 Не интерполяционные методы | 12 | | ЧАСТЬ 2. Экспериментальная часть | 15 | | 2.1 Инструментарий | 15 | | 2.2 ПО и выходные данные на станок | 16 | | 2.2 Разные методы перемещения по дуге на практике | 19 | | ЧАСТЬ 3. Заключение и сравнительный анализ | 23 | | Список литературы | 24 | |

**Введение.**

Если обработку по прямой линии несложно производить и на простом станке с ручным управлением, то перемещение инструмента по дуге точнее и проще выполнять на станке с ЧПУ. Большинство двухкоординатных станков работает с чертежами созданными в программах по типу AutoCAD или SolidWorks и эти программы в свою очередь создают файлы в формате .dwg, .svg, .dxf и им подобных. И проблематика заключается в том, что в отличии от обычных графических изображений, с которых можно скопировать и повторить дугу, тип графики этих чертежей является векторным и все элементы, в том числе и дуги, заданы двумя или иногда больше, координатами.

Например вот так в формате .svg выглядит желтая звезда, если бы мы хотели отобразить её на web странице или специальном ПО:  
  
<?xml version="1.0" encoding="UTF-8" standalone="no"?>  
**<svg** width="198" height="188"

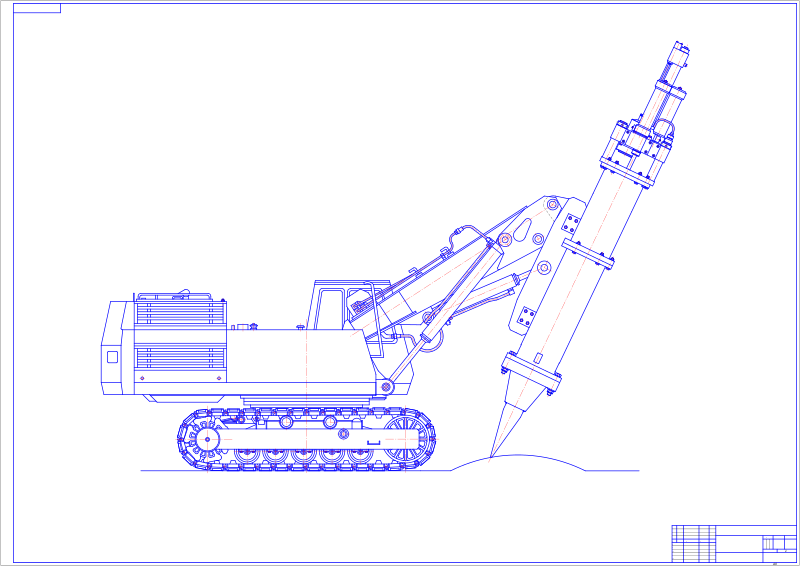
viewBox="0 0 198 188"

version="1.1"

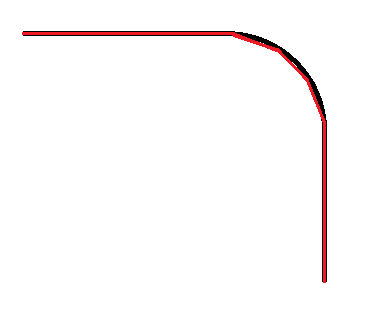
baseProfile="full"

xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ev="http://www.w3.org/2001/xml-events"**>**   
**<g** id="Page-1" stroke="none" stroke-width="1" fill="none" fill-rule="even-odd"**>**   
 **<polygon** id="Star-1" stroke="#979797" stroke-width="3" fill="#F8E81C" points="99 154 40 185 51 119 4 73 69 64 99 3 128 64 194 73 147 119 158 185 "**>** **</polygon>**  
 **</g>**  
**</svg>**

Такой способ представления чертежей сейчас используется во всех любительских и промышленных машинах и он полезен тем, что его можно масштабировать без потери качества, он занимает мало объема на жестких дисках и легко поддаётся редактированию, но это создаёт некоторые трудности при интерпретации чертежа в команды G-code для микроконтроллеров ЧПУ станков.

