TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE ENSENADA

1INGENIERÍA EN MECATRÓNICA

RESIDENCIA PROFESIONAL

DISEÑO DE RUTAS PARA UN VEHÍCULO DE RUEDAS DIFERENCIALES MEDIANTE UNA INTERFAZ GRÁFICA

ELABORADO POR:

MIGUEL EDUARDO VENEGAS MONROY

No. CONTROL: **13760039**

LUGAR DE ADSCRIPCIÓN:

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN SUPERIOR DE ENSENADA

ASESOR INTERNO:

ANA YAVENI AGUILAR BUSTOS

ASESOR EXTERNO:

JOSÉ RICARDO CUESTA GARCÍA

ENSENADA, B.C.

DEL 13 DE SEPTIEMBRE DE 2018 AL 13 DE MARZO DE 2019.

Agradecimientos

Este trabajo ha sido posible gracias al apoyo y asesoría del Doctor Ricardo Cuesta, técnico académico del Departamento de Control del posgrado de Física Aplicada del Centro de Investigación Científica y Educación Superior de Ensenada (CICESE). Agradezco también la revisión de este trabajo por parte de la Doctora Ana Yaveni Aguilar Bustos profesora de la licenciatura en Ingeniería en Mecatrónica del Instituto Tecnológico de Ensenada.

Por su tiempo, compresión, apoyo y su incondicional amor agradezco a mi madre Olga Patricia Monroy García, a mi hermana Alejandra Venegas Monroy quienes han estado siempre a mi lado apoyándome con palabras de aliento, muchas gracias.

Por otro lado quiero dar un especial agradecimiento a mi pareja Carolina Armijo quien me ha comprendido y apoyado infinitamente en esta etapa de mi vida, ella ha festejado mis logros, apoyado mis sueños y me ha abrazado en mis derrotas; con este documento quiero decirte que aquí demuestro todas tus palabras y todas tus acciones de apoyo que han servido para forjar este hombre que te ha amado toda la vida, muchas gracias. Este proyecto no acaba aquí, es la continuación de un proyecto de vida que estoy seguro tendrá mucho futuro en robótica móvil.

Resumen

Este trabajo se enfoca en el análisis matemático de la interpolación de un polinomio de tercer orden de Lagrange que ayudará a generar una ruta para un vehículo robotico de ruedas diferenciales con la finalidad de que el robot pueda seguir una ruta definida por el usuario. Así, a partir de ahí, poder generar varias rutas para la navegación del vehículo.

Este trabajo ayudará a que en el futuro ya no se generen trayectorias simples "lineales" por medio de una funciones lineales ya que con ellas se obtienen ángulos agudo, recto y obtuso; en este trabajo se utilizará un polinomio de 3er orden y obtendremos mayor definición en las curvas de la trayectoria lo que ayudará a rodear con mayor facilidad un objeto y generar una ruta con más puntos de precisión de navegación y la pregunta por que no se utiliza un polinomio de orden cuarto, la razón es simple no se ocupa tanto procesamiento computacional.

Índice

Índice de Código y Figura

Capitulo 1 Generalidades del proyecto

1.1 Introducción

“El presente documento tiene como finalidad generar información sobre 1 solo método para generar trayectorias de vehículos autónomos de carga media para facilitar el trabajo del ser humano. ” En el trabajo se describe el uso de la interpolación polinómica de Lagrange en la generación de rutas para un vehículo robot de ruedas diferenciales. Estos son acercamientos a las trayectorias de robots móviles, que en este caso se aplicó a un vehículo tipo oruga.

1.2 Empresa u organización

El lugar donde se realiza la Residencia Profesional es el Centro de Investigación Científica y Educación Superior de Ensenada, Baja California (CICESE). El CICESE es una institución de referencia en el contexto científico nacional e internacional, su excelencia académica apoya el desarrollo nacional, la formación de recursos humanos y contribuye a generar el conocimiento que puede coadyuvar en la solución de problemas que afectan el entorno social y económico de México.([1]).

El proyecto se está llevando a cabo en el laboratorio de control con asesoría del Doctor Ricardo Cuesta.

1.3 Problemas a resolver

- Crear una ruta para un vehículo a partir de un polinomio de tercer orden.
- Trabajar con un orden mucho mayor en el futuro.
- Generar un polinomio de tercer orden que pueda aceptar N valores, en el eje X, Y.
- Generar una interfaz,hecha en Pascal Delphi para que pueda ser multiplataforma y en Dev C para el primer paso donde el usuario puede diseñar la ruta para el vehículo.

