



Escuela de  
Ingeniería y Arquitectura  
Universidad Zaragoza

# Trabajo fin de Grado

## Simulación de trayectorias y modelos de medición en plataformas multisensor.

Autor: Pedro José Pérez Torrero

Director: Daniel Gutiérrez Gómez

Ponente: José J. Guerrero Campo

Grado en ingeniería de las tecnologías industriales

Noviembre 2016





## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D<sup>a</sup>. \_\_\_\_\_,

con nº de DNI \_\_\_\_\_ en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)  
\_\_\_\_\_, (Título del Trabajo)

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, \_\_\_\_\_

Fdo: \_\_\_\_\_

# RESUMEN

El objetivo de este trabajo es generar una trayectoria simulada en el entorno virtual de Matlab mediante B-splines cúbicos, para la obtención de este objetivo se han introducido conceptos de teoría y ejemplo de otras trayectorias más sencillas, los cuales nos van a ayudar al mejor entendimiento y comprensión de nuestra trayectoria generada.

Cabe destacar que todas las trayectorias creadas en este trabajo van a ser generadas de manera aleatoria, es decir los puntos de paso de cada trayectoria van a ser elegidos aleatoriamente por nuestro programa de cálculo.

La creación de esta trayectoria es debido a que se desea su implementación en un vehículo de movimiento en 3D. El hecho de querer implementar un vehículo de movimiento en 3D nos produce la necesidad del modelado de una serie de sensores, para que dicho vehículo sea capaz de percibir su entorno y su movimiento, en este trabajo se han simulado tres sensores: Sensor de velocidad, Sensor de aceleración y GPS.

En definitiva se va a generar un vehículo simulado de movimiento 3D en el entorno virtual de Matlab, que realice una trayectoria acelerada y curvilínea y posea una serie de sensores.

# TABLA DE CONTENIDO

---

1	Introducción.....	4
1.1	Objetivo y alcance .....	4
1.2	Herramientas utilizadas .....	5
1.3	Fases del desarrollo.....	5
1.4	Motivación .....	6
2	Generación de trayectoria rectilínea .....	6
2.1	Trayectoria rectilínea con velocidad constante .....	7
2.2	Trayectoria acelerada.....	9
3	Movimiento de un sólido en $R^3$ .....	13
3.1	Introducción .....	13
3.2	Matrices de Rotación .....	13
3.2.1	Matrices de rotación en el espacio $SO$ .....	13
3.2.2	Coordenadas exponenciales .....	16
3.2.3	Obtención coordenadas exponenciales a partir de la matriz de rotación en $SO(3)$ .....	17
3.2.4	Conclusión .....	18
3.3	Matrices de traslación.....	18
3.3.1	Representación homogénea .....	19
3.3.2	Coordenadas exponenciales. ....	20
3.3.3	Obtención de coordenadas exponenciales a partir de la matriz de traslación en $SE(3)$ .....	21
3.3.4	Conclusión .....	22
4	Representación de una trayectoria mediante Splines .....	23
4.1	Introducción .....	23
4.2	B-Splines cúbicos sin matriz de rotación.....	24
4.2.1	Descripción de la ecuación de movimiento .....	24
4.2.2	Condiciones especiales.....	26
4.2.3	Ejemplo.....	28
4.3	B-Splines cúbicos con matriz de rotación.....	29
4.3.1	Descripción de la ecuación de movimiento .....	29
4.3.2	Matrices de control .....	30
4.3.3	Condiciones especiales.....	31
4.3.4	Adaptación de los puntos de control .....	32
4.3.5	Rotación .....	33
4.3.6	Ejemplo.....	34

5	Sensores .....	36
5.1	Sensor de velocidad .....	37
5.1.1	Tipos de sensores de velocidad.....	37
5.1.2	Cálculo teórico.....	37
5.1.3	Ejemplo.....	38
5.2	Sensor de aceleración .....	39
5.3	GPS .....	41
5.3.1	Errores atmosféricos. ....	41
5.3.2	Error en el reloj.....	43
5.3.3	Errores de multitrayectoria. ....	43
5.3.4	Posición de los satélites. ....	44
5.3.5	Modelado .....	44
6	Conclusiones.....	46
7	bibliografía .....	48
Anexo A:	Índice funciones de Matlab.....	50
A.1	Trayectoria rectilínea uniforme .....	50
A.2	Trayectoria Acelerada .....	50
A.3	B- Splines cúbicos sin matriz de rotación.....	51
A.4	B-Splines cúbicos con matriz de rotación .....	51

# 1 INTRODUCCIÓN.

---

## 1.1 OBJETIVO Y ALCANCE

En los últimos años se está produciendo una revolución en el campo de la robótica, siendo el control de vehículos no tripulados un importante campo de investigación dentro de él. La singularidad de este campo es dotar a dichos vehículos de una cierta autonomía, es decir que sean capaces de percibir su entorno y responder ante cualquier dificultad que esté presente.

¿Cómo se consigue esta autonomía? Hoy en día estos vehículos están provistos de una serie de diferentes sensores capaces de detectar diferentes obstáculos, medir tanto la velocidad como la aceleración e incluso son capaces de detectar la distancia que los separa de los diferentes objetos. Todo esto es posible gracias a la revolución tecnológica que se ha producido tanto en cámaras digitales, Kinect o en los diferentes sensores de medición o localización espacial.

Otro aspecto muy importante en el campo de vehículos no tripulados, es la trayectoria que va a seguir dicho vehículo, dicha trayectoria no debe ser rectilínea, es decir que le vehículo se mueva en línea recta de un punto a otro. En algún caso puntual sí que nos interesa esta trayectoria rectilínea, pero en el caso de los vehículos 3D va a predominar una trayectoria curvilínea que una los diferentes puntos de paso del vehículo, dicha trayectoria es más natural para un movimiento en 3D.

Se puede observar que el campo de los vehículos no tripulados, es un campo muy amplio ya que en un mismo conjunto se tienen que modelar diferentes aspectos. Por lo que el objetivo de este trabajo va a ser el desarrollo de un entorno de simulación, que permita crear un vehículo simulado que realice diferentes trayectorias aleatorias. Dicho vehículo va a poseer la particularidad de que sea posible modelar en él diferentes tipos de sensores, capaces de percibir el entorno así como las características del movimiento.

El trabajo esta desarrolla en el departamento de informática e ingeniería de sistemas de la Universidad de Zaragoza, concretamente en el área de visión por computador, por lo que confiere una gran utilidad la creación de un modelo en el cuál puedan implementarse diferentes cámaras para la medición del entorno.

En definitiva se va a modelar un programa que nos defina una trayectoria aleatoria, en el cual se van a implementar un sensor de velocidad, otro de aceleración y un GPS.

## 1.2 HERRAMIENTAS UTILIZAS

Para la realización de este trabajo se ha usado principalmente el programa informático de Matlab, este programa posee una gran utilidad para nuestro caso, debido a que nos permite trabajar de forma gráfica con una gran facilidad, a la vez que se implementa un código en él.

Además de la facilidad de trabajo que nos proporciona el programa de Matlab, cuenta con la ventaja de que posee una gran colección de toolboxes predominando la toolbox de Peter Corke la cual realiza una simulación de un vehículo con diferentes sensores, la cual me ha servido como introducción al campo de trabajo.

## 1.3 FASES DEL DESARROLLO.

El desarrollo de este proyecto puede diferenciarse en las siguientes fases de trabajo:

1. Estudio del problema, antes de empezar a abordar el problema planteado se ha necesitado un estudio previo del problema y visualización de diferentes casos similares. Esto es debido a que con anterioridad del inicio del proyecto no poseía los conocimientos necesarios para iniciar a abordar el problema. Además ha sido necesaria una formación en el lenguaje de programación de Matlab ya que con anterioridad nunca había trabajado profundamente con este programa.

Para la introducción al problema abordar y a la vez en la del lenguaje de programación de Matlab, la toolbox de Peter Corke ha sido de gran ayuda.

2. Realización de las primeras simulaciones de trayectorias. En esta fase se han desarrollado dos trayectorias de cálculo las cuales no poseían complejidad matemática, pero me han proporcionado una introducción al modelado de diferentes trayectorias, así como al espacio de trabajo de trabajo de Matlab y su utilización.
3. Formación complementaria en transformaciones espaciales y movimiento del sólido rígido. En esta fase sobretodo se ha realizado un profundo estudio de las matrices de rotación y las transformaciones de la misma en los diferentes espacios de cálculo, debido a su posterior utilidad en las ecuaciones de generación de las trayectorias.
4. Generación de los algoritmos de las trayectorias. En esta fase se ha desarrollado los algoritmos de control de las trayectorias utilizando la información sobre matrices de traslación del anterior apartado.

Estos algoritmos se han desarrollado en el programa de cálculo de Matlab y su función es la realizar una trayectoria aleatoria en 3D que componga una



componente de rotación, para así conseguir la simulación del sistema. Dicha trayectoria ha sido generada mediante un método de B-splines cúbicos tal como describe [Lovengrove et al, 2013]

5. Modelado de los sensores. Una vez que ya se poseen los modelos de trayectorias, es necesario realizar la implementación de los sensores en dicha trayectoria, para lo cual primero se tiene que realizar un profundo estudio acerca del funcionamiento de los sensores y los errores que pueden producirse en ellos.
6. Conclusión, en esta fase se ha realizado una reflexión sobre la elaboración del proyecto, esta reflexión sobre todos e ha basado en los diferentes tipos de trayectorias disponibles, así como la implementación suya en diferentes casos.

## 1.4 MOTIVACIÓN

Como se ha comentado anteriormente el campo de los vehículos no tripulados es un campo en auge en estos momentos, por lo que es un campo interesante para el estudio. La idea surge a partir del artículo de Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras [Lovengrove et al, 2013], en el cual se describe un modelo de trayectoria muy interesante a realizar por un vehículo no tripulado. En este artículo se muestran las ecuaciones teóricas para la realización de una trayectoria mediante splines, pero no muestra ninguna simulación de dicha trayectoria, por lo que es un elemento interesante de investigación la simulación de dicha trayectoria y como se adaptan los sensores a dicha trayectoria.

## 2 GENERACIÓN DE TRAYECTORIA RECTILÍNEA

---

En este apartado se va a realizar una introducción al campo de las trayectorias en vehículos simulados, las trayectorias que se van a ver a continuación son unas trayectorias que matemáticamente hablando son más sencillas, pero nos proporcionan una valiosa información que más tarde extrapolaremos a las trayectorias mediante splines.

Por lo que las trayectorias que mostraremos a continuación serán dos trayectorias rectilíneas una de velocidad constante y otra acelerada, las cuales van a ser representadas en un sistema de referencia 2D de dimensiones 20x20 metros, esto quiere decir que los valores aleatorios de los puntos de control van a tener solo componente x e y, cuyos valores estarán comprendidos entre 0 y 20

## 2.1 TRAYECTORIA RECTILÍNEA CON VELOCIDAD CONSTANTE

Se ha realizado una trayectoria rectilínea de velocidad constante (en el ejemplo será de 2m/s), cuyos puntos de paso son generados aleatoriamente por el programa de computación.

Una vez se conocen los puntos de paso y la velocidad, se calcula el tiempo que necesita el vehículo para ir del primer punto al segundo, al ser un movimiento de velocidad constante dicho tiempo se calculara mediante la ecuación 2.1

$$t_1 - t_0 = \frac{\text{norma}(q_1 - q_0)}{v} \quad (2.1)$$

El ejemplo que se ha realizado consta de 6 puntos de paso por lo que lo dividiremos en 5 trayectorias diferentes entre cada par de puntos de paso. Para el cálculo de la primera trayectoria  $t_0=0$  segundos y  $t_1$  se obtendrá de resolver la ecuación 2.1, el cual será el tiempo que necesita el vehículo para realizar la primera trayectoria.

Una vez se conoce este tiempo se procederá al cálculo del resto de tiempos de paso por los diferentes puntos de control, esto se calcula iterando la formula anterior donde  $t_1$  se convierte en  $t_0$  de la siguiente iteración, repetiremos este proceso de iteración para cada una de las diferentes trayectoria.

Conociendo el tiempo que tarda en realizar cada iteración, se procederá a dividir cada trayectoria en diferentes intervalos de muestreo (en nuestro ejercicio serán 5), es decir la creación de 4 puntos intermedios entre cada par de puntos de control. La posición será calculada mediante la fórmula 2.2, donde  $t$  es el tiempo en cada uno de los puntos intermedios creados, este tiempo varia de una manera proporcional al tiempo que se tarda en realizar la trayectoria entre los dos puntos de control, en el ejemplo se ha decidido realizar una progresión de 0.2.

$$q(t) = q_0 + \frac{q_1 - q_0}{t_1 - t_0} * (t - t_0) \quad (2.2)$$

Este paso de dividir la trayectoria en varios segmentos más pequeños se utiliza para dotar a la trayectoria simulada de una mayor precisión, en este tipo de trayectoria no se aprecia un cambio significativo al realizarlo, pero como se verá más adelante es un paso primordial cuando trabajamos con trayectorias más complejas.

Una vez se ha calculado la posición en los diferentes intervalos de tiempo de la primera trayectoria, se procede al cálculo del resto de trayectorias, el método operativo es el mismo que el que se ha descrito para la primera trayectoria.

El código de cálculo utilizado puede apreciarse en el Anexo A, a continuación se va a mostrar un ejemplo de trayectoria aleatoria, generado a raíz del código realizado en el entorno de Matlab:

- Los puntos de paso aleatorios generados por el programa son los siguientes:

P1	P2	P3	P4	P5	P6
16.29	2.53	12.64	5.57	19.15	3.15
18.11	18.23	1.95	10.93	19.32	19.41

- Una vez se conocen los puntos de paso y la velocidad (2 m/s), se procede al calculo del tiempo de paso (en segundos) del vehículo por cada uno de dichos puntos, el cual es:

P1	P2	P3	P4	P5	P6
0	6.87	16.47	22.19	30.16	38.16

- Una vez se conocen los tiempos y puntos de paso se procede a calcular la posición del vehículo entre cada intervalo, en nuestro caso entre cada par de puntos de paso se han calculado 4 posiciones adicionales, al ser un movimiento rectilineo uniforme no haria falta su calculo pero asi dotamos a la trayectoria de una mayor precisión.
- Calculados todos los puntos de paso del vehículo, podemos observar el resultado gráfico de la trayectoria simulada en la en la figura 1.

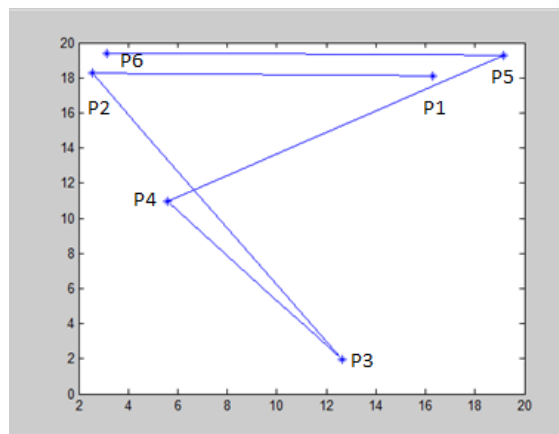


Figura 1 Ejemplo de trayectoria rectilínea

Se observa que el ejemplo representado en la figura 1 no nos serviría para modelar una trayectoria de un vehículo, debido a que produce un movimiento poco natural en el que los giros se producen de una manera repentina siendo que lo ideal sería realizarlos paulatinamente. Por lo que a continuación se va a proceder al cálculo de otras trayectorias que se adecuen mejor a las necesidades que nosotros poseemos en la vida real.