  
Рис 1. пример файла .svg

# **Постановка задачи.**

Дан 2-ух координатный станок с программным управлением. Требуется рассмотреть различные способы решения кинематической задачи, то есть создания траектории движения по дуге на плоскости. Необходимо провести сравнительный анализ и определить оптимальный способ движения инструмента по дуге и решения проблем связанных с неточным движением по скруглённым траекториям.  
  
рис 2. Пример того как срезается угол, красная линия - траектория движения инструмента.

**Актуальность.**  
 Сегодня станки пользуются универсальным машинным кодом G-code в котором есть только две команды для проецирования дуг с помощью линейной интерполяции, G-02 и G-03, для движения по часовой и против часовой стрелки, и эти команды редко используются в специализированных средах управления.  
  
Проблема среза округлых углов решается только либо высокой производительностью компьютеров или при добавлении дополнительного материала на модели, или создания отступа для дальнейшей обработки и так как ЧПУ станки используются повсеместно и в будущем будут только набирать популярность, то нахождение новых опций для перевода дуг из векторных файлов в команды для станков может быть полезным в такой быстроразвивающейся области.

**ЧАСТЬ 1. Методы перемещения по дуге**

**1.1 Способ многоугольников в G-code**

Один из часто используемых способов, который используется в домашних 3д-принтерах и многих других станках любительского уровня - это так называемый способ многоугольников. Работает он таким образом, что программное обеспечение установленное на персональном компьютере, например слайсер для 3д принтера, после визуализации дуги, разделит её на прямые отрезки как показано на рисунке ниже.

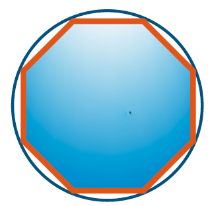


Рис 3. пример способа многоугольников

Не смотря на то что этот способ очень простой для понимания, он очень требователен к вычислительной мощности, и только на высокопроизводительных машинах получится добиться визуально гладкого результата

**1.2 Интерполяционные методы**

**Метод круговой интерполяции**

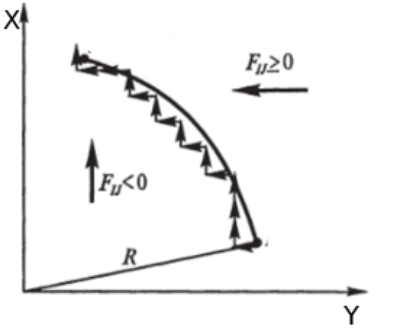
Как и любой другой метод интерполяции - это способ определения промежуточных значений по дискретному набору данных.  
  
При дуговой (круговой) интерполяции ломаной линией аппроксимируются участки дугового контура  
В нашем случае, так как мы рассматриваем векторные файлы чертежей, дуга будет задана начальными координатами, конечными координатами и координатами центра окружности, а также мы будем знать направление движения по этой дуге.   
Для примера мы будем рассматривать движение в 1 квадранте движения часовой стрелки, а центр дуги будет помещен в центр координат. Вся методика интерполяции легко обобщается на случай движения в других квадрантах часовой стрелки и против часовой стрелки.  
  
  


Рис 4. пример круговой интерполяции

Расстояние (радиус-вектор) от начала координат до текущей точки траектории с координатами xi и yj определяется соотношением:

Rij=(xi2+y2j)½  1.1

Оценочная функция определяется знаком разности квадратов текущей длины радиуса-вектора Rij фактической траектории и заданного значения радиуса дуги R, т. е.:

Fij=x2i+y2j-R2  1.2

Как видно из рисунка если промежуточная точка траектории интерполяции находится в области F > 0, то следующий шаг дается по оси X. Если же промежуточная точка траектории интерполяции находится в области F < 0, то следующий шаг дается по оси Y.  
  