Objetivos

Objetivo general

El objetivo general de este trabajo es usar un polígono de orden cúbico, y así obtener rutas con una curvatura mucho más definida para esquivar objetos estáticos o en movimiento y a su vez poder tener información respecto a la velocidad angular y la aceleración angular. Esto se hará con simulaciones en Matlab dentro de un plano XY que será el área de trabajo y se vera en el fondo diferentes mapas que el usuario va poder escoger estos mapas son de ciudades de google map y a su vez el programa te colorea la velocidad y la aceleración que el robot toma.

También a mediano plazo se tiene pensado pasar este programa hecho en Matlab a lenguaje C y aprovechar la librería Grafics.h usando el Dev-C.

Objetivos específicos

- Generar trayectorias complejas en un espacio determinado.
- Obtención de las velocidades y aceleraciones de la curvatura.
- Visualización de velocidades y aceleraciones vía por color.
- Leer vectores muy largos y procesarlos usando el método numérico.

Justificación

Ayudará a entender como un robot navega dentro de una trayectoria definida por el usuario, esto con la ayuda de una interfaz de programación de aplicaciones como ejemplo Google Maps Platform , api de geolocalización.

Capitulo 2 Marco teórico

2.1 Polinomios

$$f(x)=a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 \tag{1}$$

Los polinomios vistos en (1) son expresiones matemáticas utilizadas muy frecuentemente en el modelado y resolución de problemas científicos. En muchos casos, el polinomio representa una forma práctica de crear ecuaciones para resolver determinados problemas, correspondiéndose la solución a estos problemas con la ecuación del polinomio. **[2]**

$$g(x)=a_0+a_1 x+a_2 x^2+\ldots.+a_N x_0^N \tag{2}$$

La serie de polinomios como se ve en (2) son funciones que tienen la forma: donde los coeficientes $a_n, a_{n-1}, \ldots.., a_1, a_0$ son números reales, y n es un número entero positivo que es el grado n del polinomio, la variable independiente x puede tomar tanto val res reales como complejos $X \epsilon \mathbb{C}$ **[2].**

A continuación se muestran diferentes grados de polinomio.

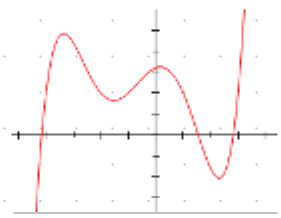
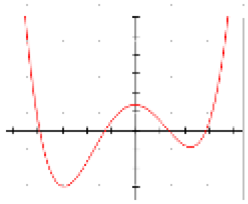
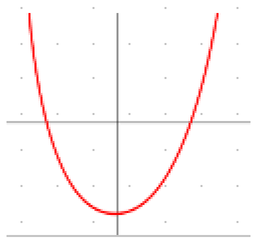
$f(x)=5x^5+6x^2+7x+3$ //Polinomio de grado 5
$f(x)=2x^4+6x^2+7x+3$ //Polinomio de grado 4
$f(x)=9x^3+4x^2$ //Polinomio de grado 3
$f(x)=6x^2+7x$ //Polinomio de grado 2

$f(x)=x+3x$ //Polinomio de grado 1

$f(x)=3$ //Constante. Polinomio de grado 0

Se muestra de manera visual algunos de los diferentes grados de los polinomios que pueden existir.

Tabla 1 . Diferentes grados de los polinomio

a) 	Polinomio de quinto grado $f(x)=ax^5+bx^3+cx+d=0$
b) 	Polinomio de cuarto grado $f(x)=ax^4+bx^2+cx+d=0$
c) 	Polinomio de tercer grado $f(x)=ax^3+bx^2+cx+d=0$
d) 	Polinomio de segundo grado $f(x)=ax^2+cx+d=0$

Capitulo 2.2 Interpolación de un polinomio

Una función de interpolación es aquella que pasa a través de puntos dados como datos, los cuales se muestran comúnmente por medio de una tabla de valores o de una función, en este caso vamos a estar usando vectores que serán almacenados en las variables X,Y.
La interpolación de los datos se puede basar mediante un polinomio. Hay diferentes tipos de interpolaciónes, como son interpolación de Lagrange, Interpolación de Newton, Interpolación de Lagrange mediante puntos de Chebyshev, Interpolación de Hermite, Spline cúbico, entre otras.

$$g(x)=\frac{(x-x_2)(x-x_3)(x-x_4)}{(x_1-x_2)(x_1-x_3)(x_1-x_4)}y_1+\frac{(x-x_1)(x-x_3)(x-x_4)}{(x_2-x_1)(x_2-x_3)(x_2-x_4)}y_2+..$$

$$\frac{(x-x_1)(x-x_2)(x-x_4)}{(x_3-x_1)(x_3-x_2)(x_3-x_4)}y_3+\frac{(x-x_1)(x-x_2)(x-x_3)}{(x_4-x_1)(x_4-x_2)(x_4-x_3)}y_4 \tag{3}$$

2.3 Polinomio cúbicos de lagrange

El documento se basa en la interpolación cubica spline de Lagrange.