## 2.2 TRAYECTORIA ACELERADA

En este apartado se ha procedido a realizar una trayectoria acelerada simulada en la que el programa, nos proporciona de manera aleatoria los puntos de paso del vehículo y la velocidad del vehículo en dichos puntos. El procedimiento de cálculo es similar al anterior, aunque en este caso la trayectoria no será puramente rectilínea, debido a que al introducir nuevas variables la ecuación para el cálculo de posición es de tercer grado.

La trayectoria comenzara en el instante  $t_0$ , que siempre será igual a 0 segundos, conociendo este dato, las velocidades y los puntos de paso, se procederá al cálculo del tiempo de paso por el primer punto, mediante la ecuación 2.3.

$$t_1 - t_0 = \frac{\text{norma}(q_1 - q_0)}{(v_1 + v_0)/2} \quad (2.3)$$

En este caso la velocidad ya no es constante por lo que para el cálculo del tiempo trabajaremos con la velocidad media de la trayectoria. Con esto ya conocemos el tiempo de paso por el primer punto para conocer el resto de tiempos, se procederá a la iteración de la ecuación 2.3, donde  $t_0$  pasara a ser el tiempo del primer punto, dicho cambio también se aplicara a la velocidad y posición.

Una vez ya se conocen los tiempos de paso de todos los puntos, se procede a calcular posición del vehículo en cada instante de tiempo que se determine, para realizar esta determinación se ha procedido a una normalización del tiempo entre cada par de puntos de paso, dicho tiempo normalizado se calculara de la siguiente manera:

$$\tau = \frac{t - t_0}{t_1 - t_0} \quad (2.4)$$

$\tau$  esta comprendido entre 0 y 1, siendo 0 cuando se encuentra en la posición inicial y 1 cuando se encuentra en la final. En la práctica no se ha usado la ecuación 2.4 sino que se han ido dando valores al tiempo normalizado siguiendo una progresión de paso constante desde 0 hasta 1, cuanto más pequeño es el paso de esta progresión se conseguirá una mayor precisión en la trayectoria resultante, debido al mayor número de instantes calculados, lo que dotara de un mayor número de posiciones conocidas en la trayectoria del vehículo.

Definido el paso del tiempo normalizado se procede a la obtención de la ecuación de posición del vehículo, la cual será un polinomio temporal de grado 3 que poseerá la siguiente forma:  $q(t) = at^3 + bt^2 + ct + d$ , la cual posee 4 constantes que se resolverá, según [Guerrero-2013] aplicando las condiciones de contorno impuestas por nuestro problema ( $q_0, q_1, v_0$  y  $v_1$ ). Se resuelve el sistema con las condiciones de contorno y cambiando la variable  $t$  por la del tiempo normalizado, así se obtendrá la

ecuación 2.5, con la que se hallara la posición del vehículo para cada tiempo normalizado.

$$q(\tau) = (v_1 + v_0 - 2q_1 + 2q_0)\tau^3 + (-v_1 - 2v_0 + 3q_1 - 3q_0)\tau^2 + (v_0\tau) + q_0 \quad (2.5)$$

Como se ha definido antes se tiene que dividir la trayectoria en intervalos entre cada par de puntos de control, por lo que aplicaremos la ecuación 2.5 a cada uno de estos intervalos, el número de posiciones que debemos calcular en cada intervalo va a depender del avance del tiempo normalizado.

Se puede observar que la ecuación que nos determina la posición del vehículo para cada instante de tiempo es de tercer grado, por lo que la trayectoria a recorrer por el vehículo ya no será completamente rectilínea sino que posee una mayor complejidad, por lo que es una trayectoria mucho más “natural” que la descrita en el apartado anterior.

A continuación se va a mostrar un ejemplo de trayectoria, el cuál ha sido realizado mediante la aplicación del código que se muestra en el Anexo A. Dicho ejemplo contiene 6 puntos de control generados aleatoriamente por el programa, los cuales se pueden observar en la tabla 1.

La velocidad del vehículo en cada punto de control es calculada de manera aleatoria por el programa de cálculo, esta velocidad ya no es constante sino que variara debido a que se está trabajando con un movimiento acelerado, en la tabla 1 se pueden apreciar los valores escogidos por el programa.

	P1	P2	P3	P4	P5	P6
x	17.857	16.424	2.153	10.596	6.708	8.762
y	6.639	0.833	11.901	8.376	12.45	14.717
v	5.18	5.788	6.453	9.902	8.198	4.132
t	0	1.09	4.04	5.159	5.781	6.278

Tabla 1

Los valores del tiempo de paso del vehículo por cada punto de control también se han introducido en la tabla 1, realmente estos valores no son necesario para el cálculo de la posición del vehículo, debido a que esta se calcula mediante el tiempo normalizado entre dos puntos, para lo cual no es necesario introducir el tiempo en el punto inicial y final. Aunque el tiempo de paso sí que es un dato interesante para el análisis de la trayectoria acelerada porque es un indicador de las variaciones de velocidad que sufre el vehículo, además de proporcionar una información muy valiosa para el control de nuestro de vehículo porque se conoce la posición de ste en determinados instantes de tiempo

Una vez conocidos los datos de la tabla 1 se procede al cálculo de interpolación de la posición mediante la ec 2.5, entre cada par de puntos de paso. Lo primero que se debe fijar para realizar este cálculo es el número de posiciones que se calcularán entre cada punto de control, es decir los diferentes instantes de tiempo normalizado para los que calcularemos una posición. Dicho tiempo está programado para que avance con cada iteración de cálculo de una posición entre dos puntos de control, a este avance se le denominará paso. A continuación se va a mostrar cómo influye la modificación de este paso para trayectorias que poseen los mismos puntos de control y velocidades que los mostrados en la tabla 1.

- Se empezará mostrando la trayectoria con un paso de 0.2, es decir entre cada par de puntos de control se va a calcular la posición para  $\tau=0, 0.2, 0.4, 0.6, 0.8$  y 1, como se ha explicado anteriormente los puntos generados con  $\tau=0$  y 1 pertenecen a los puntos de control por lo que no será necesario su cálculo mediante la ec 5. La trayectoria resultante se puede apreciar en la figura 2, se puede observar que al coger un paso “grande” la trayectoria no resulta natural, debido a la poca cantidad de puntos generados entre cada par de puntos de control para trazar la trayectoria.

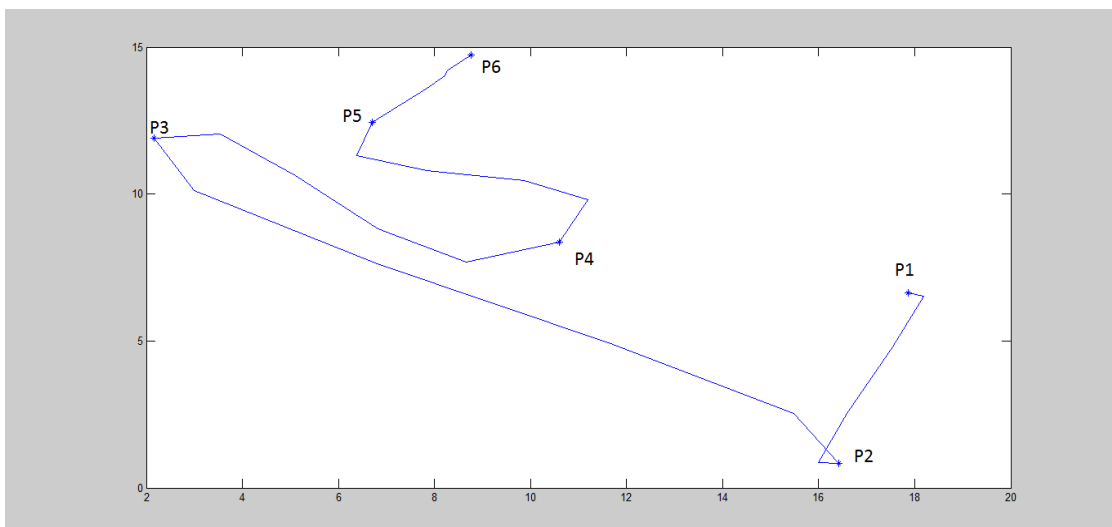


Figura 2 trayectoria acelerada de paso 0.2

- En la figura 3 se va a mostrar una trayectoria con un paso de 0.1

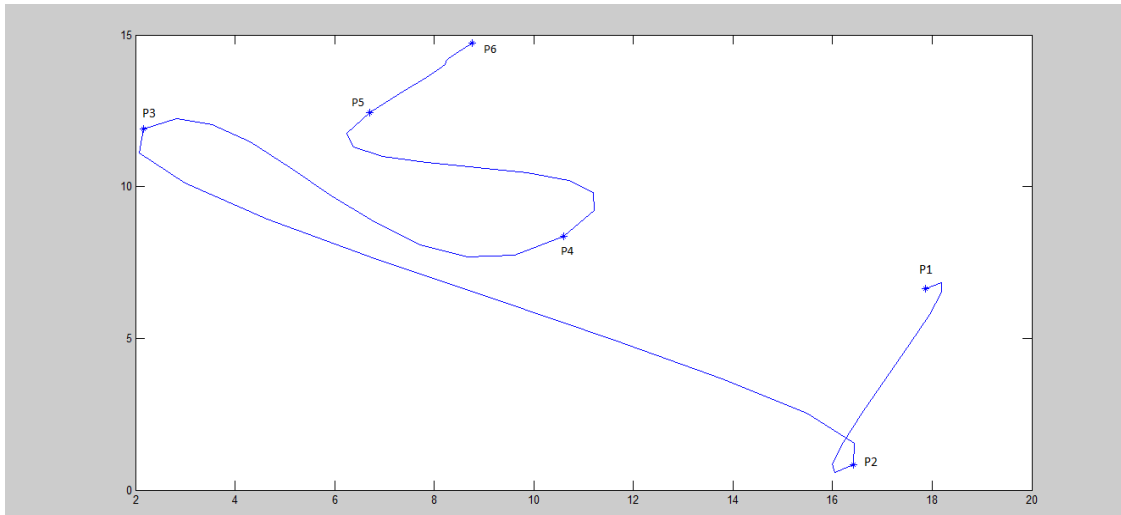


Figura 3 trayectoria acelerada con paso de 0.1

- En la figura 4 se va a mostrar la trayectoria con un paso de 0.02

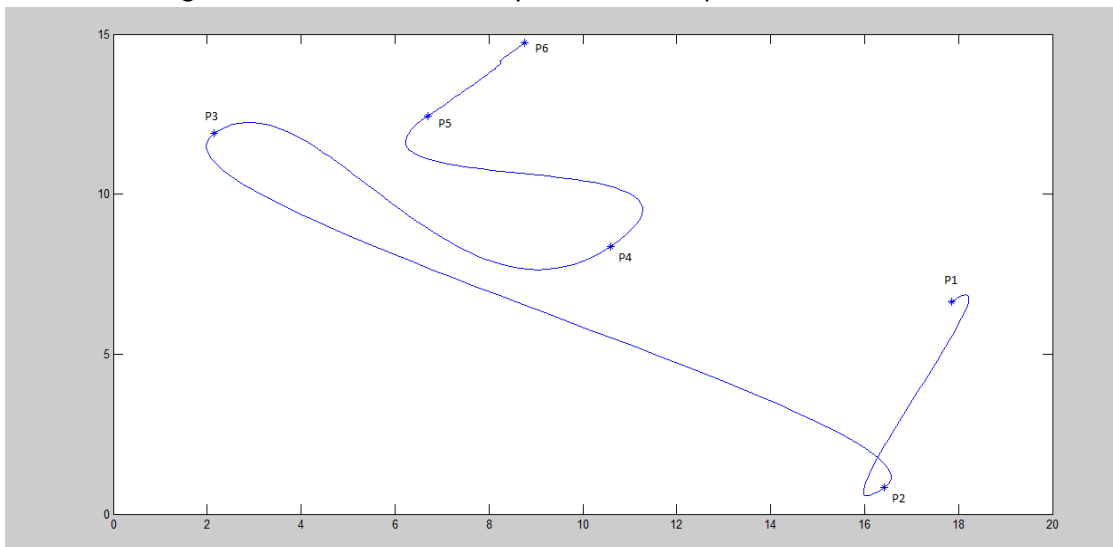


Figura 4 trayectoria acelerada con paso de 0.02

En las figuras anteriores se puede observar que conforme se va disminuyendo el paso de la trayectoria esta se vuelve más natural, además de irse reduciendo los cambios repentinos de dirección, esto se debe a que cuanto más pequeño es el paso, el programa calcula un mayor número de posiciones entre cada par de puntos de control lo que hace que el sistema tenga un mayor número de posiciones para así poder generar la trayectoria con una mayor precisión. Aunque llegara un determinado momento en el que ya no es recomendable reducir más el paso debido a que no modificara significativamente la trayectoria. Además cada vez que se reduce el paso se aumenta el número de operaciones que nuestro programa debe resolver aumentando así el tiempo de cálculo, este tiempo no es significativo en nuestro programa actual pero si en programas más complejos resueltos posteriormente.

## 3 MOVIMIENTO DE UN SÓLIDO EN $\mathbb{R}^3$

---

### 3.1 INTRODUCCIÓN

La realización de este apartado de movimiento de un sólido en  $\mathbb{R}^3$ , se ha basado en el trabajo desarrollado por [Murray et al, 1994] y [Guerrero-2013]

A la hora de parametrizar la trayectoria de un vehículo debemos de tener en cuenta más factores que los que se han empleado para la realización del anterior apartado.

En los casos anteriores solo se ha hablado de la posición del vehículo en un sistema 2-D, a partir de ahora se realizara sobre un entorno de trabajo en 3-D, que nos añadirá una variable de coordenadas adicional. A parte de ese cambio, se va a introducir un concepto nuevo, que es la rotación de dicho vehículo mientras recorre la trayectoria definida, es decir los giros que el vehículo realiza.

Este concepto de rotación es un poco complejo por lo que en este apartado nos dedicaremos a desarrollarlo bien y definirlo correctamente para su posterior uso en algunos ejemplos de simulación. Esta rotación viene dada por una matriz de rotación que para un espacio 3D será de 3x3.

Para armonizar ambos movimientos de rotación y traslación en una sola matriz se utilizan las matrices de transformación las cuales nos indican la rotación y traslación que sufre el vehículo al desplazarse de un punto origen A a otro punto B, estas matrices son de 4x4, se profundizara más en ellas conforme avance el capítulo.

### 3.2 MATRICES DE ROTACIÓN

Las matrices de rotación nos indican el giro que se ha producido en el vehículo de una posición inicial a otra posición final, es importante definir que estas matrices no provocan traslación en el vehículo, pero sí que se producen un cambio en las coordenadas del vehículo debido a la rotación del mismo, aunque si el giro se produce sobre un eje de referencia, en dicho eje no se producirá un cambio de coordenadas.