При шаге по оси Y получаем уj+1 = уj - 1 а оценочная функция в таком случае равна:

Fi,j+1 =x2i + (yj-1)2 - R2 = Fij + (2zj+1) 1.3

При шаге по оси X получаем xi+1 = xi + 1, оценочная функция при этом будет:

Fi+1,j=(xi+1)2 +zj2- R2 = Fij + (2xi + 1) 1.4

Величины (—2y,+ 1) и (2х,+ 1) называются константами круговой интерполяции в точке с координатами xi и yj.

Аналогично можно получить расчетные соотношения для оценочной функции в других квадрантах и при других направлениях движения.   
Суть заключается в том, чтобы записать выражения для новых значений координат при шаге в соответствующем квадранте по тем или иным осям в заданном направлении и полученные новые значения координат подставить в выражение для оценочной функции при дуговой интерполяции.

Этот способ сильно сокращает объемы информации вводимые в устройство ЧПУ.

Сейчас существуют две команды, которые используют круговую интерполяцию в G-code, но на моей практике я не встречал случаев их использования. Скорее всего это обусловлено их громоздкостью и плохой реализацией.

Код G02 используется для перемещения по дуге по часовой стрелке, a G03 – против часовой стрелки. Направление перемещения определяется, когда мы смотрим на инструмент со стороны шпинделя, в отрицательном направлении оси Z.

Для использования этих команд надо знать и радиус и центр окружности, и предполагать какой сектор окружности занимает дуга, хотя всю эту информацию можно автоматически взять из файлов чертежа.

**Метод кубического сплайна и логарифмической интерполяции**  
  
В силу своего происхождения интерполяция сплайнами, хоть и даёт большой простор для реализации большого количества сложных чертежей кривых, но в большинстве случаев такие кривые лишь походят на оригиналы из чертежа, и имеют небольшие отличия, что делает метод кубических сплайнов не применимым в точных машинах, а также из-за большого количества вычислений с плавающей точкой (экспоненциальная форма представления вещественных чисел, в которой число хранится в виде мантиссы и порядка) является сложно или совсем нереализуемым на чипах разрядностью меньше 32-ух бит.  
В случае с логарифмической интерполяцией ситуация такая же, как и с интерполяцией сплайнами, помимо что для её выполнения не хватает начальных данных, и требуется создавать дополнительное ПО для создания функции, такой метод также сглаживает получившиеся данные, что неприменимо для точных машин.  
Полезность этих двух методов нивелируется методом круговой интерполяции.

**1.3 Неинтерполяционные методы**  
  
**Геометрический метод с помощью формулы угла сектора и радиуса.**  
  
Перед тем как использовать такой подход, требуется найти угол сектора, начальный и конечный, зная длину дуги и радиус мы можем это сделать:

180/fi=(Pi\*r)/L 2.1  
  
А затем:  
  
x = r\*cos(fi) ; 2.2   
y = r\*sin(fi); 2.3  
   
Зная три точки, начало дуги, конец дуги и центр окружности дуги из файла чертежа мы можем вычислить углы, а затем зная радиус, постепенно меняя угол построить координаты нужных нам точек для движения инструмента.  
  
Такой способ может быть достаточно точным и средне-зависимым от производительности, но ограничивается дугами, которые являются секторами окружности, а также редактировать такую траекторию получится только после прогона через формулу, из уже готовых данных.

**Графический метод**  
  
Графический метод заключается в конвертировании чертежа в графический файл, png, jpg, bmp. А затем на основе пикселей построить траекторию.

Обработка данных была выполнена в программе самостоятельно написанной в Visual Studio C# 2016 для растровой гравировки, и она прекрасно подходит для наших требований.