La interpolación polinómica (3) de Lagrande, llamado en honor a Joseph-Louis de Lagrange, es una forma demostrar el polinomio que interpola un conjunto de puntos dados.

Un polinomio de orden N que pasa a través de $N+1$ puntos es único. Esto significa que, independientemente de la fórmula de interpolación, todas las interpolaciónes polinomios que se ajustan a los mismos datos son matemáticamente idénticas.

A continuación suponga que dan $N+1$ puntos como $x_0, x_1, ... x_N f_0, f_1, ... f_N$donde los $x_0, x_1, ..$ son las abscisas de los puntos (puntos) dados en orden creciente.
El valor i =0,…,N

Los espacios entre los puntos son arbitrarios. De acuerdo al polinomio de orden N que pasa a través del $N+1$ puntos se puede escribir en una serie de potencias como donde los a_i son coeficientes. El ajuste de la serie de potencias a los $N+1$ puntos dados da un sistema de ecuaciones lineales:

Aunque los coeficientes a_i pueden determinarse resolviendo las ecuaciones simultáneas por medio de un programa computacional, dicho intento no es deseable por dos razones. Primera, se ocupa un programa que resuelva un conjunto de ecuaciones lineales; y segunda, la

solución de la computadora no es tan precisa por eso hay otros métodos que pueden determinar una interpolación polinomio sin resolver las ecuaciones lineales, en este documento se trabajará con la interpolación de Lagrange.
En seguida se presenta la fórmula de interpolación de Lagrange de orden N :

Esta ecuación es equivalente a la serie de potencias, y a su vez a un orden N que lo que genera un polinomio de una magnitud difícil de poder procesar en una computadora.

Por desgracia, la fórmula de interpolación de Lagrange no es adecuada para obtener una forma en serie de potencia, ya que es molesto desarrollar la interpolación de Lagrange en una serie de potencia. En este trabajo nos enfocaremos al uso de una función polinómica de Lagrange de orden cúbico, es decir:

Capitulo 3 Resultados

En este proyecto se utilizo el programa Matlab que es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio.
Está disponible para las plataformas Unix,Windows, Mac osc y Linux , la idea usarlo es que no lo enseñaron en la escuela y ademas llevamos una materia de Métodos Numéricos, pues se nos facilito mucho el utilizar métodos numéricos y enfocarlo mi residencia a que se ha mas analítico, y aprender de las matemáticas lo bellas que son.

Se analizo literatura y pensamientos críticos para llegar a una solución mas viable que pudiera tomar n valores dentro de un polinomio cubico.

Este proyecto fue desatollado en el laboratorio de Control en Física Aplicada,

Aquí es un ejemplo de que se tiene 10 puntos, de los cuales podemos tomar los siguientes segmentos.

1. Del punto 1 hasta el punto 4.
2. Del punto 4 hasta el punto 7.
3. Del punto 7 hasta el punto 10.

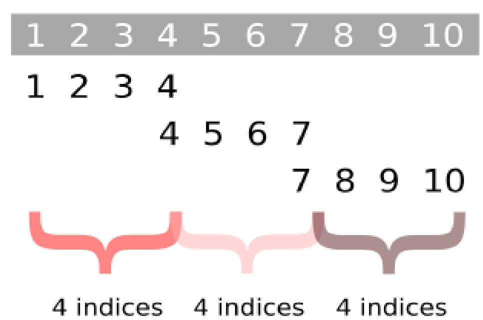


Figura 1: Sustracción de 4 índices.

Observe que se utilizan siempre 4 puntos (Figura 2) de los cuales se utilizan en el polinomio de orden cúbico. El motivo de segmentar una interpolación es poder utilizar el polinomio de orden 3, debido a que hay 10 índices en nuestra variable.
Al estar trabajando únicamente con 4 índices la función del polinómico de Lagrange hace que estemos trabajando en paquetes de cuatro números reales, haciendo posible que nos enfoquemos en tomar paquetes de cuatro existiendo la posibilidad de n paquetes que se ajusten a los 4 paquetes. Fórmula del polinomio cubico (4).