#### 3.2.1 Matrices de rotación en el espacio SO

La matriz R es una matriz de componentes que pertenecen al espacio real de 3x3, cuyas columnas  $r_1$ ,  $r_2$  y  $r_3$  son mutuamente orthonormal, esto quiere decir que debe de cumplir la ecuación 3.1.



$$r_i^T * r_j = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases} \quad (3.1)$$

La ecuación 3.1 también puede ser escrita como la ecuación 3.2 (siendo I la matriz identidad de R) de la cual se deduce la ecuación 3.3 que son la propiedad básica de toda matriz de rotación debe cumplir.

$$RR^T = R^T R = I \quad (3.2)$$

$$\det R = \pm 1 \quad (3.3)$$

El conjunto de matrices de 3x3 que cumplen las propiedades descritas en las ecuaciones 3.1, 3.2 y 3.3 están denominadas como SO (3). La abreviatura SO significa special orthonormal. Estas matrices pertenecientes al grupo SO (3) satisfarán los siguientes axiomas de cálculo:

a) cerradura: si  $R_1$  y  $R_2$  pertenecen a SO(3) se cumple:

$$\det(R_1 R_2) = \det(R_1) * \det(R_2) = 1$$

b) Identidad: la matriz identidad es la identidad de cada elemento.

c) Inversa: la inversa R es igual a la matriz traspuesta de R.

d) Asociación: sigue las propiedades de asociación de la multiplicación de matrices es decir  $R_1 * (R_2 * R_3) = (R_1 * R_2) * R_3$

Como se ha descrito al principio del apartado la matriz de rotación se utiliza para definir un giro en grados realizado en una dirección, para aplicar el cambio de coordenadas debido al giro se aplica la ecuación 3.4, donde se conoce el vector de posición B y se quiere hallar la posición A, sabiendo que  $R_{ab}$  nos describe el giro que se produce.

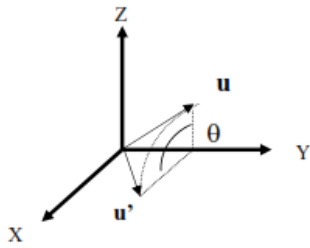
$$q_a = R_{ab} * q_b \quad (3.4)$$

Las matrices de rotación pueden ser combinadas entre sí, para así formar nuevas matrices que pueden resultar más útiles. Por ejemplo se conoce  $R_{ab}$  y  $R_{bc}$  se puede obtener mediante la ecuación 3.5 la matriz de rotación  $R_{ac}$ , esta matriz nos resulta muy útil para el cálculo de la posición A sin necesidad de calcular primero la posición B, como se puede observar las matrices pueden ser combinadas según los intereses particulares en cada momento.

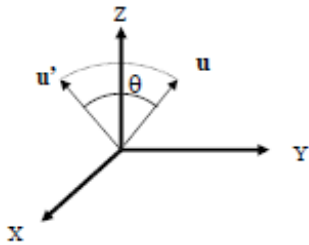
$$R_{ac} = R_{ab} * R_{bc} \quad (3.5)$$

El cálculo de la matriz R depende del ángulo de giro y de la dirección del mismo. El caso más sencillo es cuando el giro se produce en la dirección de un eje de coordenadas, en las figuras 5, 6 y 7 se puede observar cómo quedaría la matriz

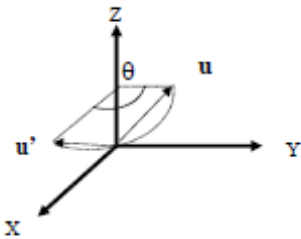
después de realizar un giro de  $\theta$  grados, en la dirección de los diferentes ejes de coordenadas.



$$R(y, \theta) = \begin{pmatrix} \cos\theta & 0 & \text{sen}\theta \\ 0 & 1 & 0 \\ -\text{sen}\theta & 0 & \cos\theta \end{pmatrix} \text{ Figura 5}$$



$$R(x, \theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\text{sen}\theta \\ 0 & \text{sen}\theta & \cos\theta \end{pmatrix} \text{ Figura 6}$$



$$R(z, \theta) = \begin{pmatrix} \cos\theta & -\text{sen}\theta & 0 \\ \text{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ Figura 7}$$

La obtención de la matriz R con un giro aplicado en un eje de coordenadas es proceso sencillo, pero si la dirección de giro no coincide con ningún eje de coordenadas la obtención de la matriz R se complica. Para obtener dicha matriz se tiene que aplicar la ecuación 3.6, en donde el vector r, es el eje de giro. Para que la fórmula no sea demasiado extensa se ha aplicado las siguientes equivalencias:

$$V\theta = 1 - \cos\theta, \quad C\theta = \cos\theta, \quad S\theta = \text{sen}\theta$$

$$R(r, \theta) = \begin{bmatrix} r_x^2 V\theta + C\theta & r_x r_y V\theta - r_z S\theta & r_x r_z V\theta + r_y S\theta \\ r_x r_y V\theta + r_z S\theta & r_y^2 V\theta + C\theta & r_y r_z V\theta - r_x S\theta \\ r_x r_z V\theta - r_y S\theta & r_y r_z V\theta + r_x S\theta & r_z^2 V\theta + C\theta \end{bmatrix} \text{ (3.6)}$$

### 3.2.2 Coordenadas exponenciales

Como se ha podido ver en el apartado anterior, el cálculo de la matriz de rotación con un eje de giro que no coincida con un eje de coordenadas es un proceso muy laborioso por lo que en estos casos usamos las coordenadas exponenciales, que reducen el proceso de cálculo de la matriz R.

Antes de todo es importante introducir el concepto de matriz antisimétrica. Dicha matriz surge de la necesidad de representar de una manera visual un producto vectorial. Para la explicación de dicha matriz primero es necesario saber realizar un producto vectorial, el cual se resuelve de la siguiente manera:

$$a \times b = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$

Esta misma operación puede ser representada mediante la multiplicación de una matriz por un vector, a esta matriz se le denomina matriz antisimétrica que se obtiene mediante la ecuación 3.7. Por lo que tenemos que  $a \times b = (a)^\wedge b$ , siendo  $(a)^\wedge$  la matriz antisimétrica aunque normalmente es denotada como  $\hat{a}$ .

$$\hat{a} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (3.7)$$

Se puede observar que las matrices antisimétricas siempre cumplen que  $-\hat{a} = \hat{a}^T$ , por lo que son contenidas en el espacio so (3), el 3 se debe a que durante la realización de todo este proyecto se va a trabajar con matrices de 3x3.

En las coordenadas exponenciales se trabaja con dos variables:

- La primera es el vector de dirección del giro el cual se va a denotar como w y será el vector con el que se forma la matriz antisimétrica
- La segunda es el ángulo de giro que denotaremos como  $\theta$  el cual se mide en grados.

Estamos trabajando en coordenadas exponenciales por lo que para obtener la matriz de rotación del modelo se debe resolver la ecuación 3.8.

$$R(w, \theta) = e^{\hat{w}\theta} \quad (3.8)$$

Si se cumple que  $||\hat{w}||=1$  y que  $\theta$  es un numero real el exponencial de la ecuación 3.8 puede resolverse mediante un desarrollo en serie de Taylor ( 3.9)

$$e^{\hat{w}\theta} = I + \hat{w}\theta + \frac{\theta^2}{2!}\hat{w}^2 + \frac{\theta^3}{3!}\hat{w}^3 + \dots \quad (3.9)$$

Como se puede apreciar la ecuación 3.9 es una serie de infinitos términos, por lo que no es interesante desarrollarla desde el punto de vista operacional, debido a que para conseguir una precisión admisible se debe realizar un gran número de operaciones. Para obtener un resultado cerrado de la exponencial sin necesidad de realizar la ecuación 3.9 se utiliza la fórmula de Rodrigues' (ecuación 3.10), la cual es la que usualmente se utiliza para realizar la exponencial de la matriz antisimétrica por el ángulo  $\theta$ .

$$e^{\hat{w}\theta} = I + \hat{w}\sin\theta + \hat{w}^2(1 - \cos\theta) \quad (3.10)$$

Como se puede observar si se junta la ecuación 3.8 con la 3.10 se obtiene la solución para obtener la matriz de rotación en el espacio SO (3), por lo que nos quedara la siguiente ecuación:

$$R(w, \theta) = I + \hat{w}\sin\theta + \hat{w}^2(1 - \cos\theta) \quad (3.11)$$

En conclusión cuando el vector dirección de giro no coincida con un eje de coordenadas sino que tenga un vector de dirección, no se aplicara la ecuación 3.6 sino que con ese vector dirección crearemos la matriz antisimétrica  $\hat{w}$  la cual se multiplicara por el ángulo de giro y se realizara la exponencial de la multiplicación, para así obtener la misma matriz de rotación pero de una manera más sencilla.

### 3.2.3 Obtención coordenadas exponenciales a partir de la matriz de rotación en SO(3)

En el apartado anterior se ha descrito el proceso para obtener la matriz de rotación a partir de las coordenadas exponenciales, en este apartado se va a describir el proceso inverso.

Lo primero que se debe realizar es realizar la traza de la matriz de rotación (ec 3.11), la traza es la suma de los elementos de la diagonal principal de una matriz sobre la que es aplicada.

$$traza(R) = (r_{11} + r_{22} + r_{33}) \quad (3.11)$$

Una vez se conoce la traza se calcula el ángulo de giro  $\theta$  mediante la ecuación 3.12, para que esta ecuación tenga solución la traza debe estar acotada de la siguiente manera,  $-1 \leq traza(R) \leq 3$

$$\theta = \cos^{-1}\left(\frac{traza(R)-1}{2}\right) \quad (3.12)$$

Una vez se conoce el ángulo de giro se procede al cálculo del vector de dirección  $w$  mediante la ecuación 3.13, para poder realizar el cálculo de esta operación se tiene que cumplir que  $\theta \neq 0$

$$w = \frac{1}{2 \sin \theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} \quad (3.13)$$

Como se puede ver mediante las ecuaciones 3.12 y 3.13 se puede obtener las coordenadas exponenciales a partir de una matriz de rotación. Este paso también es importante porque cuando se realiza la simulación de una trayectoria, no solo se necesita la transformación de coordenadas exponenciales a la matriz de rotación sino que también se necesita el proceso inverso. Por ejemplo si se quiere conocer el giro en grados que provoca una cierta matriz de rotación.

### 3.2.4 Conclusión

En este apartado se ha introducido el concepto de matriz de rotación, un concepto muy importante a la hora de modelar una trayectoria. En la figura 8 se puede apreciar de una forma visual lo descrito a lo largo de este apartado, así como los cambios de grupos espaciales que se han producido a lo largo del desarrollo de este apartado.

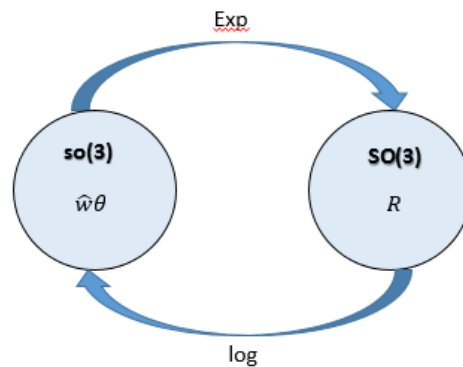


Figura 8

## 3.3 MATRICES DE TRASLACIÓN

En el apartado anterior se ha introducido el concepto de matriz de rotación, esta matriz describe el giro que se produce entre dos posiciones, pero esto ya no es suficiente para modelar la trayectoria de un vehículo, debido a que además de un giro se produce un movimiento en el espacio de dicho vehículo, es decir se produce una traslación en sus coordenadas. Para trabajar conjuntamente con un giro y una traslación se trabaja con lo que se denomina matriz de traslación que va a ser descrita en este apartado.

### 3.3.1 Representación homogénea

En las matrices de traslación se cuenta con dos elementos claramente diferenciables una es la matriz de rotación (R) y otra es el vector de posición (t) que nos indica la traslación que ha sufrido el sólido desde una posición inicial a otra final.

Tal como se ha definido en el apartado anterior la matriz de rotación pertenece al espacio SO(3), mientras que el vector de posición estará formado por un grupo de tres números reales. Si agrupamos estos dos términos en una única matriz de traslación, dicha matriz pertenecerá al grupo SE(3) que significa special euclidean group.

Una vez se conoce el vector de traslación y la matriz de rotación, se puede hallar el punto final de una trayectoria definida por los anteriores parámetros mediante la ecuación 3.14, aunque para simplificar esta operación se utiliza la representación homogénea.

$$q_a = t_{ab} + R_{ab}q_b \quad (3.14)$$

La representación homogénea de la posición de un punto P en el espacio viene denotado por el vector p (ecuación 3.15), en donde  $\lambda$  es el factor de escala. El valor de este factor de escala  $\lambda$  dependen de las condiciones del sólido, para dilataciones se trabajara con un valor mayor que 1, mientras que para contracciones del sólido este valor será menor que 1, aunque a lo largo de este trabajo se tomara  $\lambda = 1$ .

$$\bar{p} = \begin{bmatrix} \lambda p_x \\ \lambda p_y \\ \lambda p_z \\ \lambda \end{bmatrix} \quad (3.15)$$

Mediante la anterior notación homogénea de los puntos la representación de la ecuación 3.14 viene definida de la siguiente manera:

$$\begin{bmatrix} q_a \\ 1 \end{bmatrix} = T_{ab} \begin{bmatrix} q_b \\ 1 \end{bmatrix}$$

En donde  $q_a$  y  $q_b$  son las posiciones del sólido y  $T_{ab}$  es la matriz de traslación del sólido, la cual nos describe cómo evoluciona la trayectoria de un punto inicial b a otro final a, dicha matriz tiene la siguiente representación:

$$T_{ab} = \begin{bmatrix} R_{ab} & p_{ab} \\ 0 & 1 \end{bmatrix}$$

Donde  $R_{ab}$  es la matriz de rotación de punto inicial b a otro final a y como se ha definido en capítulos anteriores sus dimensiones son de 3x3,  $p_{ab}$  representa la traslación que se produce del punto b al punto a y posee unas dimensiones de 3x1.

Se puede apreciar que en las dos anteriores formulas aparecen unos esto se debe a la transformación a representación homogénea de la matriz.

El precio que se paga por tener una representación homogénea del movimiento de un sólido, es que se incrementa la dimensión de los factores a trabajar de 3 a 4, generando un mayor coste computacional.

Una vez ha quedado definida la matriz de traslación es necesario hablar de las propiedades algunas de las propiedades que ella posee, son las siguientes:

- $T_{wa} = (T_{aw})^{-1}$
- $T_{ab} = T_{aw}T_{wb}$
- Propiedad asociativa:  $T_1 * (T_2 * T_3) = (T_1 * T_2) * T_3$

Estos son algunos ejemplos de propiedades de la matriz de traslación, además de estas existen más, pero se ha decidido centrarse únicamente en estas ya que son las que se van a emplear posteriormente para el cálculo de la trayectoria simulada que se va a realizar.

### 3.3.2 Coordenadas exponenciales.

El concepto introducido en la sección 3.2.2 puede ser extendido del grupo SO (3) al grupo SE (3) con el cual se está trabajando en las matrices de traslación.

Estas coordenadas eran empleadas debido a la complejidad de hallar una matriz de rotación cuyo eje de giro no coincidía con un eje de coordenadas. Para la generación de la matriz de traslación en coordenadas exponenciales se necesitan dos variables: una es la matriz antisimétrica del vector dirección de un giro ( $\hat{w}$ ) y la segunda es el vector  $v$  que nos indica la velocidad de traslación.