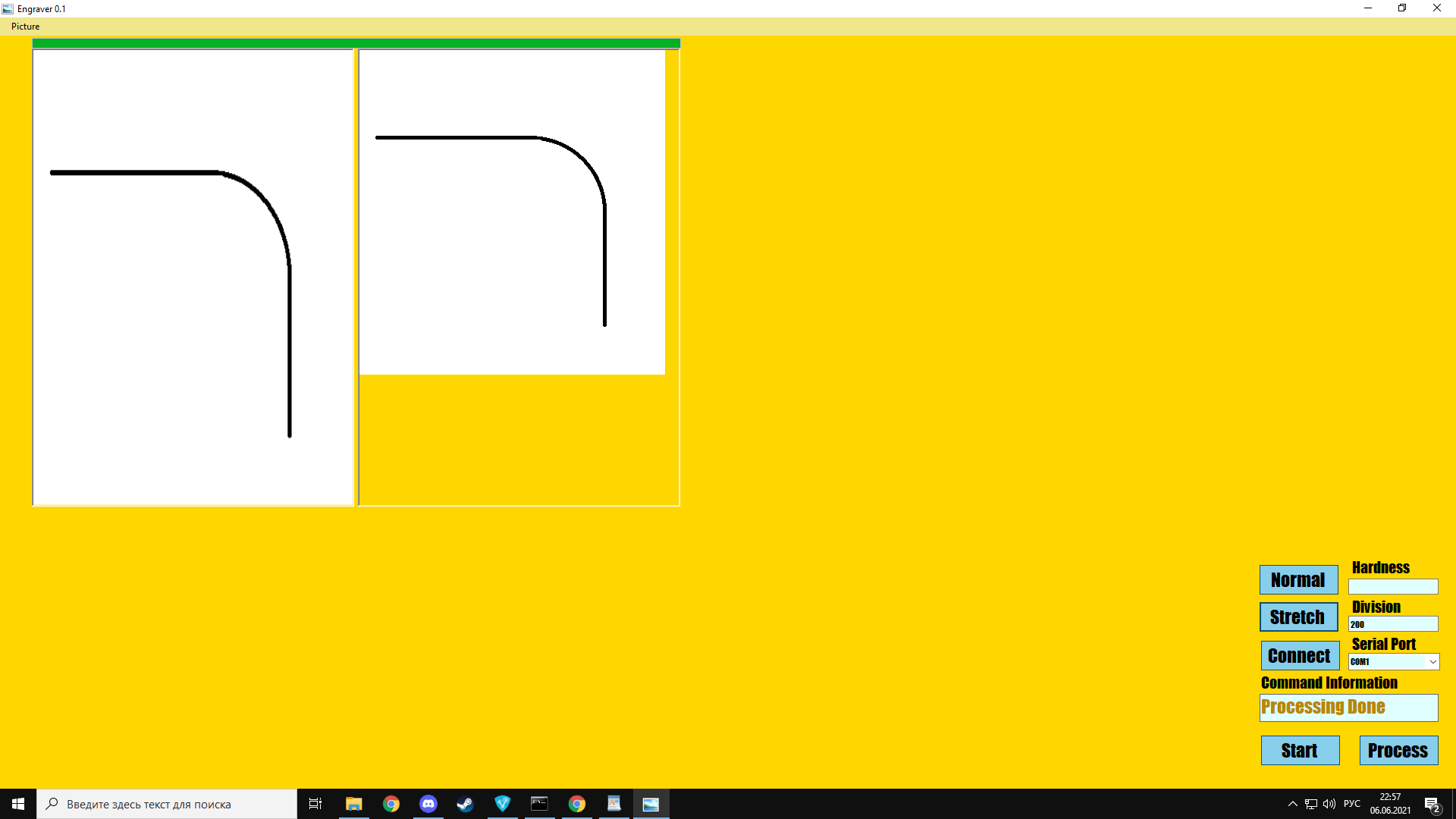


Рис 5. Оригинальная программа обработки изображений для лазерного гравера

Такой способ позволяет обрабатывать чертежи любой сложности, но как правило такие программы работают в растровом режиме, то есть каждую “строку” пикселей они интерпретирую в сигналы инструменту и проходят их сверху вниз, по каждой “строке” в независимости от сложности чертежа, из-за этого такой способ сильно проигрывает в скорости выполнения задачи и энергоэффективности.