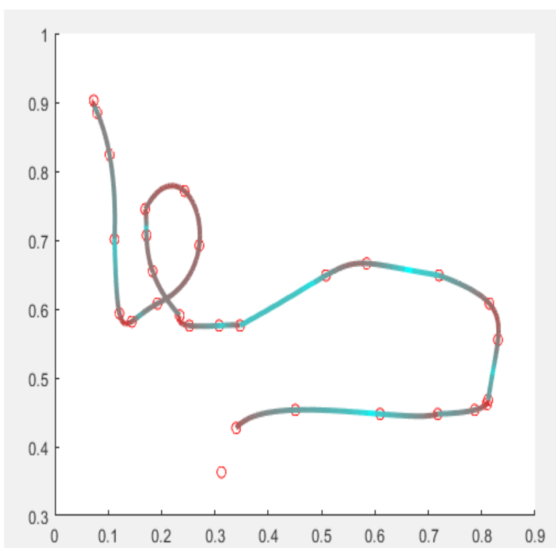
$$g(x)=\frac{(x-x_1)(x-x_2)\ldots(x-x_N)}{(x_0-x_1)(x_0-x_2)\ldots(x_0-x_N)}f_0+\ldots$$

$$\frac{(x-x_0)(x-x_2)\ldots(x-x_N)}{(x_1-x_0)(x_1-x_2)\ldots(x_1-x_N)}f_1+\ldots \tag{4}$$

$$\frac{(x-x_0)(x-x_1)\ldots(x-x_{N-1})}{(x_N-x_0)(x_N-x_1)\ldots(x_N-x_{N-1})}f_N$$

Este análisis nos ayudará a concebir que es posible que se manejen más índices, mientras se respete que la interpolación debe tomar 4 puntos, considerando esto, si el usuario pretende poner 30 índices no será posible visualizar las 10 secciones ya que le faltará llegar a 31 índices.

El usuario debe de ingresar los valores deseados respetando lo que se habló sobre los 4 índices, para que pueda evitar problemas, como se ve en las figuras 3 y 4.



Gráfica del polinomio sin contemplar 31 índices.

Entonces el usuario está forzado a poner 4 índices, y esto da la posibilidad de plasmarlo en una variable $f(x)=a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$, los puntos deseados por el usuario, y esto ayudará a obtener un valor en otra variable denotada como $a_n, a_{n-1}, \ldots, a_1, a_0$ cuya fórmula se muestra a continuación.

$$nc = 15 - mod(15+2,3) \therefore es\, 13 \tag{5}$$

Recordemos que el Operador módulo da como resultado el residuo de una división, y gracias a esto, el Ejemplo de operación de modulo, ya que el usuario si ingresa en la variable n como por ejemplo, que son los datos a utilizar, se verá reflejado en su salida el 13 que será la variable n.

Sabemos que el 3 es el residuo y la variable M es el divisor siempre se le sumará 2, con esto podremos obtener un valor al cual se le estará restando 3 en el ciclo del **For** donde tendremos un incremento del $1\, al\, 3$, esto se ve en la formula 6.

$$nc = M - mod(M+2,3) \tag{6}$$

La finalidad del incremento $1\, al\, 3$ es utilizarlo en los índices que sabemos que se emplean solo 4, esto posibilitará el tomar valores dentro de esa sección y se le van a ir sumando, como ejemplo.

$$1, 1+1=2, 2+1=3, 3+1=4, 4+1=5\ldots$$

Se muestra como el incremento se puede utilizar para ir aumentando los valores deseados dentro del ciclo del **For.**

En esta sección se muestran los resultados obtenidos al hacer de cada coordenadas X,Y, por separado recordemos que cada coordenadas X,Y tiene sus vectores, y esos vectores son ingresados por el usuario, como el ejemplo que hemos estado usando de 31 puntos, esos 31 puntos se ven reflejados en cada gráfica de cada coordenada X,Y.