Dicha matriz de traslación en coordenadas homogéneas poseerá la forma que se describe en la ecuación 3.16, la cual poseerá una dimensión de 4x4.

$$\hat{\xi} = \begin{bmatrix} \hat{w} & v \\ 0 & 0 \end{bmatrix} \quad (3.16)$$

La matriz antisimétrica pertenece al espacio  $so(3)$  mientras que el vector  $v$  es un conjunto de tres números reales, por lo que la matriz conjunta de los dos ( $\hat{\xi}$ ) pertenecerá al grupo  $se(3)$ .

La ecuación 3.16 nos muestra nuestra matriz de traslación en coordenadas exponenciales perteneciente al  $se(3)$ , si se desea pasar esta matriz al grupo SE(3) se debe multiplicar por el ángulo de giro  $\theta$  y realizar su exponencial es decir:

$$T = e^{\hat{\xi}\theta} \in SE(3)$$

Para cambiar del mapa exponencial  $se(3)$  a SE(3) pueden darse dos casos:

- Caso 1:  $w = 0$ , si sustituimos  $\hat{w} = 0$  en la matriz  $\hat{\xi}$  y realizamos su exponencial mediante el desarrollo en serie de Taylor observamos que dicha matriz elevada a un número es siempre igual 0, por lo que el desarrollo en serie nos queda:  $\exp(\hat{\xi} \theta) = I + \hat{\xi} \theta$ , lo que define la matriz de traslación en el espacio SE(3) tal como describe la ecuación 3.17.

$$T = e^{\hat{\xi} \theta} = \begin{bmatrix} I & v\theta \\ 0 & 1 \end{bmatrix} \quad (3.17)$$

- Caso 2:  $w \neq 0$ , por lo que se asume que  $\|w\| = 1$ , condición necesaria para poder aplicar las operaciones realizadas en el capítulo 3.2.2 donde se ha explicado cómo se realiza la exponencial de la matriz antisimétrica  $\hat{w}\theta$  mediante la ecuación de Rodrigues' (3.11), una vez se sabe realizar esta exponencial, la matriz de traslación en el espacio SE(3) quedara de la siguiente manera:

$$T = e^{\hat{\xi} \theta} = \begin{bmatrix} e^{\hat{w}\theta} & (I - e^{\hat{w}\theta})(\hat{w}v) + ww^T v\theta \\ 0 & 1 \end{bmatrix} \quad (3.18)$$

Por lo que se puede definir una operación  $Q(\theta)$  que nos relacione los vectores de posición  $t$  y  $v$ , dicho operador tendrá la siguiente forma:

$$Q(\theta) = (I - e^{\hat{w}\theta})(\hat{w}) + ww^T \theta \quad (3.19)$$

Por lo que utilizando este operador definido en 3.19 el cambio de espacio nos queda de la siguiente manera para los vectores de posición:

$$t = Q(\theta)v$$

Una vez han sido definidas las ecuaciones de cálculo de la exponencial ya estamos en posición de poder realizar dicha operación, es importante recordar bien el concepto de matriz exponencial y las operaciones que ello conlleva debido a que posteriormente en los ejemplo de trayectoria que se van a simular dicho concepto estará siempre presente en las ecuaciones de cálculo.

### 3.3.3 Obtención de coordenadas exponenciales a partir de la matriz de traslación en SE(3)

En determinadas situaciones de cálculo de trayectorias se necesita realizar la operación inversa que la descrita en el apartado anterior, es decir el logaritmo neperiano de la matriz de traslación  $T$ , para así obtener la matriz  $\hat{\xi}$ .

Como en el anterior caso primero se procede a realizar el logaritmo de la función de rotación  $R$  y después del vector de traslación  $t$ .

Como se ha descrito en el apartado 3.2.3, el ángulo de giro  $\theta$  se calcula realizando la operación 3.12 y el vector de dirección  $w$  se calcula mediante la fórmula 3.13.



Para calcular el vector  $v$  a partir del vector de posición  $t$  se tiene que realizar la operación inversa realizada en el anterior apartado es decir:

$$v = Q(\theta)^{-1}t$$

$Q(\theta)$  está definido en la ecuación 3.19 por lo que solo será necesario realizar la inversa de dicho operador, no se va a entrar en detalle de cómo se realiza esta operación inversa ya que el programa de cálculo utilizado lo soluciona automáticamente.

### 3.3.4 Conclusión

En este apartado se ha desarrollado el concepto de matriz de traslación el camino para su obtención y los cambios de espacio que se producen en ella, pueden ser un poco complejos, en la figura 9 se puede apreciar de una manera visual, los diferentes espacios de trabajo y como se transforma dicha matriz al pasar de uno a otro. Es de relevante importancia la comprensión de dichas matrices y sus respectivos espacios de trabajo debido a que a partir de ahora dichos conceptos van a ser usados para el modelado de trayectorias, por lo que es de especial importancia el conocimiento y entendimiento de la figura 9.

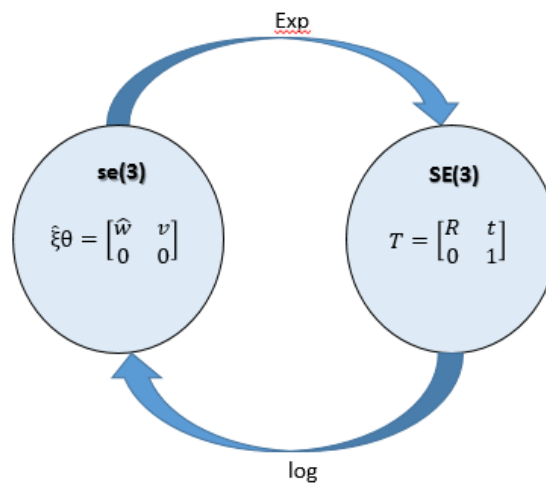


Figura 9

## 4 REPRESENTACIÓN DE UNA TRAYECTORIA MEDIANTE SPLINES

---

### 4.1 INTRODUCCIÓN

En el apartado anterior se han descrito las matrices de rotación y traslación para el movimiento de un sólido, en este capítulo se va a describir una trayectoria en un vehículo en 3D mediante splines, usando los conceptos del anterior apartado para modelar los movimientos que se producen.

Antes de todo se debe introducir el concepto de spline, hace mucho tiempo en dibujo un spline era una banda flexible que se utilizaba para hacer pasar una curva por un conjunto de puntos, el concepto no ha variado mucho a lo largo de los años,

Actualmente el objetivo es expresar esa curva de forma matemática mediante una función polinómica cúbica cuya primera y segunda derivadas sean continuas a lo largo de la curva.

Por lo tanto el termino de curva de spline se refiere a cualquier curva compuesta con secciones polinómicas que satisfacen ciertas condiciones de continuidad en cada frontera del intervalo. Existen diferentes especificaciones de splines que difieren en el polinomio y en las condiciones de frontera que requiere.

Especificamos una determinada curva spline al proporcionarle un conjunto de puntos a los que denominamos puntos de control. Según [González] estos puntos de control son ajustados después por funciones polinómicas continuas de una de las siguientes formas:

1. La curva realiza una interpolación de los puntos de control, adaptándose a las secciones polinómicas de modo que la curva pase través de cada punto de control.
2. La curva realiza una aproximación de los puntos de control, los polinomios se ajustan a la trayectoria definida por los puntos de control pero sin necesidad de pasar a través de ningún punto de control.

Las diferentes formas de representación que se han descrito de la curva generada mediante splines, pueden apreciarse de una forma gráfica en la figura 10. Las diferentes trayectorias que se han generado para la realización de este apartado van a utilizar las curvas que realizan una aproximación de los diferentes puntos de control, como se ha explicado anteriormente esto es debido a la naturaleza de los polinomios elegidos para representar dichas curvas.

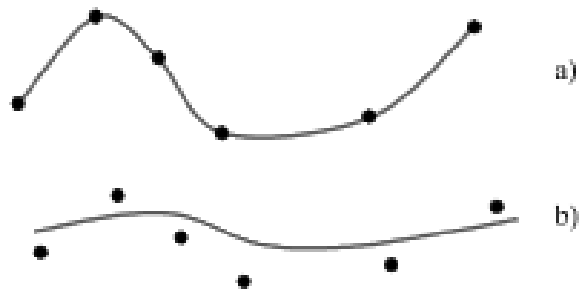


Figura 10 a) interpolación b) aproximación

A lo largo de este apartado se va a trabajar con B-Splines cúbicos debido a que nos proporcionan una buena representación para trayectorias en  $\mathbb{R}^3$ , aunque este tipo de splines tienen una desventaja y es que no son fáciles de aplicar cuando trabajamos con Rotaciones en 3D, así como su interpolación en el espacio de trabajo  $SO(3)$ .

Las ecuaciones de movimiento mediante B-splines cúbicos, que se va a mostrar tanto en el apartado 4.2 como el 4.3 han sido obtenidas por [Lovengrove et al, 2013]

## 4.2 B-SPLINES CÚBICOS SIN MATRIZ DE ROTACIÓN

En este apartado se va a simular una trayectoria que modele el movimiento de un vehículo en un mapa virtual 2D, dicha trayectoria no poseerá componente de rotación, esto quiere decir que en las matrices de traslación descritas en el apartado anterior, el componente de rotación será la matriz identidad.

### 4.2.1 Descripción de la ecuación de movimiento

La función estándar de un B-Spline de grado  $k-1$ , viene dada por la ecuación 4.1, en donde  $p_i$  son los puntos de control en determinados instantes de tiempo, los cuales pertenecen al espacio de los números reales, mientras que  $n$  significa el grado que posee el spline.

$$p(t) = \sum_{i=0}^n p_i B_{i,k}(t) \quad (4.1)$$

Como se ha comentado antes se va a trabajar con B-Splines cúbicos lo que quiere decir que  $k=4$ , por lo que la trayectoria resultante de aplicar estos splines será de grado 3. El grado de una trayectoria generada por splines, nos indica el número de puntos de control que se tienen en cuenta en el polinomio para la generación de  $p(t)$ .

Para la mejor comprensión de los puntos de control que se tienen en cuenta para la generación de  $p(t)$ , se va a realizar un ejemplo basado en la figura 11. Para el cálculo de  $p(t)$  la ecuación de cálculo va a tener en cuenta los siguientes puntos de control: los dos puntos siguientes en el tiempo (  $p3$  y  $P4$ ), el punto  $P2$  el cual es el punto inicial del tramo de interpolación de la trayectoria calculada y en el punto  $P1$  ya que dicha ecuación de cálculo también utiliza como referencia un punto anterior al intervalo calculado.

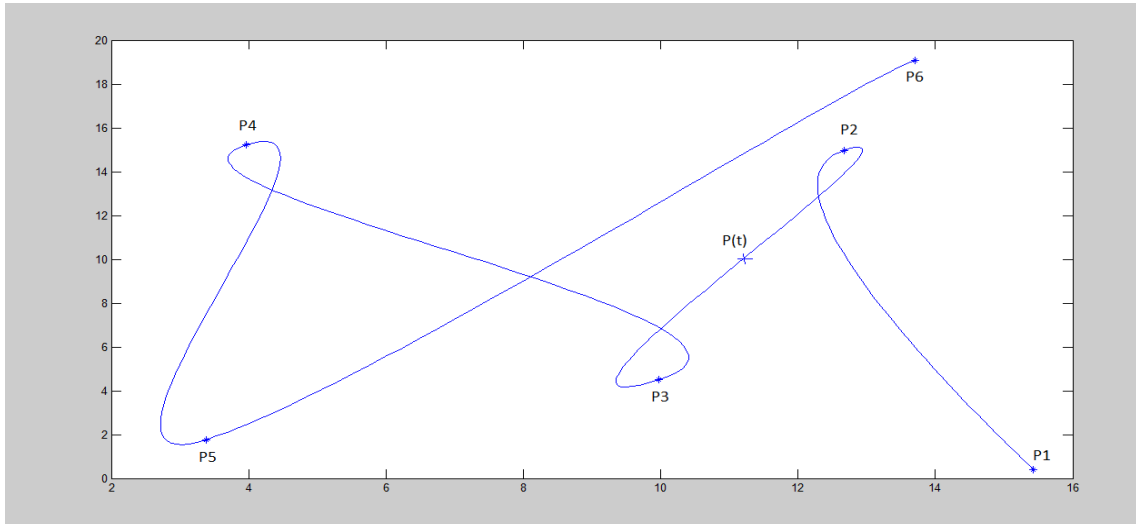


Figura 11

Una vez que se conoce el grado del spline se procede a la obtención de la matriz  $\tilde{B}(t)$  que se ha descrito en 4.1, la cual al realizar la especificación de su grado dependerá de  $\tau$  que es el tiempo normalizado en el instante en el que se está trabajando, anteriormente ya se ha hablado sobre el tiempo normalizado el cual se puede obtener mediante la ecuación 2.4, cabe recordar que el tiempo normalizado se utiliza para definir el tiempo de paso del sólido entre cada par de puntos de control. En la ecuación 4.2 se puede observar la forma que posee la matriz  $\tilde{B}(\tau)$  para un polinomio de grado 3.

$$\tilde{B}(\tau) = C \begin{bmatrix} 1 \\ \tau \\ \tau^2 \\ \tau^3 \end{bmatrix} \quad (4.2)$$

Donde la matriz  $C$  es una matriz de dimensiones  $4 \times 4$  que tiene la siguiente forma:

$$C = \frac{1}{6} \begin{bmatrix} 6 & 0 & 0 & 0 \\ 5 & 3 & -3 & 1 \\ 1 & 3 & 3 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Una vez que se conoce el grado del spline y se ha definido la matriz 4.2 que nos provoca un cambio de la variable tiempo ( $t$ ), se procede a la obtención de la ecuación de movimiento del sistema (4.3) la cual depende del tiempo normalizado  $\tau$  y que se deduce de la ecuación 4.1.

$$p(\tau) = p_0 \tilde{B}_0(\tau) + \sum_{i=1}^3 (p_i - p_{i-1}) \tilde{B}_i(\tau) \quad (4.3)$$

En donde el subíndice de la matriz  $\hat{B}$  representa la fila seleccionada para el cálculo de dicha matriz siendo 0 la primera fila de la matriz y 3 la 4 fila de dicha matriz. Los puntos de control utilizados para el cálculo de la trayectoria siguen la misma regla que los descritos en el ejemplo de la figura 11, donde  $p_0$  será el punto anterior al punto de control del inicio del intervalo, por lo que  $p_1$  será el punto de control del inicio del intervalo a calcular, el resto de puntos calculados seguirán dicha progresión.

La simulación de la trayectoria se ha realizado con el programa de cálculo Matlab por lo que para introducir dicha fórmula en Matlab debemos realizar un ajuste de los subíndices para la correcta interpretación de dicho programa debido a que Matlab no interpreta los subíndices 0, siendo  $i$  el número que indica que punto de control inicia el intervalo calculado. Por lo que la ecuación de movimiento quedará de la siguiente manera:

$$p(\tau) = p_{i-1} + \sum_{k=1}^3 (p_{i-1+k} - p_{i+k-2}) \tilde{B}_{k+1}(\tau)$$

Como se puede observar se ha eliminado de la ecuación el primer término de la matriz  $\tilde{B}$  esto se debe a que es igual a 1 y no nos produce ninguna modificación de la ecuación.