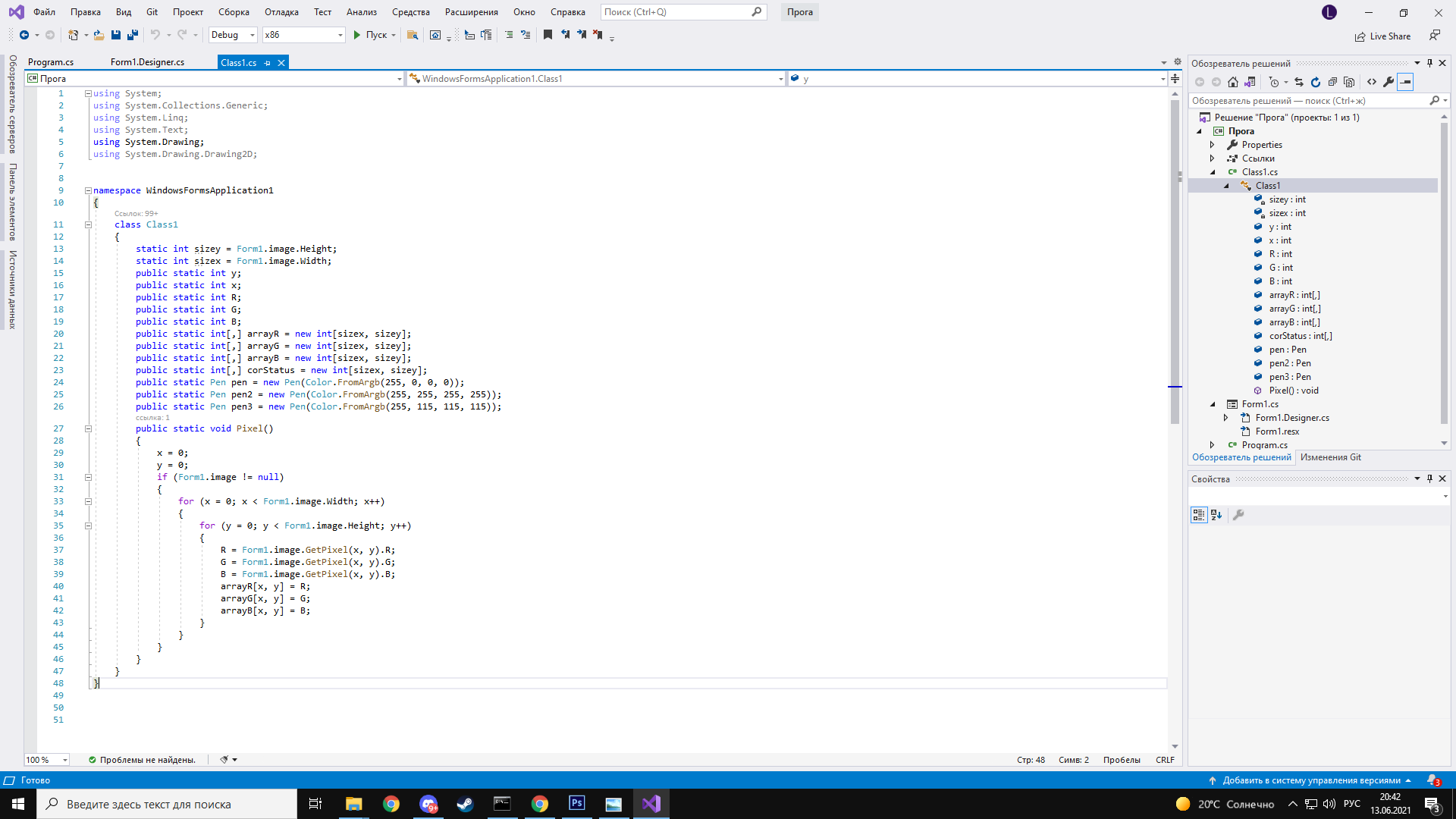


Рис 6. Вид основного класса и структуры приложения в Visual Studio C#

**ЧАСТЬ 2. Экспериментальная часть**  
  
 **2.1 Инструментарий**

В качестве станка с движущимся в двухкоординатной плоскости инструментом мы возьмём лазерный гравёр, с помощью него можно будет легко визуально оценить результаты успешности решения кинематической задачи движения инструмента по кривой.  
  