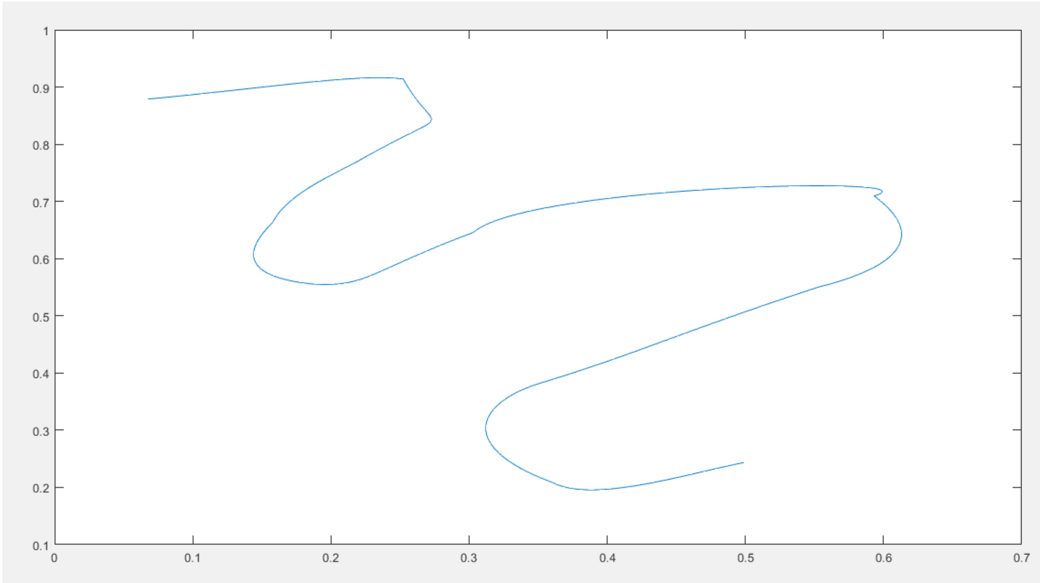


Figura 2: Trayectoria de las dos variables de int_y int_x unidas.

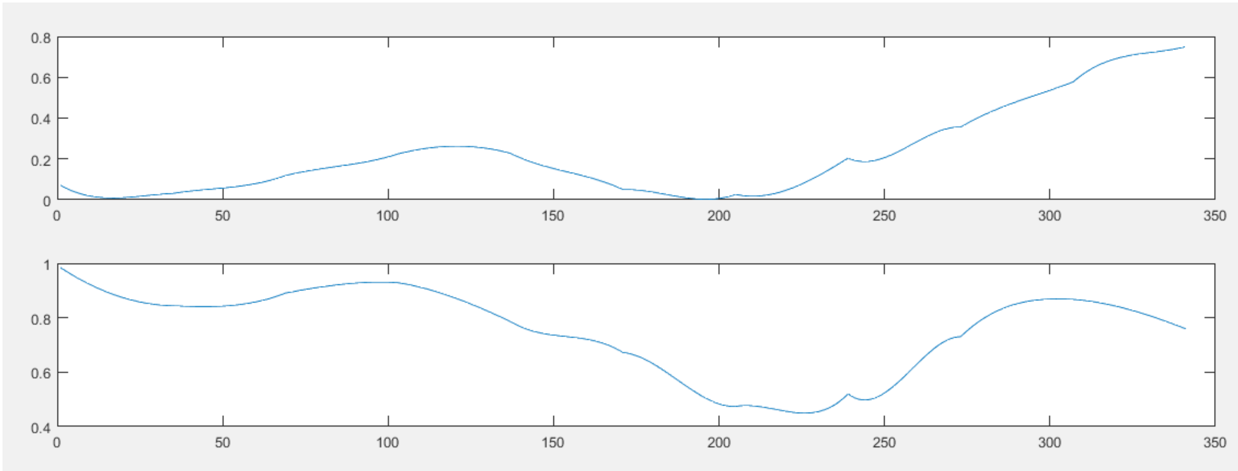


Figura 3: Trayectoria de las dos variables de mi función de mi_interpolación gráficas individualmente.