La ecuación 4.3 es aplicable a todos los sistemas en los que no se produzca una rotación del sólido, ya que como se ha comentado anteriormente este tipo de spline tiene muy buena representación para trayectorias en el espacio de los reales, pero para calcular una trayectoria que posea rotación el proceso de obtención de la ecuación característica de dicha trayectoria se complica.

#### 4.2.2 Condiciones especiales

Una vez se conoce y se ha adaptado la ecuación de cálculo de la posición se procede a la generación del programa de cálculo de la trayectoria, el cual se puede apreciar en el anexo A. En dicho programa se ha tenido que añadir algunas condiciones de cálculo para el correcto funcionamiento del mismo:

- Uno de los problemas que se han tenido a la hora de realizar el cálculo, era como hacer saber al programa cuando no existe ningún punto anterior al punto de control del intervalo o cuando el punto siguiente al intervalo no existe y es el fin de la trayectoria. Para solventar este problema se ha usado una orden condición "if" que nos indicaba si el valor que la ecuación de cálculo

demandaba existía o no, y si no existía se cogía el punto de control más próximo en su lugar.

- El otro problema a tener en cuenta, es que el polinomio realiza una aproximación de los puntos de control, es decir dicha curva generada por el B-spline no pasa por los puntos de control. Para la realización de nuestro caso nos es indiferente que no pase por los puntos de control peor sí que nos interesa que dicha trayectoria empiece en el primer punto de control y acabe en el último punto de control.

Para la resolución de este problema se ha decidido añadir dos puntos de control adicionales a nuestra matriz de puntos de control, los cuales estarán situados uno al principio y otro al final de dicha matriz. El método de obtención de estos puntos cambia de uno a otro, para la obtención del primer punto se a cogido la ecuación 4.3 y se ha aplicado la condición de que el primer punto de la trayectoria sea el primer punto de control que se ha generado de manera aleatoria, es decir que  $p(\tau = 0) = p_1$ , una vez se ha aplicado dicha condición se procede a la simplificación de dicha ecuación quedando de la siguiente forma:

$$p_{-1} = 2p_0 - p_1 \quad (4.4)$$

En la ecuación 4.4 el punto de referencia posee el subíndice 0 el cual es el primer punto de control de la matriz generado de manera aleatoria, por lo que el que posee el subíndice -1 es el punto que está situado delante de el en la matriz de referencia es decir el punto que estábamos buscando. La obtención del último punto de control a añadir en la matriz de puntos de control se realiza de una manera muy similar a la del primero, mediante la ecuación 4.5 que es una reorganización de la ecuación 4.4 descrita con anterioridad.

$$p_1 = 2p_0 - p_{-1} \quad (4.5)$$

Se observa que la fórmula es muy parecida a la anterior, pero en este caso el punto de referencia ya no es el primero punto de control sino que es el último de los que han sido generados aleatoriamente por el programa de cálculo. Por lo que el subíndice 1 indica el punto siguiente al último punto de control generado de manera aleatoria.

Estos dos puntos que han sido calculados no se van a introducir en el proceso de cálculo, es decir no se va a realizar una interpolación entre uno de estos puntos y el punto de control cercano, sino que son usados para que las ecuaciones de posición entre los puntos de control generados aleatoriamente contengan una mayor información, lo que quiere decir que cumplan mejor las necesidades impuestas.

### 4.2.3 Ejemplo

Una vez se han descrito todos los elementos a tener en cuenta en la elaboración de nuestra trayectoria, se va a proceder a la elaboración de un ejemplo de la trayectoria de B-splines cúbicos sin rotación que se ha descrito anteriormente.

La trayectoria resultante de realizar el ejemplo de cálculo puede apreciarse en la figura 12, para la generación de dicha trayectoria, se ha utilizado la función de cálculo definida en el Anexo A, en la cual se han tenido en cuenta los siguientes parámetros como entradas del sistema:

1. El programa de cálculo ha generado 7 puntos de control de manera aleatoria, los cuales varían en un rango entre 0 y 20 metros, tanto en su coordenada x como en su coordenada y.
2. El paso del tiempo normalizado ha sido fijado en 0.1, es decir el tiempo normalizado de la interpolación entre cada par de puntos de control avanza de 0.1 en 0.1 con cada iteración del sistema.
3. El tiempo de paso de cada punto de control, el cual es necesario para el cálculo del tiempo normalizado entre cada par de puntos, se ha definido mediante el siguiente vector:  $t = [0 \ 3 \ 5 \ 6 \ 7 \ 9 \ 11]$

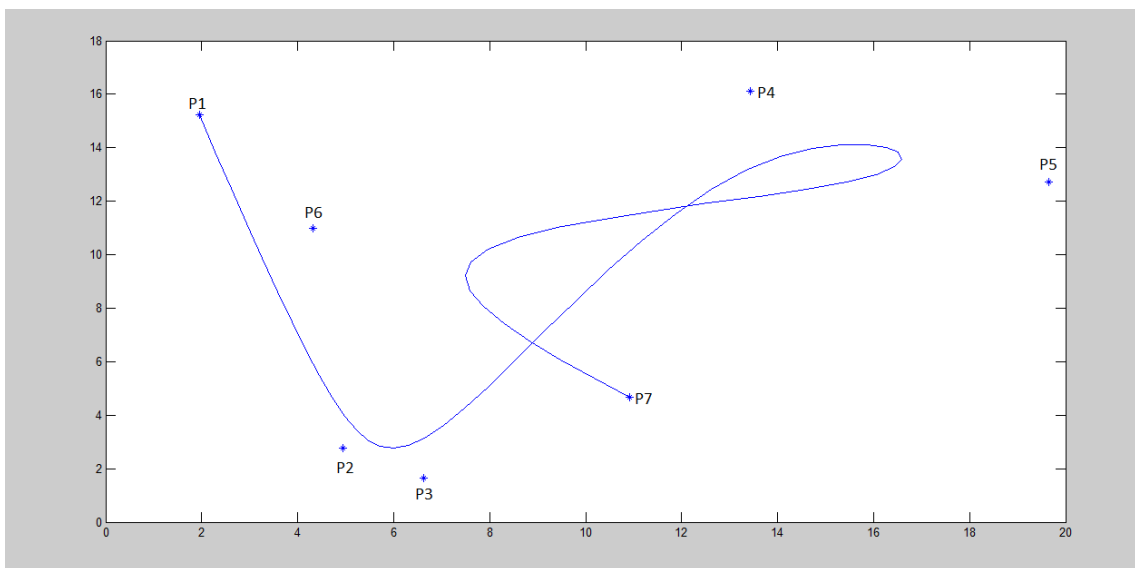


Figura 12 Trayectoria B-splines cúbicos

Como se puede observar en la figura 12 la curva representa una aproximación de los puntos de control, es decir no llega a pasar por los puntos de control aleatorios del sistema, menos en el punto inicial y final. Esto es debido a la condición de que la recta empieza en el punto inicial y acabe en el final, que se ha introducido en el programa de cálculo, dicha condición ha sido descrita con anterioridad. El hecho de que la recta no pase justo por los puntos de control intermedios no es influyente para nuestro caso debido a que dichos puntos intermedios en la realidad van a ser generados por unos sensores del vehículo y podemos adaptarlos sabiendo que forma va a tener nuestra trayectoria.

Se puede observar que dicha figura 12 posee una trayectoria más natural que las descritas en apartados anteriores. Pero aun así no representa bien un movimiento de un vehículo real debido a que no posee una componente de rotación, ya que un todo vehículo puede rotar en el espacio, además se ha simulado en un entorno 2D siendo que en la realidad un vehículo se mueve en 3D. Estos conceptos para dotar a la trayectoria de una mayor naturalidad van a ser realizados en el siguiente apartado.

### 4.3 B-SPLINES CÚBICOS CON MATRIZ DE ROTACIÓN

#### 4.3.1 Descripción de la ecuación de movimiento

En este apartado se va a desarrollar el modelo de curva mediante B-splines cúbicos, a diferencia con el apartado anterior en este la curva generada va a tener un componente de rotación y va a ser simulada en un entorno virtual 3D.

Haciendo uso de los mapas logarítmicos y exponenciales de las matrices de traslación, los cuales han sido descritos en el anterior apartado, se puede reescribir la ecuación 4.3 para representar trayectorias pertenecientes al espacio SE(3). Esto se consigue mediante la sustitución, de las diferencias de los puntos de control por la representación del mapa logarítmico entre dos matrices de traslación de control, además se debe sustituir el sumatorio por el multiplicativo. De momento se va a trabajar en tiempo continuo y no normalizado por lo que cogeremos la ecuación 4.3 sustituyendo la variable  $\tau$  por  $t$ . Una vez realizado todos estos pasos la matriz de cambio de coordenadas entre la referencia mundo y el punto calculado, queda de la siguiente forma:

$$T_{w,s}(t) = \exp(\tilde{B}_{0,k}(t) \log(T_{w,0})) \prod_{i=1}^n \exp(\tilde{B}_{i,k}(t) \Omega_i) \quad (4.6)$$

En donde el mapa logarítmico entre dos matrices de traslación, que es  $\Omega_i$  se representa de la siguiente manera:

$$\Omega_i = \log(T_{w,i-1}^{-1} T_{w,i}) \quad (4.7)$$

Una vez se conoce la ecuación 4.6 hay que particularizarla para el grado de nuestro caso de trayectoria de B-spline la cual será cúbica, este grado nos define la matriz  $\tilde{B}$  tal como se muestra en la ecuación 4.2. La variable de dicha matriz es el tiempo normalizado por lo que además se deberá realizar un cambio de variable en la ecuación 4.6 quedando de la siguiente manera:

$$T_{w,s}(u) = T_{w,i-1} \prod_{j=1}^3 \exp(\tilde{B}(u)_j \Omega_{i+j}) \quad (4.8)$$



Por lo tanto va a ser la ecuación 4.8 la que se use para el cálculo de la posición y orientación del sólido a lo largo de la trayectoria, dicha trayectoria está formulada a partir de la trayectoria indicada en el apartado anterior, por lo que poseerá la misma característica de aproximación de los puntos de control, es decir la curva no pasara por los puntos de control sino que realizara una aproximación a ellos.

#### 4.3.2 Matrices de control

En este apartado ya no se va a trabajar con puntos de control sino con matrices de traslación de control, ya que el sólido en cada instante de tiempo va a poseer una determinada orientación y posición, por lo que es necesario modelar esta rotación. Por lo que lo primero que se debe hacer es la generación de un vector de matrices de control el cual va a estar modelado de forma aleatoria por el ordenador.

La generación de la matriz aleatoria de control no es tan sencillo como en el apartado anterior en el cual solo se daban valores aleatorios a los vectores de posición.

La generación del vector  $w$ , el cual nos indica el vector dirección en el que se produce el giro del sólido, no se puede calcular simplemente dándole valores aleatorios, sino que todos los puntos generados aleatoriamente deben estar uniformemente distribuidos en la superficie de una esfera, esto se ha conseguido muestreando los diferentes puntos 3D con una distribución gaussiana y proyectando dichos puntos en la superficie de la esfera, este proceso se realiza mediante la función `rand_unit_sphere` indicada en el anexo A. Otra manera de simular este vector  $w$ , sería mediante la utilización de las coordenadas esféricas, en las cuales se daría un valor aleatorio entre 0 y  $2\pi$  al ángulo  $\phi$  y entre 0 y  $\pi$  al ángulo  $\theta$ , generándose así un vector incluido dentro de la esfera, este método no se ha utilizado debido a que los puntos tienden a acumularse en los polos, por lo que se produce una distribución poco uniforme a lo largo de la esfera.

Una vez realizado el proceso anterior ya se conocen los vectores de dirección de giro para cada punto de control de nuestro sistema. Además del vector dirección es necesario calcular el vector de traslación ( $t$ ) y el ángulo de giro que se produce en cada punto de control. El vector traslación lo generara el programa de manera aleatoria, como se ha realizado en casos anteriores y tendrá valores comprendidos entre 0 y 25 metros, mientras que el ángulo de giro ( $\theta$ ) de cada punto de control se calculara generando un valor aleatorio entre 0 y 1 y multiplicándolo por  $2\pi$ , por lo que se obtendrá el giro de cada punto de control en radianes.

Ya hemos generado de manera aleatoria  $w$ ,  $t$  y  $\theta$  aunque para la formación de la matriz de traslación en el espacio SE (3), tenemos un problema debido a que  $w$  y  $\theta \in so(3)$ , por lo que deberemos realizar la matriz antisimétrica de  $w$ , multiplicarla por el ángulo de giro  $\theta$  y aplicar la ecuación de Rodrigues' (3.10) a dicha matriz para así realizar su exponencial, quedándonos como resultado de la exponencial la matriz de rotación  $R$  que pertenece al espacio SO(3). Una vez que hemos obtenido  $R$  ya estamos en condiciones de generar las matrices de traslación

de cada punto de control referenciadas al mundo, las cuales se formaran de la siguiente manera:

$$T_{w,i} = \begin{bmatrix} R_{3x3} & t_{3x1} \\ 0_{1x3} & 1 \end{bmatrix}$$

Se va a necesitar almacenar todas las matrices de control en un mismo sistema, esto se va a realizar de la misma manera que se ha agrupado la matriz A en la figura 13, por lo que tendremos una agrupación en 3D de las matrices de control de nuestra trayectoria. Las matrices dentro de la ordenación en 3D van a estar numeradas siendo uno la primera y así sucesivamente, para entender bien como trabaja este conjunto de matrices vamos a realizar un ejemplo, imaginemos que tenemos un conjunto de matrices T1 y deseamos obtener la componente de la cuarta fila la segunda columna de la tercera matriz, lo cual se representaría de la siguiente manera: T1( 4,2,3), como se observa se añade un nuevo número a la numeración de las matrices que nos indica la posición que ocupa dicha matriz en el conjunto 3D.

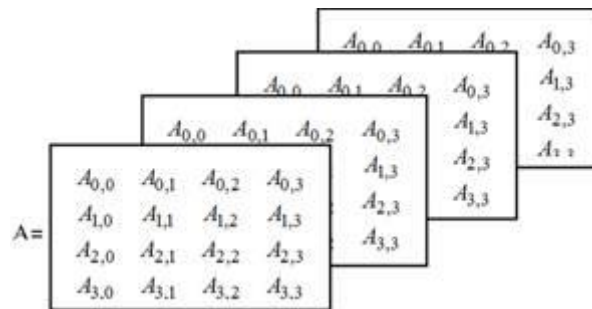


Figura 13 Agrupación de matrices en 3D

En conclusión se va a generar un conjunto de matrices de control situadas en un solo conjunto, las cuales modelaran el movimiento de nuestro vehículo. Estas matrices están referenciadas respecto del mundo, por lo que el vector t de cada una de estas matrices nos va a dar la situación de los puntos de control, cabe destacar que estos puntos tendrán una orientación en el espacio.

#### 4.3.3 Condiciones especiales

A lo largo de este apartado se va a necesitar pasar la matriz de traslación de espacio de cálculo, lo cual se consigue realizando la operación exponencial de dicha matriz. En el capítulo anterior se ha descrito el proceso para el cálculo de dicha operación. El proceso se basa en aplicar la ecuación 3.10 a la matriz antisimétrica del vector dirección de giro multiplicada por el giro, para así obtener la matriz de rotación y para el cálculo del vector de traslación t, multiplicábamos el vector v por el coeficiente Q el cual está definido en la ecuación 3.19.