**Механическая составляющая проекта**

2- Шаговых двигателя 17HS3001

2- драйвера a4988

2- зубчатых ремня 1/16

2- зубчатых шестерни на 32 зуба

4- гладких вала 8мм

6- скользящих подшипников LM8UU

2- фланцевых миниатюрных подшипника

1- плата Arduino uno rev 3  
1- Лист фанеры  
2- концевых выключателя мц10  
1- кулер 2Вт  
1- лазер - твердотельный полупроводниковый диод 2000мвт   
1- блок питания 12в

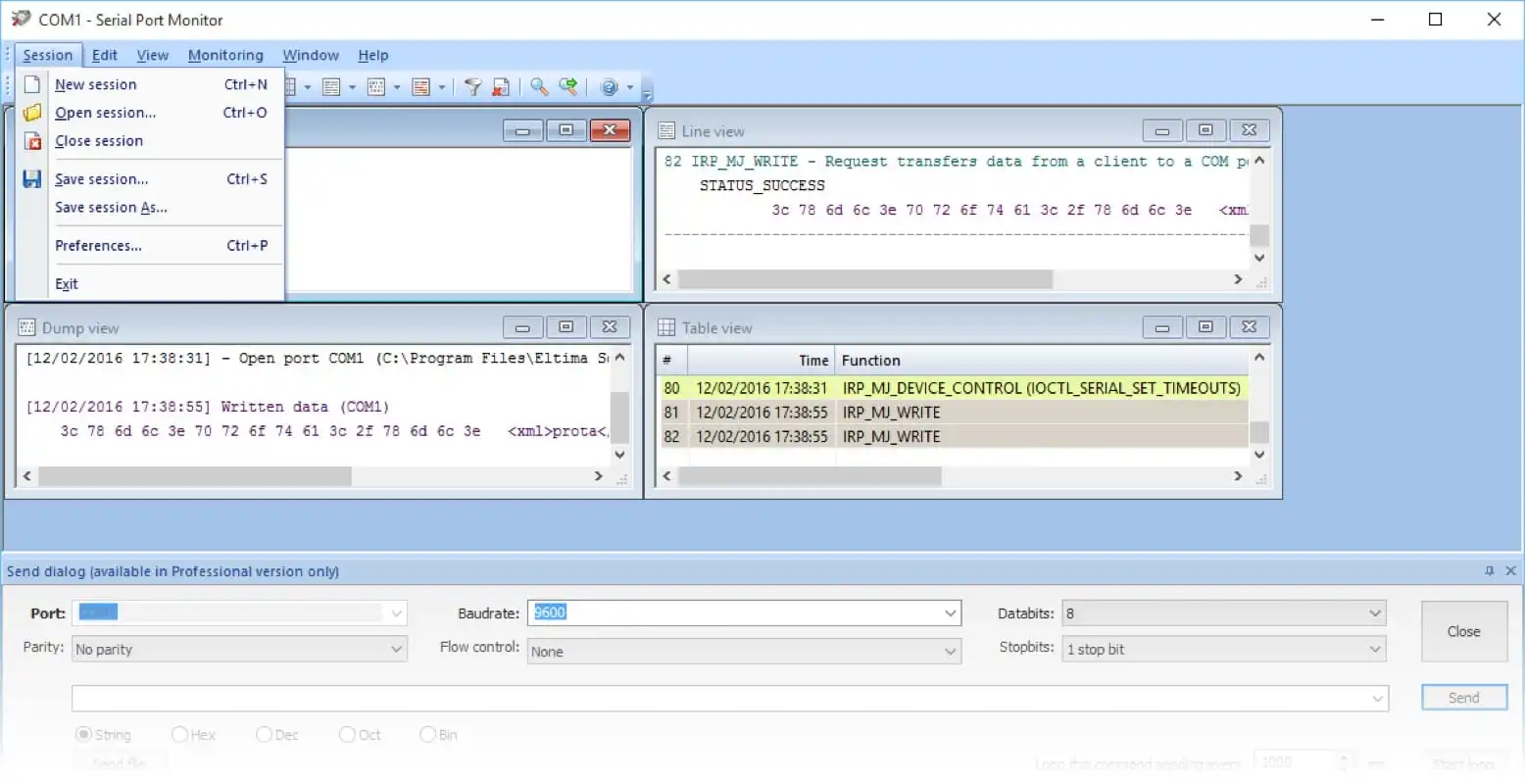
-Набор деталей сделанных на 3д принтере Все модели сделаны в Autodesk 123D Design.  
-Провода, конденсаторы разной емкости и резисторы