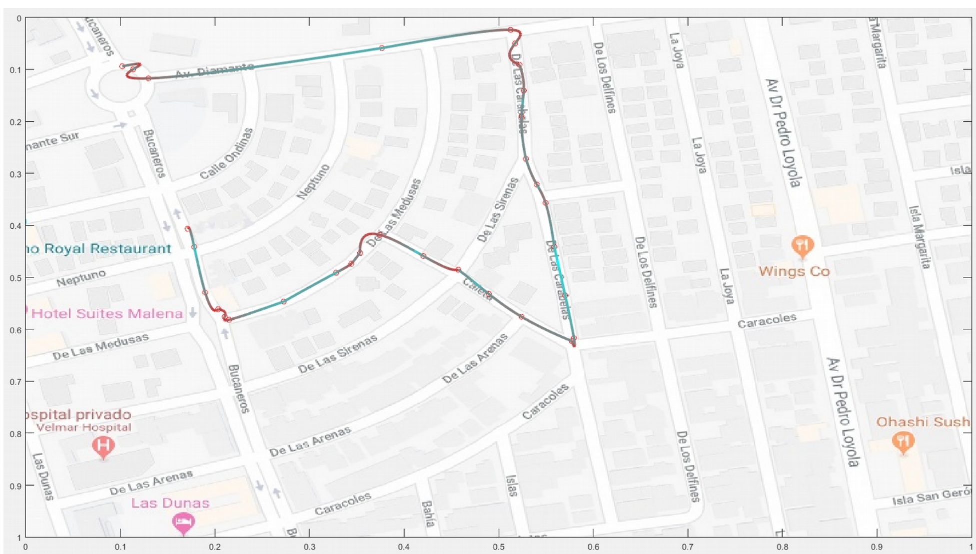


Figura 4: Trayectoria hecha por el usuario

A continuación se presenta el código 1 que contiene la función de ingreso de datos como ejemplo cuantos puntos quiere asignar en su ruta eso se ve en la parte de [X,Y] =ginput(31), en esa sección el usuario puede poner mas puntos y así generara una trayectoria de longitud muy larga.

De igual manera también se gráfica el eje X, Y, que serán llamada de mi función mi_interpolación(X,numero_deseado) o mi_interpolación(Y,numero_deseado).

Ese numero_deseado es controlado por el usuario ya que puede tener la opción de tener una mala calidad o una buena calidad en la ruta.

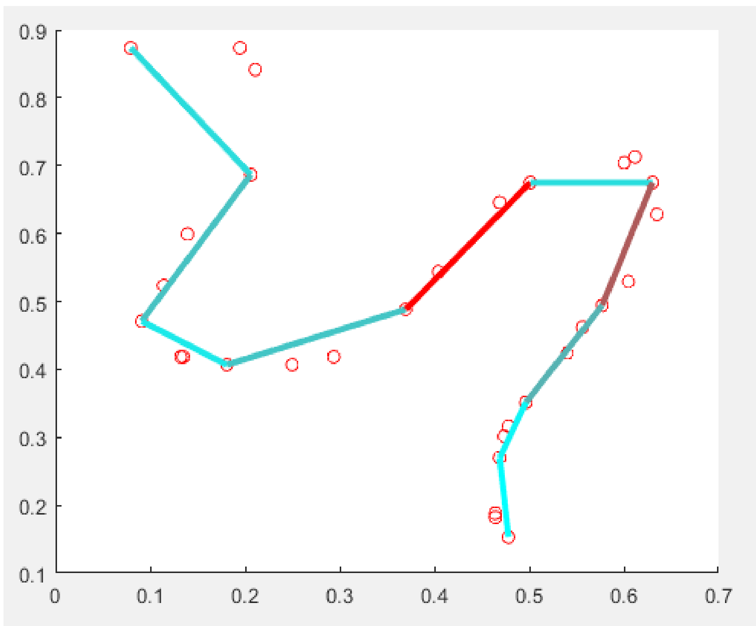


Figura 5: Trayectoria de las dos variables de int_y int_x , con numero_deseado=2

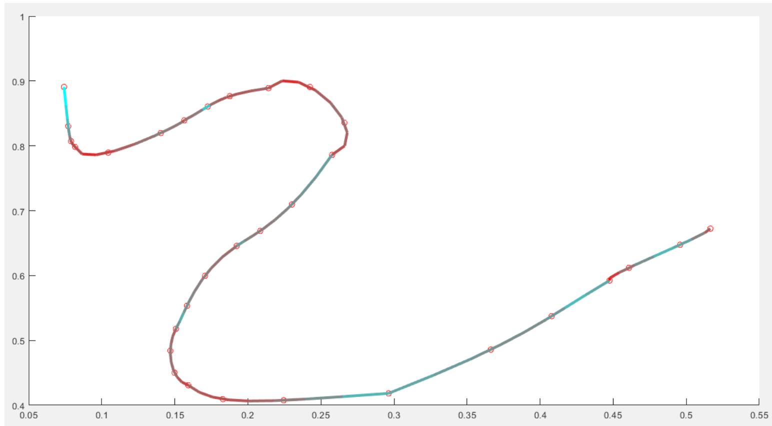


Figura 6: Trayectoria de las dos variables de int_y int_x , con numero_deseado=10

Aquí plantearemos la ecuación de la curvatura que sera la variable K, y la Fórmula de la curvatura es $K=\frac{\dot{x}*\ddot{y}*-\dot{y}\,\ddot{x}}{(\dot{x}^2)+(\dot{y}^2)^{3/2}}$, la cual obtenemos una **Kmax**, que es el valor máximo obtenido en la curvatura, recordemos que la curvatura tiene la primer derivada y la segunda derivada de los las coordenadas (X,Y).