El proceso de cálculo exponencial no funciona  $\theta$ , sino que para valores de dicho ángulo cercanos a 0 las ecuaciones antes descritas no producen el cambio de espacio correctamente, por lo que hay que realizar un cambio en dichas ecuaciones, las cuales quedaran de la siguiente manera:

1. Ya no se aplicara la ecuación de Rodrigues' para obtener la matriz de rotación, sino que usaremos la siguiente ecuación:

$$R = I + \hat{w} + 0.5\hat{w}^2$$

2. Para la componente de traslación el operador Q, se nos modifica y quedara de la siguiente forma:

$$Q = I + 0.5 \hat{w} + \frac{1}{6} \hat{w}^2$$

Una vez definidas las ecuaciones anteriores ya estamos en disposición de calcular operación exponencial para cualquier condición de ángulo de giro.

Como sucede con la operación exponencial la operación logarítmica, la cual ha sido descrita en el anterior capítulo, también produce fallos en determinadas situaciones de cálculo. Para el correcto funcionamiento de la operación logarítmica de una matriz de traslación se ha implementado un código, el cual queda descrito en el anexo A, en dicho código quedan definidas todas las posibles condiciones de cálculo y los cambios que dichas condiciones producen en las ecuaciones de cambio de espacio. Cabe recordad que la operación logaritmo se usa cuando tenemos una matriz de traslación en el espacio SE(3) y queremos cambiarla al espacio se(3).

#### 4.3.4 Adaptación de los puntos de control

Como se ha indicado en el apartado 4.2 la curva resultante del movimiento realiza una aproximación de los puntos de control, es decir la trayectoria no va a pasar por ningún punto de control, mientras que en los puntos de control intermedios nos es indiferente, sí que se requiere que la trayectoria empiece en el primer punto de control y acabe en el último punto de control.

El método que se va a usar para resolver este problema va a ser muy similar al empleado en el apartado anterior, aunque la obtención de este va a suponer un mayor reto debido a que la ecuación del cálculo de la posición es más compleja. Dicho método consistirá en la obtención de dos nuevas matrices de control que serán situadas al principio y al final del conjunto de matrices de control y cuya finalidad sea únicamente la de ser introducidas en la ecuación posición 4.8 para que las posiciones iniciales y finales calculadas coincidan con la primera y última posición de control generadas de manera aleatoria.

Cabe destacar que estas dos nuevas matrices de control no van a ser usadas para el cálculo de nuevas trayectorias de interpolación entre dos puntos simplemente se utilizan para que las interpolaciones entre los puntos calculados aleatoriamente nos produzcan un resultado satisfactorio a nuestras necesidades.

Como se ha dicho el método de obtención va a ser muy similar al usado en el apartado anterior, por lo que se va a coger la ecuación 4.8 y se le va a imponer como condición que la matriz de traslación en el instante  $t=0$  resultante de la ecuación sea igual que la primera matriz de control que se ha calculado de manera aleatoria. De esta manera fijaremos la matriz resultante y nos centraremos en la obtención de la matriz de control anterior. Una vez se ha impuesto dicha condición hay que proceder a la simplificación de la ecuación, la ecuación resultante de dicha simplificación es la siguiente:

$$R_{w,-1} = R_{w,0} R_{w,1}^{-1} R_{w,0}$$

Se ha trabajado con las matrices de rotación para así simplificar el proceso resolución, pero esta ecuación en forma de matrices de rotación puede extrapolarse a matrices de traslación. Por lo que la ecuación nos quedara la siguiente forma:

$$T_{w,-1} = T_{w,0} T_{w,1}^{-1} T_{w,0} \quad (4.9)$$

La ecuación 4.9 es la que usamos para el cálculo de la primera matriz de control, los subíndices simplemente indican el orden con respecto a la matriz de referencia 0, para el cálculo de dicha matriz, la matriz de referencia será la primera matriz de control generada de manera aleatoria. El cálculo de la última matriz de traslación se realiza de una manera muy similar pero en este caso no calcularemos la matriz con subíndice -1 sino la que posee subíndice 1, debido a que la matriz de referencia será la última matriz de control generada de manera aleatoria, lo cual se obtendrá mediante una sencilla modificación de la ecuación 4.9, quedando la ecuación de la siguiente forma:

$$T_{w,1} = T_{w,0} T_{w,-1}^{-1} T_{w,0}$$

Una vez se ha calculados las dos matrices de control, e introducidas en nuestro algoritmo, las posiciones de inicio y final de la trayectoria ya coincidirán con el primer y último punto de control calculados de manera aleatoria.

#### 4.3.5 Rotación

Esta trayectoria tiene de especial que posee una componente de rotación, para la observación de esa rotación en determinados puntos de la trayectoria se ha añadido un “foco” que nos muestra la orientación que posee nuestro vehículo. Este foco va a tener las características que se aprecian en la figura 14.

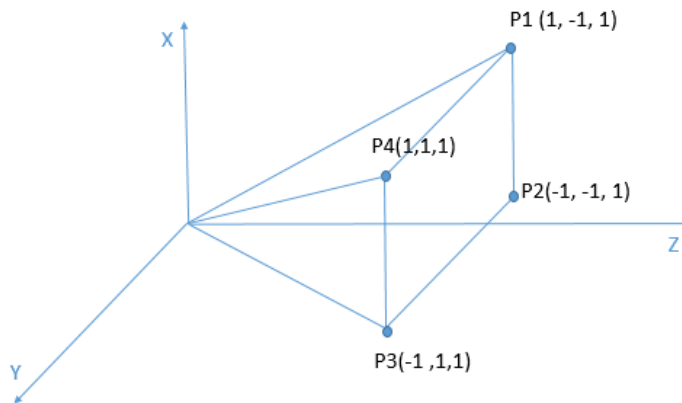


Figura 14

Se puede apreciar en la figura 14 que el foco posee la dirección del eje z, es decir el eje z corta con el punto central del cuadrado, por lo que la parte delantera de nuestro vehículo simulado coincidirá con el eje z propio del vehículo. Cabe destacar que en nuestro programa no va a ser 1 el dato de las coordenadas, sino que se va a definir una constante "scale\_cam" la cual definirá las dimensiones de nuestro foco, la creación de esta constante es un paso muy importante a la hora de implementar la trayectoria ya que nos permitirá adaptar las dimensiones del foco a la de la trayectoria.

Este foco va a ser dibujado una vez terminada cada interpolación entre dos puntos, es decir cada vez que el programa de cálculo en su iteración llegue a la matriz de control y vuelva a empezar su iteración.

El foco está referenciado en el eje z del vehículo pero para la representación de la rotación en la trayectoria este eje tiene que estar referenciado respecto del mundo, esto se consigue realizando la ecuación 4.10, en la cual se utilizan la matriz de traslación de la posición donde quiere ser representada dicha rotación.

$$P_{w1} = R * P_1 + t \quad (4.10)$$

Esta operación tendrá que repetirse para el resto de puntos que componen el foco y que han sido descritos en la figura 14. Una vez se han obtenidos los puntos referenciados al mundo se procede a su representación indicándonos la orientación que posee el vehículo.

En conclusión este es un método muy sencillo y eficaz de poder observar la rotación que se produce en el sólido a lo largo de la trayectoria.

#### 4.3.6 Ejemplo

Este apartado se va a dedicar a la generación de un ejemplo de la trayectoria que se ha sido descrita a lo largo de todo este capítulo. Para el cálculo de la trayectoria se han escogido los siguientes parámetros como datos de entrada del sistema:

- Se consideran 7 puntos de control, anteriormente ya se ha descrito como se generan las matrices de control para cada uno de esos puntos de manera aleatoria.
- El vector de tiempo de paso por cada uno de esos 7 puntos va a ser de la siguiente manera:  $t = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6]$ , se observa que para la traslación de un punto a otro se necesita de 1 segundo para todos los casos.
- La escala de la cámara será de 1 metro, esto es debido a que nuestras coordenadas oscilan entre 0 y 25 metros, por lo que si se escoge un mayor escalado se perdería visión de la propia trayectoria.
- Se ha decidido que el tiempo normalizado este dividido en 20 intervalos, es decir que en cada interpolación de la trayectoria entre dos puntos de control se realizan 20 cálculos de posición.

Una vez definidos los anteriores parámetros ya estamos en condiciones de generar una trayectoria la cual es definida en la figura 15. En dicha figura se pueden observar los “focos” de control de la orientación que se han definido anteriormente.

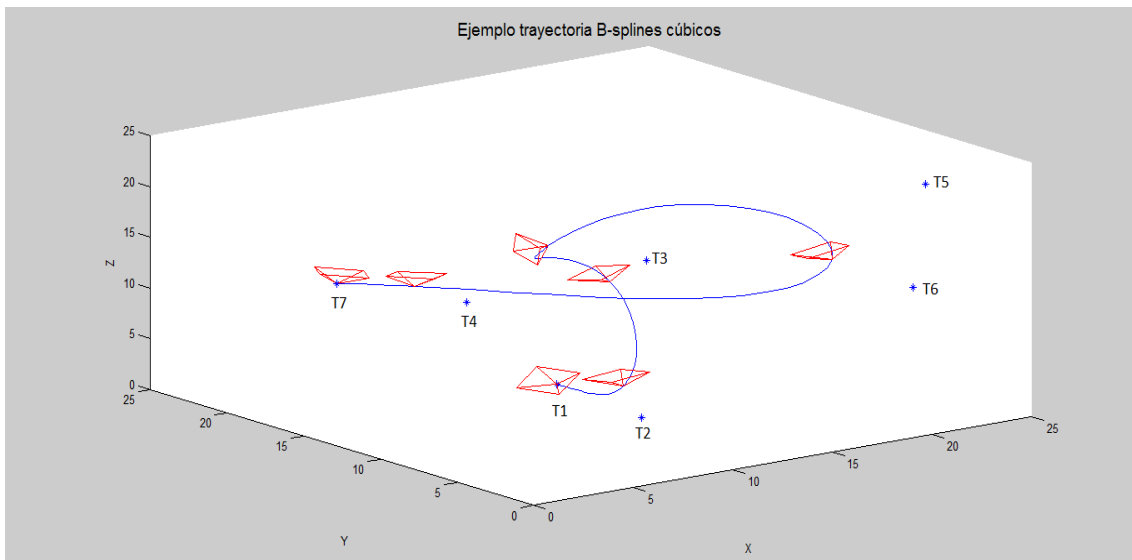


Figura 15 Ejemplo trayectoria B-splines cúbicos

Esta representación nos facilita una buena comprensión de la trayectoria ya que podemos ver la posición del vehículo en el espacio y además la orientación que posee. Al estar en una representación 3D su visualización puede ser un poco confusa por lo que se va a añadir una nueva representación (figura 16) de la misma trayectoria vista desde otra perspectiva.

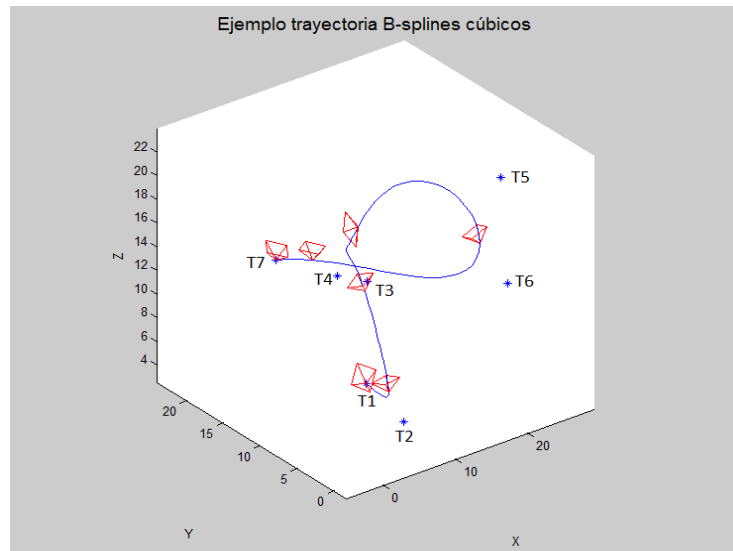


Figura 16 Ejemplo trayectoria B-splines cúbicos

En conclusión esta curva es más realista que las descritas en todos los apartados anteriores, por lo que se va a emplear la trayectoria representada en la figura 15y 16 para modelar los sensores que van a estar presentes en dicho vehículo. Como se ha comentado anteriormente la curva realiza una aproximación de los puntos de control, esta aproximación en algún punto queda un poco distante pero esto se debe a que nuestra trayectoria no solo tiene en cuenta el siguiente punto de control sino que trabaja con varios puntos de control, por lo que esta distancia no es significaba ya que se realiza minimizar el tiempo de traslación.

## 5 SENSORES

---

Los sensores son un elemento muy importante en nuestro entorno de simulación de trayectorias robóticas, ya que de ellos va a dependerla estimación de la trayectoria final. Hasta ahora los puntos han sido generados de una manera aleatoria, pero el objetivo es que nosotros solo fijemos el punto de inicio y final de la trayectoria, mientras que los puntos intermedios sean generados por los sensores.

En este apartado se va a hablar de los sensores que pueden ser considerados en nuestro vehículo. Se ha decidido modelar dichos sensores en el último tipo de trayectoria que ha sido descrito (apartado 4.3), esto es debido a que es la trayectoria que posee una mayor complejidad, además de una mayor facilidad de adaptarla al mundo real.

En resumen en este apartado se va a modelar unos determinados tipos de sensores, capaces de percibir el entorno que rodea al vehículo simulado, el cual recorra una trayectoria generada mediante B-Splines cúbicos con rotación.

## 5.1 SENSOR DE VELOCIDAD

La velocidad que posee un vehículo en todo momento es una variable muy interesante para el cálculo de una trayectoria. Por lo que dicha variable se va a estudiar a lo largo de este apartado.

### 5.1.1 Tipos de sensores de velocidad

Un sensor de velocidad nos da medidas de dicha componente cada determinado espacio de tiempo. Actualmente en el mercado se dispone de una amplia gama de sensores de velocidad, a continuación se va a describir los dos más usados según [velocity-Sensors]:

- tacómetro: es el sensor más utilizado en el mundo de la automoción, este sensor nos produce como salida las revoluciones por minuto que posee una rueda, conociendo la longitud de dicha rueda se obtiene la velocidad del móvil de una manera muy sencilla. Este sensor es muy usado para vehículos que se mueven en 2D pero para vehículos que operan en 3D no se puede implementar ya que dichos vehículos no funcionan con ruedas.
- Laser surface velocity: es un sensor de velocidad de no contacto que usa el principio del efecto Doppler para recordar el efecto Doppler es el cambio de frecuencia de una onda producida por el movimiento de la fuente con respecto a un observador. Este solo es útil para medir una medida no un conjunto de medidas cada intervalo de tiempo seleccionado por lo que no nos será de utilidad para nuestros propósitos.

El cálculo de la velocidad en este apartado va a estar referenciado como se ha dicho anteriormente al tipo de trayectorias del apartado 4.3 (B-splines cúbicos). Al ser un modelo en 3D es difícil modelar un sensor de los que se han descrito anteriormente en este tipo de trayectoria, por lo que se la velocidad se va a obtener a través de su cálculo teórico.

### 5.1.2 Cálculo teórico

Para el cálculo teórico, es necesario realizar la derivada de la posición, pero en nuestro ejemplo de trayectoria la posición viene definida en forma de matriz de traslación por lo que es necesario realizar la derivada de dicha matriz.