Рис 7. Внешний вид лазерного гравера, на котором будет проходить тест вышеописанных методов.

**2.2 ПО и выходные данные на станок**

Во время эксперимента была использована полпулярная прошивка laserGRBL, а также самостояельно написанная на C#, использующие G-code.  
 Помимо программы для графической обработки м прошивки мы также будем пользоваться Notepad++ для написания скриптов, с помощью которых будут создаваться таблицы координат, которые мы будем отправлять на плату с помощью программы Serial Port Monitor.

  
Рис 8. Рабочая вкладка программы Serial Port Monitor

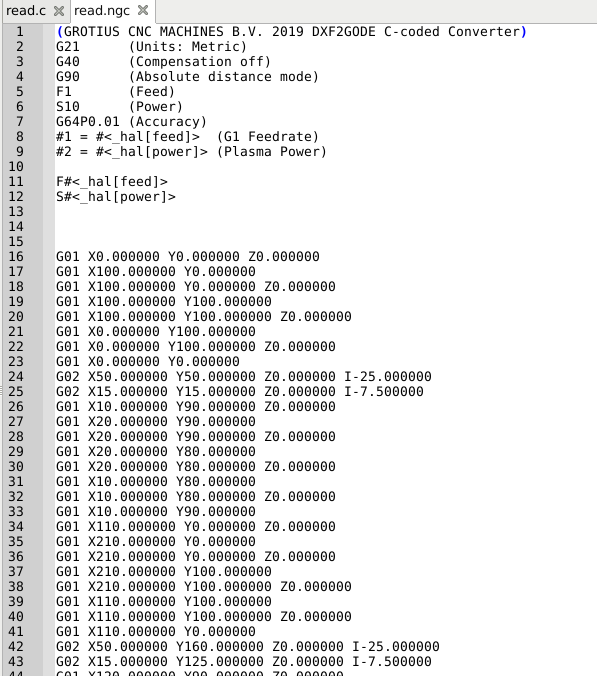


Рис 9. Пример сформированных данных для отправки на ЧПУ

Внедрение новых способов создания траектории гладких кривых в програмном обеспечении лазерного гравера, таком как **LaserCut** или **AutoLaser,** для некоторых подходов очень трудо и время затратно, поэтому для эксперимента мы будем частично заполнять и отправлять данные(командные строки G-code) вручную, с помощью текстовых файлов и программы **Serial Port Monitor**.  
  
Компьютер который мы будем использовать для создания строк команд и обработки файлов, имеет следующие интересующие нас параметры:  
  
Оперативная память 16гб  
Процессор Ryzen 3500x 4.4 ггц  
ОС Windows 10  
  
Видеокарта и другие составляющие компьютера влияют на скорость обработки данных незначительно.