Recordemos que la primer derivada es la aceleración y la segunda derivada es la velocidad, con esos valores se verán reflejados en mi dx (que es mi derivada de x) y mi ddx (que es mi segunda derivada).

```
clear all; clc; clf;
%datos X
%subplot(3,1,3)
[X,Y] = ginput(31);
% X = [1,3,-1,4,3,-4,1];
% Y = [2,4,3,-5,1,2,3];
 grosor_de_linea=3;
[int_x, dx,ddx] = mi_interpolación(X,35);%35 numero deseado recuerda deben ser
maximo 35 ya
                                          % que si ponemos 2 la definicion del
                                          % poligono es muy

                                                 de mala calidad.
[int_y, dy,ddy] = mi_interpolación(Y,35);%35
K=abs(((dx.*ddy)-(dy.*ddx))./(dx.^2+dy.^2).^(3/2));
K = log(K)-min(log(K))
Kmax=max(K);
% subplot(3,1,1)
% plot(int_x)
% subplot(3,1,2)
% plot(int_y)
% subplot(3,1,3)
% plot(int_x,int_y)
hold on
plot(X,Y,'ro')
for i=1:length(K)-1
    plot([int_x(i),int_x(i+1)],[int_y(i),int_y(i+1)],...
        'color',[K(i)/Kmax 1-K(i)/Kmax 1-K(i)/Kmax],'LineWidth',grosor_de_linea)
end
```

Figura 7: Código Función ingreso de datos.

A continuación se presenta el Código 8 correspondiente a la primer derivada, la cual lo usaremos para calcular la aceleración.

```
function Dsalida =derivada(T,D,z)
b=1;
     d = T(b);
x = T(b+1);
y = T(b+2);
w = T(b+3);

D1 = D(b)/(((T(b)-T(b+1))*(T(b)-T(b+2))*(T(b)-T(b+3))));
D2 = D(b+1)/(((T(b+1)-T(b))*(T(b+1)-T(b+2))*(T(b+1)-T(b+3))));
D3 = D(b+2)/(((T(b+2)-T(b))*(T(b+2)-T(b+1))*(T(b+2)-T(b+3))));
D4 = D(b+3)/(((T(b+3)-T(b))*(T(b+3)-T(b+1))*(T(b+3)-T(b+2))));

    Dsalida=D1*(3*z^2-2*z*y-2*z*x+x*y-2*z*w+y*w+x*w)+...
            D2*(3*z^2-2*z*y-2*z*d+y*d-2*z*w+y*w+d*w)+...
            D3*(3*z^2-2*z*x-2*z*d+x*d-2*z*w+x*w+d*w)+...
            D4*(3*z^2-2*z*x-2*z*d+d*x-2*z*y+x*y+d*y);
```

Figura 8: Código de Derivada

El Código 9, se ve la doble derivada que es la velocidad obtenida del polinomio cubico de interpolación de Lagrange, este dato nos va servir para poder usarlo en la la Fórmula de la curvatura.

```
function DDsalida =segundaderivada(T,D,z)
b=1;
d = T(b);
x = T(b+1);
y = T(b+2);
w = T(b+3);

DD1 = D(b)/(((T(b)-T(b+1))*(T(b)-T(b+2))*(T(b)-T(b+3))));
DD2 = D(b+1)/(((T(b+1)-T(b))*(T(b+1)-T(b+2))*(T(b+1)-
T(b+3))));
DD3 = D(b+2)/(((T(b+2)-T(b))*(T(b+2)-T(b+1))*(T(b+2)-
T(b+3))));
DD4 = D(b+3)/(((T(b+3)-T(b))*(T(b+3)-T(b+1))*(T(b+3)-
T(b+2))));

   DDsalida=DD1*(6*z-2*y-2*x-2*w)+...
            DD2*(6*z-2*y-2*d-2*w)+...
            DD3*(6*z-2*x-2*d-2*w)+...
            DD4*(6*z-2*x-2*d-2*y);
```