La derivada de la matriz de traslación se resuelve según [Lovengrove et al, 2013] a partir de la ecuación 5.1.

$$\dot{T}_{w,s}(\tau) = T_{w,i-1} (\dot{A}_0 A_1 A_2 + A_0 \dot{A}_1 A_2 + A_0 A_1 \dot{A}_2) \quad (5.1)$$

En donde  $A_j$  y  $\dot{A}_j$  quedan de la siguiente manera:



$$A_j = \exp(\Omega_{i+j} \tilde{B}(\tau)_j) \quad (5.2)$$

$$\dot{A}_j = A_j \Omega_{i+j} \dot{\tilde{B}}(\tau)_j \quad (5.3)$$

En la ecuación 4.2 se ha descrito  $\tilde{B}$  para realizar la derivada de dicho vector se realiza la derivada de cada uno de sus componentes, la forma final se puede observar en la ecuación 5.4. En donde  $\Delta t$  es la duración en tiempo del intervalo que se está calculando en la interpolación.

$$\dot{\tilde{B}}(\tau) = \frac{1}{\Delta t} C \begin{bmatrix} 0 \\ 1 \\ 2\tau \\ 3\tau^2 \end{bmatrix} \quad (5.4)$$

Recordar que la matriz C tiene la siguiente forma:

$$C = \frac{1}{6} \begin{bmatrix} 6 & 0 & 0 & 0 \\ 5 & 3 & -3 & 1 \\ 1 & 3 & 3 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Una vez realizada la ecuación 5.1, se obtiene la derivada de la matriz de traslación. Para la obtención de la velocidad tenemos que observar la 4ª columna donde la primera fila será la velocidad en el eje de coordenadas "x", la segunda fila será en el eje "y" y la tercera fila en el eje "z". Por lo que ya conocemos la velocidad del sólido en cada una de sus coordenadas para una determinada posición de cálculo.

### 5.1.3 Ejemplo

En este ejemplo se va a representar la velocidad que posee un vehículo a lo largo de su trayectoria, como se ha indicado anteriormente esta trayectoria va a ser la misma que la que se ha trazado en el ejemplo del apartado 4.3.

La representación de la velocidad se va a representar en la figura 17 la cual está dividida en tres gráficas una para coordenada de la velocidad, con lo que se obtiene la velocidad con la que se mueve el vehículo en cada dirección. En el eje de abscisas se ha representado los diferentes puntos de cálculo del proceso, es decir para cada posición del vehículo calculada por el programa de cálculo se ha calculado también su derivada por lo tanto su velocidad.

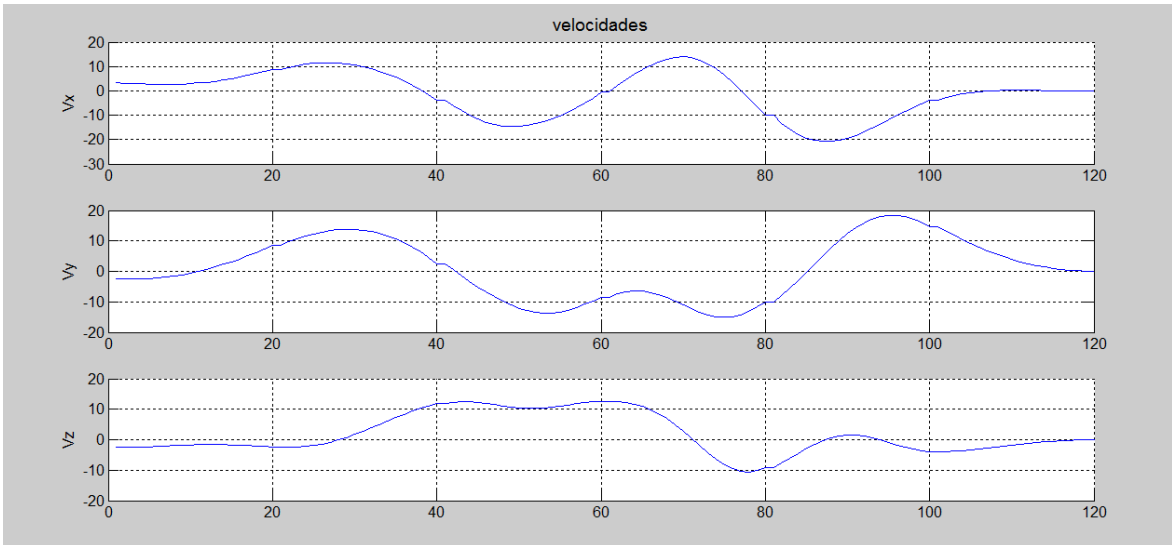


Figura 17 Velocidades en el vehículo

Como la trayectoria es un movimiento acelerado la velocidad no será constante sino que varía a lo largo de la trayectoria, esto se debe a que tiene un segundo para ir de punto a punto y la distancia entre puntos no es constante, por lo que para cumplir estas características tiene que ir variando. El hecho de que la velocidad sea negativa no es relevante simplemente indica que el vehículo se está moviendo en la dirección negativa del eje de coordenadas. La figura 17 nos revela mucha información acerca del movimiento, debido a que podemos conocer cómo evoluciona la velocidad a lo largo de la trayectoria, información muy valiosa para el modelado en la vida real del vehículo de movimiento ya que tienen que poder alcanzar dichas velocidades.

## 5.2 SENSOR DE ACELERACIÓN

La aceleración nos proporciona una información muy valiosa a la hora de modelar una trayectoria, por lo que es recomendable su conocimiento una vez el vehículo está en funcionamiento.

Es difícil el conocimiento de la aceleración mediante un sensor incorporado en el vehículo y aún más el modelado de dicho sensor en un entorno virtual, por lo que se ha optado por la obtención teórica de dicha variable. Como sabemos para la obtención teórica de la aceleración se debe realizar la doble derivada de la posición. Por lo que nos adentramos en el mismo problema que en el apartado anterior la realización de una doble derivada sobre una matriz de traslación, pero según [Lovengrove et al, 2013] se puede obtener mediante las siguientes operaciones:

$$\ddot{T}_{w,s}(\tau) = T_{w,i-1}[\ddot{A}_0 A_1 A_2 + A_0 \ddot{A}_1 A_2 + A_0 \dot{A}_1 \dot{A}_2 + 2(\dot{A}_0 \dot{A}_1 A_2 + \dot{A}_0 A_1 \dot{A}_2 + A_0 \dot{A}_1 \dot{A}_2)] \quad (5.5)$$

En donde  $\dot{A}_j$  y  $A_j$  están definidas en las ecuaciones 5.3 y 5.2 respectivamente y la doble derivada de A queda definida de la siguiente manera:

$$\ddot{A}_j = \dot{A}_j \Omega_{i+j} \dot{\ddot{B}}(\tau)_j + A_j \Omega_{i+j} \ddot{\ddot{B}}(\tau)_j \quad (5.6)$$

Para la obtención de la ecuación 5.6 es necesario el conocimiento de la derivada de B y su doble derivada, la derivada de B se puede apreciar en la ecuación 5.4 mientras que la doble derivada de B se puede obtener mediante la siguiente ecuación:

$$\ddot{\ddot{B}}(\tau) = \frac{1}{\Delta t^2} C \begin{bmatrix} 0 \\ 0 \\ 2 \\ 6\tau \end{bmatrix}$$

Una vez definidas todas las ecuaciones, se procede a realizar la doble derivada de la matriz de traslación (ecuación 5.6) que como resultado se obtiene otra matriz de 4x4. Para la obtención de la aceleración lineal tenemos que observar la 4ª columna donde la primera fila será la aceleración en el eje de coordenadas "x", la segunda fila será en el eje "y" y la tercera fila en el eje "z". Si nos fijamos en la cuarta columna de la matriz de traslación se situaba el vector de posición y la aceleración es la doble derivada del vector de posición.

Una vez definido el vector aceleración dentro de la doble derivada de la matriz de traslación se procede a la representación de dicho vector, la cual se puede observar en la figura 18. Para facilitar la interpretación de la gráfica se han representado tres gráficas en la que en cada una de ellas se puede observar la aceleración en los distintos ejes de coordenadas.

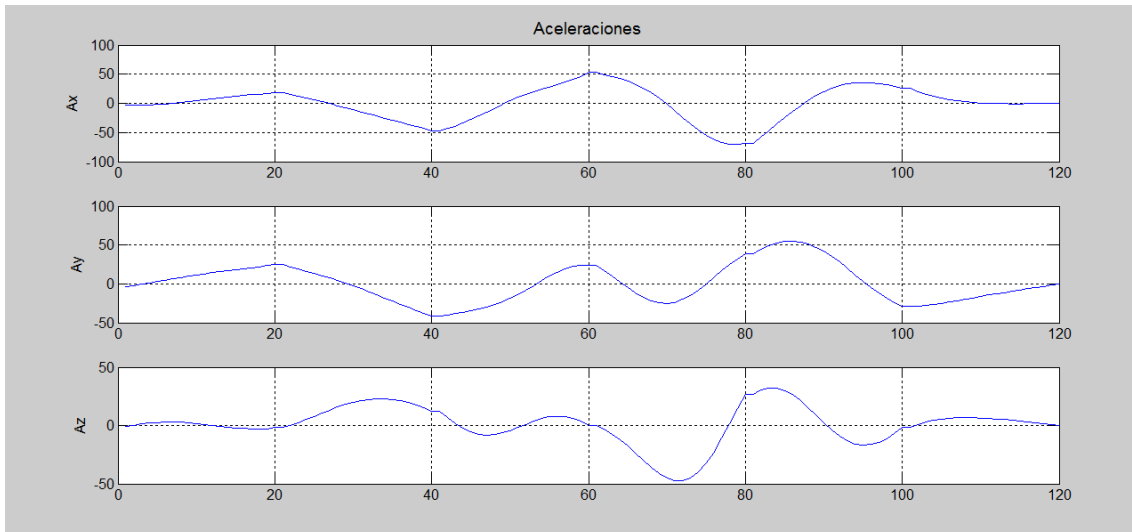


Figura 18 Aceleraciones en el vehículo

En la figura 18 podemos ver cómo avanza la aceleración en función de los puntos calculados para la denominación de la trayectoria, cabe recordad que entre cada par de puntos de control se han calculado 20 iteraciones entre ellos, esto quiere decir que el punto de control primero pertenecerá al punto 1, el segundo punto de

control al punto 20 el tercer punto de control al punto 40 y así sucesivamente, hasta alcanzar los 7 puntos de control definidos en el ejemplo.

La figura 18 muestra un dato muy importante a la hora de extrapolar los cálculos teóricos a la realidad, debido a que un vehículo real tiene que ser capaz de alcanzar las aceleraciones que se describen en la figura, si quiere cumplir con las especificaciones de la trayectoria, por lo que este dato es importante para modelar las características del vehículo real.

### 5.3 GPS

GPS son las siglas en inglés de Global Positioning System, este sistema permite determinar a lo largo de toda la tierra la posición de un objeto. Para la obtención de la posición este sistema se basa en una serie de satélites que orbitan alrededor de la tierra, actualmente son 24 los cuales están sincronizados para cubrir toda la superficie de la tierra. La obtención de las medidas se produce mediante la triangulación de la señal que genera tu dispositivo, entre tres o más satélites.

Como se ha comentado el sistema GPS obtiene su ubicación mediante la triangulación de su señal, esto se realiza midiendo la distancia entre el satélite y el aparato receptor, una vez que se ha obtenido dichas distancias se produce al estudio conjunto de las medidas dándonos un único punto posible que cumpla todas las medidas conjuntamente.

A la hora de modelar un sistema GPS en nuestra trayectoria, nos es interesante los diferentes errores que provocan una distorsión en las medidas, se han identificado los siguientes errores:

- Debido a los efectos atmosféricos
- Errores en la órbita de los satélites
- Errores en la precisión del reloj.
- El número de satélites, cuantos más satélites utilices para la obtención de tu ubicación mayor será la precisión de la medida proporcionada por ellos.
- Errores de multitrajectory.
- Geometría de los satélites

#### 5.3.1 Errores atmosféricos.

Cuando hablamos de errores atmosféricos nos referimos a los errores que se producen en la troposfera y en la ionosfera. Por lo que vamos a desarrollar los errores producidos en dichas capas de una manera diferenciada.

### 5.3.1.1 Ionosfera

Lo primero es entender el proceso que se produce en la ionosfera, los rayos ultravioletas solares y otros tipos ionizan las partículas gaseosas que se encuentran en dicha capa liberando electrones, esto provoca que la ionosfera sea un medio disperso de propagación.

El índice de refracción de la ionosfera depende de la frecuencia de la onda y afecta de diferente manera a la fase y al código, por eso las ondas de GPS sufren retardo o adelanto según sea fase o código.

Según [Garrido-2016] este efecto puede eliminarse mediante una combinación lineal de dos frecuencias (L1 y L2), generando así una señal que elimina este retraso, denominada libre ionosférica (L3). Para poder realizar este método es necesario que el receptor GPS sea de doble frecuencia, pero no todos los GPS fabricados son de una única frecuencia por lo que no sería posible aplicar este método de corrección. Según [Leica] existe otro método de corrección de este efecto, el cual consiste en el cálculo de un valor promedio de la reducción de la luz causada por la ionosfera, y aplicar este factor a una serie de cálculos a resolver. Al trabajar con un valor promedio, lógicamente este resultado no será correcto durante todo el tiempo, por lo que se produce un error en el posicionamiento del vehículo. Dicho error quedara influenciado por el ángulo de elevación de los satélites como se puede apreciar en la figura 19, que nos muestra que para un ángulo de elevación bajo la onda tiene que recorrer una mayor distancia en la ionosfera, produciéndose así una mayor dispersión.

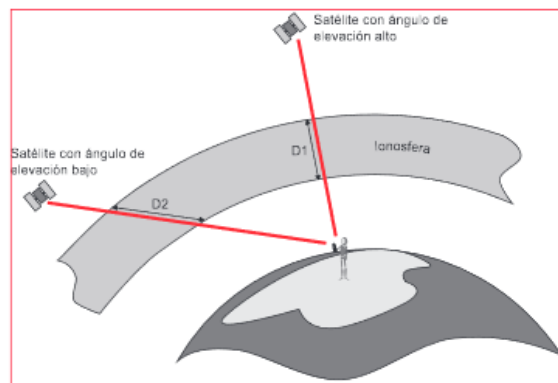


Figura 19

### 5.3.1.2 Troposfera

Como ya se sabe la troposfera es la capa de la atmosfera terrestre que está en contacto con la tierra. Se trata de un medio no disperso por lo que no tienen influencia sobre la frecuencia, pero sí que producirá un retraso de la señal. Sobre todo estos efectos se concentran en los primeros kilómetros donde la onda estará afectada por los efectos de la humedad.

Según [Garrido, 2016] el retardo depende de la ruta de la señal a través de la atmósfera neutra y por eso se debe modelar en función del ángulo de elevación del satélite. El retardo calculado sólo para la observación cenital, necesita de funciones de proyección para retardos diferentes a observables cenitales, y por eso se han desarrollado funciones de mapeo, como la desarrollada por Marini, Niell, Viena, Global Mapping etc.