**2.2 Разные методы перемещения по дуге на практике**  
***Каждый эксперимент был проведен при длине дуги равной 10 см.***

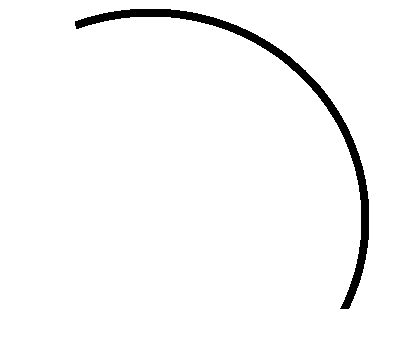
Ниже будет предоставлены результаты гравировки дуг на лазерном гравере на фанерном листе с помощью разных методов.  
  


Рис 10 и 11, пример исходной дуги и проведенной с помощью графического метода в оригинальной программе.

**Реализация метода многоугольников.**

Во время эксперимента всё прошло успешно, хоть и заняло много времени. Траектория была сделана с расчетом 28 сторон на 1 см.



Рис 12. Результат использования подхода многоугольников

**Реализация метода круговой интерполяции.**  
  
При реализации данного метода, всё прошло успешно и результат удовлетворительный.



Рис 13. Результат использования интерполяционного подхода

**Реализация геометрического метода.**  
  
Оказалось что при реализации геометрического метода, из-за частого повторения координат поблизости и по диагонали, гравёру не хватало шага в 1/80 мм.  


Рис 14. Результат использования геометрического подхода

**ЧАСТЬ 3. Заключение и сравнительный анализ**  
  
В ходе выполнения выпускной работы мы рассмотрели 3 разных подхода решения проблемы интерполяционный, не интерполяционный и метод многоугольников, затем рассмотрели механическую и программную часть инструментов и провели эксперимент. На основе получившихся данных мы можем составить таблицу результатов, нас интересует длительность создания конечного пакета команд, отправляемого на лазерный гравер, время выполнения этих команд, и визуальная оценка качества

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Время обработки | Время выполнения | Оценка качества |
| Метод многоугольников | 34 секунды | 40 секунд | Высокое |
| Круговая интерполяция | 6 секунд | 43 секунды | Высокое |
| Графический метод | 12 секунд | 4 минуты  27 секунд | Среднее |
| Геометрический метод | 4 секунды | 25 секунд | Низкое |

В результате сравнения результатов можно сделать вывод о каждом методе.  
Метод многоугольников хоть и потребовал большой мощности, выдал вполне хороший результат, хоть и заметны места, где были срезаны углы.  
Круговая интерполяция на мой взгляд оказалась лучшим решением, сочетая в себе высокое качество и быстроту обработки.  
Графический метод показал средние результаты, его применение имеет смысл только в том случае, если применить два предыдущих подхода не является возможным.  
Геометрический метод оказался самым быстрым при реализации и обработке, но на лазерном гравере, который у нас в распоряжении, он выполнялся с недочётами, этот метод имеет потенциал в машинах требующих большей скорости выполнения и с высокой точностью позиционирования.

Получившиеся данные при дальнейшем развитии могут иметь большой потенциал для новых инноваций в области точной механики или использоваться в учебном процессе при изучении проблем кинематики.

**Список литературы**

1. Модулятор Лазерного излучения Е.И. Тренкаль, А.А. Бамбизов, К.Ю. Осипов.
2. [Уральский Федеральный университет им. Б.Н. Ельцина «УПИ»](https://studfile.net/urfu/) Правильный учет величины и характера нагрузок и условий работы.
3. [Grbl](https://bengler.no/grbl) by [Simen Svale Skogsrud](https://bengler.no/simen).
4. <https://wiki.eltima.com> [Serial Port Monitor User Guide](https://wiki.eltima.com/user-guides/spm-user-guides.html), Eltima IBC.
5. Лазерный гравер: основы практического использования, Артем Корнишин, журнал LASERCUT.
6. Методы интерполяции в техническом диагностировании, Твердохлебов В.А. институт РАН. г Саратов.
7. Аверьянов О.И. Модульный принцип построения станков с ЧПУ. - М.: Машиностроение, 1987.