Figura 9: Código de Segunda Derivada

A continuación se presenta el código 10 que corresponde a la función de mi_interpolación, este código trata de como generar una gráfica con varios vectores, estos vectores serán X,Y y estaremos ingresando datos a un ciclo for tendrá un polinomio cubico con una interpolación polinómica de Lagrange. Esta sección hace interesante al polinomio cubico por que se podrá meter n valores que serán agregados por el usuario respectando que al seleccionar se han 4 valores.

```
function [aa pd sd] = mi_interpolación(D,numero_des)
valor_division_incremento=numero_des-1;%pedacitos
%D = [929 902 860 760 780 800 850 876 886 990 800 700 700];
T=1:length(D);
  j=1;
  M = length(T);
  nc = M-mod(M+2,3);
for b=1:3:(nc-3)
    resta=T(b+3)-T(b);
    incremento_real=resta/valor_division_incremento;
    if b == (nc-3)
        c = 0;
    else
        c = 1;
    end
       for z=T(b):incremento_real:(T(b+3)-c*incremento_real)
         GG_1=(D(b)/1)*((z-T(b+1))*(z-T(b+2))*(z-T(b+3)))/(((T(b)-T(b+1))*(T(b)-
T(b+2))*(T(b)-T(b+3))))+...
              (D(b+1)/1)*((z-T(b))*(z-T(b+2))*(z-T(b+3)))/(((T(b+1)-T(b))*(T(b+1)-
T(b+2))*(T(b+1)-T(b+3))))+...
              (D(b+2)/1)*((z-T(b))*(z-T(b+1))*(z-T(b+3)))/(((T(b+2)-T(b))*(T(b+2)-
T(b+1))*(T(b+2)-T(b+3))))+...
              (D(b+3)/1)*((z-T(b))*(z-T(b+1))*(z-T(b+2)))/(((T(b+3)-T(b))*(T(b+3)-
T(b+1))*(T(b+3)-T(b+2))));
                Ds = D(b:b+3);
                Ts = T(b:b+3);
                pd(j)=derivada(Ts,Ds,z);
                assignin('base','primerDerivada_pd',pd)
                sd(j)=segundaderivada(Ts,Ds,z);
                assignin('base','segundaDerivada_sd',sd)
        aa(j) = GG_1;
        j= j+1;
       end
end
```

Figura 10: Código Interpolación

Capitulo 5 Conclusiones

La idea básica del programa es que el usuario pueda generar una trayectoria desarrollada por él mismo con base en un polinomio cúbico.
La trayectoria muestra con colores la curvatura de cada una de las curvas de la trayectoria, calculando la aceleración y la velocidad, lo cual servirá para conocer el movimiento angular de una curva.
Al analizar el potencial de este proyecto se puede prever que es muy grande ya que se buscará el perfeccionamiento de las trayectorias para vehículos móviles, esto con la idea de poder buscar dentro de esa rama nuevos métodos de generar trayectoria y siempre buscando la optimización en generación de trayectorias y visualización de información de los datos obtenidos como las derivadas de primer orden y segundo orden. Al saber esos datos, podremos en un futuro generar una interfaz de usuario que pueda mostrar en tiempo real las velocidades y la aceleración del vehículo, esto con la ayuda de sensores y microprocesadores. Finalmente todo esto ayudará a que las trayectorias puedan adaptarse a entornos más complejos y responder a situaciones más difíciles en el entorno de trabajo en el que se vaya a usar el vehículo robotizado.

Un punto muy importante es que este proyecto se tiene que migrar a un lenguaje que pueda interactuar con robots, y un claro ejemplo es al lenguaje ensamblador. Actualmente el proyecto está hecho en Matlab. La finalidad es migrar del lenguaje Matlab al lenguaje C para poder finalizar en Ensamblador,

Competencias desarrolladas

- Diseñar aplicaciones enfocadas a trayectorias en robots móviles utilizando lenguaje Matlab.
- Describir y aplicar un método numérico para generar interpolaciónes de orden cúbico por medio de Lagrange.

Bibliografías

[1]: CICESE. (2018). Acerca del CICESE. 2019, de CICESE Sitio web: https://www.cicese.edu.mx/welcome/acerca/

[2]: John Wiley & Sons, Inc., Hoboken(NJ),USA, Polinomios,curvas de ajustes e interpolación, 2015

[3] Wikipedia. (2019). Polinomios. 2019, de Wikipedia Sitio web: https://es.wikipedia.org/wiki/Polinomio

[4]: Schoichiro Nakamura. (2018). Metodos Numericos Aplicados con Software . The Ohio State University: Pearson Educación.

Anexos

Anexo A.

(Carta de autorización por parte de la empresa u organización para la titulación y otros si son necesarios)

Anexo B.

Evaluación de residencia.

Anexo C.

Evaluación de residencia.