### 5.3.2 Error en el reloj

Otro error que se produce en estos sistemas es la inexactitud del reloj receptor, en el reloj de los satélites no se produce un error considerable debido a que cada satélite está dotado de un reloj atómico el cual posee una precisión de nanosegundos.

Pequeñas discrepancias entre los errores del receptor y del satélite puede provocar una distorsión en las distancias calculadas. Según [mio] existen dos posibles soluciones:

- La primera solución consiste en instalar un reloj atómico en cada receptor GPS, esta solución es muy poco factible debido a que un reloj atómico cuesta en torno a 100.000 dólares.
- Usar un recurso matemático para contar el error de cronometraje según la forma en que el receptor detecta las señales de tres o más satélites, lo cual permite a dicho receptor reiniciar el reloj. Este es el método más usado por los fabricantes para el ajuste del reloj receptor.

Como se ha comentado anteriormente los relojes atómicos poseen una exactitud de nanosegundos, exactamente de 3 nanosegundos. Algunas veces estos relojes presentan una variación en la velocidad de marcha, para corregir estos errores el departamento de defensa de los estados unidos se encarga de controlar los relojes de estos satélites para que no se produzcan estos desajustes.

Es muy importante la precisión de los relojes debido a que aunque las distancias son muy grandes la onda viaja a la velocidad por lo que los tiempos de trayectoria son del orden de milésimas de segundo.

### 5.3.3 Errores de multitrayectoria.

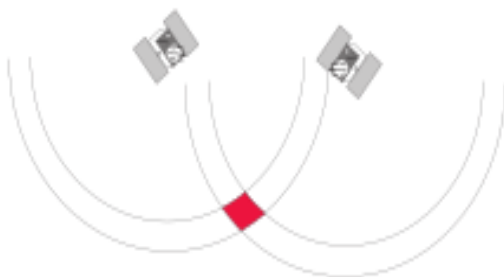
Este proceso se produce cuando el receptor se sitúa cerca de grandes superficies planas como lagos o edificios, los cuales producen que la señal del GPS no viaje directamente al receptor sino que primero rebote en ellos y posteriormente sea recibido por el receptor, lo que provoca un error en la medición de la distancia.

Según [leica], se puede reducir este tipo de errores utilizando antenas GPS especiales que incorporan un plano de tierra, el cual evita que las señales con poca elevación lleguen a la antena.

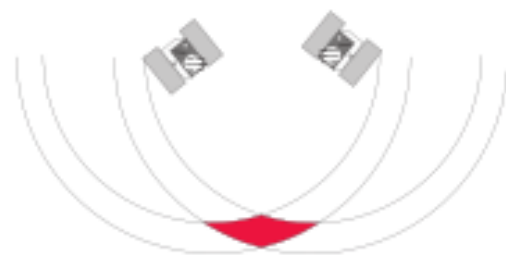
Este error solo es considerado para mediciones de alta precisión, por lo que los sensores ordinarios no poseen estos elementos.

#### 5.3.4 Posición de los satélites.

La posición de los satélites tiene un impacto en el error de lectura del GPS. En las figuras 20 y 21 puede observarse que la precisión de la medida depende del ángulo que hay entre los satélites, los satélites que poseen entre si un mayor ángulo de separación con respecto al receptor realizan una mejor medición que los que poseen poco ángulo de separación



*figura 20 satélites con poca incertidumbre de posición*



*figura 21 satélites con alta incertidumbre de posición*

#### 5.3.5 Modelado

En este apartado se va a proceder a la implementación de un sensor de tipo GPS en la trayectoria de B-splines cúbicos con rotación (apartado 4.3), la misma en la que se ha implementado para los otros dos sensores de velocidad y aceleración.

En los anteriores apartados se ha descrito los diferentes errores que se producen en la medición de un GPS y las formas que se usan para resolver dichos errores. En este apartado se van a modelar dichos errores de cálculo en nuestra trayectoria de la figura 15.

En el mercado actual existen infinidad de dispositivos GPS, los cuales poseen una diferente precisión del cálculo de las coordenadas. Se ha decidido simular un dispositivo cotidiano y no uno de finalidades científicas o militares los cuales poseen una gran exactitud en el cálculo de las coordenadas objeto receptor.

Como se aprecia en la figura 22 existen diferentes tipos de GPS cabe mencionar que el GPS diferencial es aquel que además de recibir y procesar la información de los satélites, recibe y procesa, simultáneamente, otra información adicional procedente de una estación terrestre situada en un lugar cercano. Esta información extra permite corregir posibles inexactitudes de medición desde los satélites.

METODO	FRECUENCIA	OBSERVABLES	PRECISION	APLICACIONES
ABSOLUTO	L1	COD. C/A	±10 m	NAVEGACION
DIFERENCIAL	L1	COD. C/A	1 m	CARTO/GIS
DIFERENCIAL	L1	C/A Y FASE	1cm + 2ppm	TOPOGRAFIA
DIFERENCIAL	L1 Y L2	C/A,P,FASE	5mm + 1ppm	TOPO/GEO

Figura 22 Precisión de los sistemas GPS

Para la simulación del GPS en nuestro modelo de trayectoria se van a coger los datos definidos en la figura 22 específicamente la precisión de cada tipo, esta precisión nos define la desviación en las medidas del GPS las cuales dependerán de las condiciones del entorno. Una vez se conoce la precisión del sensor a modelar se procede a su implementación en el programa de cálculo.

Para la implementación en el programa de cálculo se va a generar una matriz de las mismas dimensiones que nuestra matriz de traslación del sólido, en dicha matriz los valores van a estar comprendidos entre los intervalos del valor de precisión del GPS seleccionado. Una vez se ha generado esta matriz aleatoria procedemos a su suma con la matriz de posición reales, obteniendo así una nueva matriz que será la cual nos represente las coordenadas de la trayectoria definidas por el GPS. Cabe destacar que el GPS se utiliza para distancias grandes por lo que se ha aumentado los límites de la trayectoria, por lo tanto ahora los puntos de control estarán delimitados entre 0 y 200 metros en cada uno de los ejes.

En la figura 23 se observa la representación de la trayectoria de un vehículo que posee un GPS estándar. La línea roja nos muestra la trayectoria que nos proporciona el GPS mientras que la línea azul es la trayectoria real.

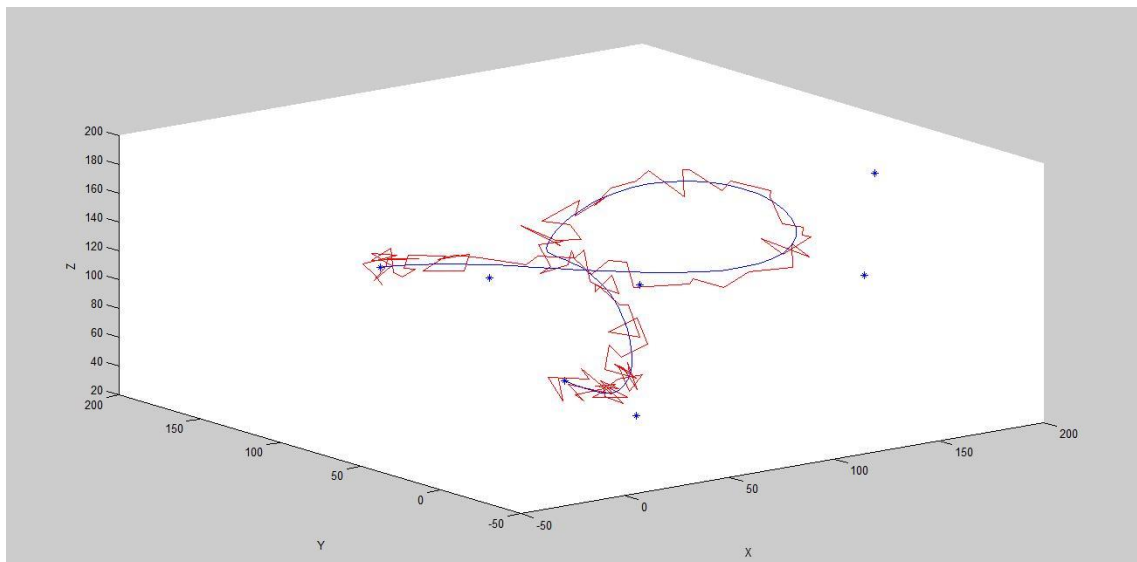


Figura 23 trayectoria con GPS estándar

EN la figura 24 se observa la representación de la trayectoria de un vehículo que posee un GPS diferencial. Como en el caso anterior la línea roja nos muestra la trayectoria del GPS, mientras que la azul nos indica la trayectoria real.



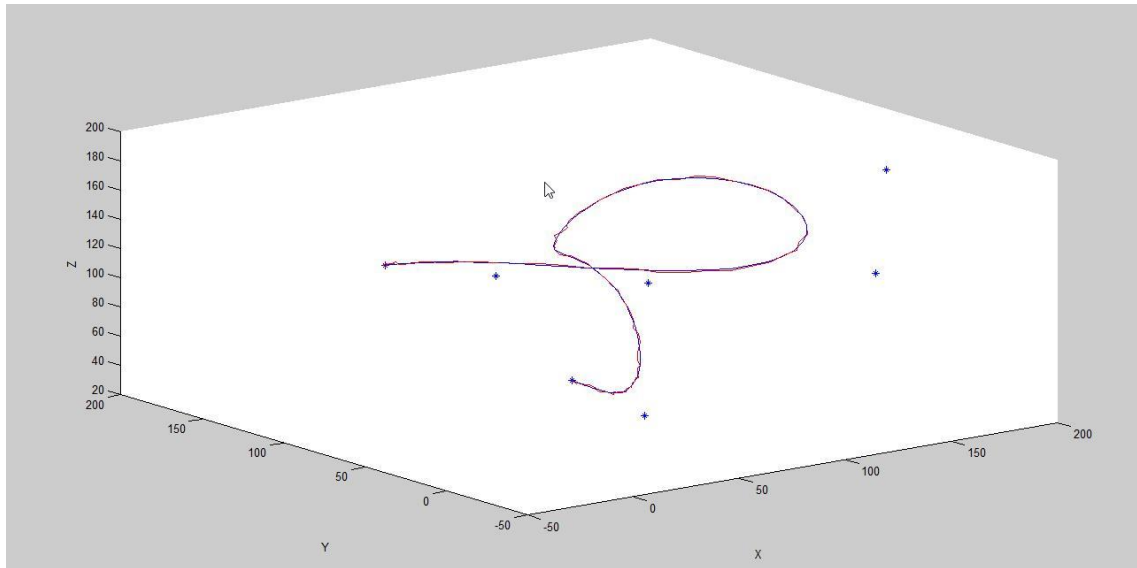


Figura 24 Trayectoria con GPS diferencial

Podemos observar la gran diferencia que hay entre un GPS estándar y uno diferencial y la mayor precisión que el diferencial nos otorga. Cabe destacar que las trayectorias realizadas por vehículos en la vida real son en magnitudes de kilómetros, en ese orden de magnitudes las diferencias entre un modelo y otro no son tan apreciables. Por lo que para distancias grades de traslación usaremos el GPS estándar, mientras que si buscamos una mayor precisión de nuestra trayectoria utilizaremos ya sea un GPS diferencial o uno que nos proporcione una mayor precisión, de acuerdo a nuestras exigencias. Como es lógico aquellos sensores que nos proporcionan una mayor precisión también tienen un mayor coste económico por lo que antes de adquirir un GPS debemos tener claras las condiciones en las que vamos a trabajar con él.

## 6 CONCLUSIONES

En la elaboración de este trabajo se han visto diferentes tipos de trayectorias, realmente las primeras trayectorias no se amoldan a las necesidades que teníamos que cumplir para la elaboración de la trayectoria, debido a que están en 2D y no poseían una componente de rotación. Finalmente se ha modelado la trayectoria de B-splines cúbicos en 3D la cual sí que cumplía nuestras especificaciones y era la trayectoria a buscar desde el inicio del inicio del trabajo. Una vez definida la trayectoria adecuada para el movimiento cabía definir los sensores a elegir, en este caso nos hemos decantado por uno de velocidad, otro de aceleración y un GPS, pero como se define a continuación se podría haber integrado otros sensores como por ejemplo una cámara. A pesar de esto el objetivo del trabajo está cumplido ya que se ha conseguido modelar un sistema de traslación en 3D que posea diferentes sensores, el cual era el objetivo inicial del trabajo.

Este trabajo tiene una buena línea de continuación, la cual sería la implementación de una cámara en el vehículo simulado que sea capaz de percibir el entorno y tomar fotografías de él, esto podría ser realizado en el mismo entorno virtual de Matlab en

el cual hemos trabajado. Para la implementación de la cámara virtual lo primero que deberíamos hacer sería la creación de un entorno virtual que no fuera cuadrado como en nuestro caso, sino que se pareciera a una habitación real (con salientes, obstáculos...), una vez se tuviera el entorno se tendría que proceder a la simulación de una cámara, la cual pueda detectar estos obstáculos y modificar los puntos de control de la trayectoria para reaccionar ante ellos.

Actualmente el campo de la robótica especialmente el de los drones está en pleno apogeo, el trabajo realizado tiene una clara aplicación en dicho campo, ya que la trayectoria de B-splines cúbicos que se ha generado podría adaptarse perfectamente a la trayectoria realizada por un dron de trayectoria planificada anteriormente, mediante la posición del punto inicial y final.

Este trabajo me ha proporcionado un conocimiento en el lenguaje de programación en Matlab, debido a que todas las simulaciones tanto de las trayectorias como de los sensores están realizados en dicho programa, este conocimiento es de gran utilidad ya que Matlab es un programa ampliamente usado en el campo de la ingeniería. Además de un conocimiento de Matlab he obtenido amplias nociones en el trabajo de trayectorias con matrices de traslación, esto es importante ya que son una herramienta muy útil a la hora de modelar un movimiento.

## 7 BIBLIOGRAFÍA

---

[Murray et al, 1994] Richard M. Murray, Zexiang Li y S. Shankar Sastry. A Mathematical Introduction to Robotic Manipulation, CRC Press, pages22-58, 1994.

[Lovengrove et al, 2013] Steven Lovengrove, Alonso Patron-Pérez y Gabe Sibley. Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras. In department of computer Science, George Washington University, 2013.

[Garrido-2016] Natalia Garrido-Villen. Errores atmosféricos en GPS. Estudio y eliminación. Universidad de valencia, 2016.

[Guerrero-2013] José Jesús Guerrero. Asignatura automatización flexible y robótica, Universidad de Zaragoza. Temas 5 y 8, 2013.

[Mio] <https://eu.mio.com/es/es/global-positioning-system-4992.htm>

[Leica] Introducción al sistema GPS (sistema de posicionamiento global), Leica. Suiza 1999.

[velocity-Sensors] [es.slideshare.net/Manishd94/velocity-sensors-inrobotics](https://es.slideshare.net/Manishd94/velocity-sensors-inrobotics)

[González] Carlos González Morcillo. Splines: Curvas y Superficies, Introducción al dibujo de curvas de aproximación e interpolación por computador. Universidad de Castilla-La Mancha.

# ANEXOS

## ANEXO A: ÍNDICE FUNCIONES DE MATLAB

---

Para simular cada una de las trayectorias que se han descrito a lo largo de este trabajo, se han creado diferentes funciones de Matlab. En este apartado se va a proceder a la enumeración y descripción de cada una de las funciones creadas.

### A.1 TRAYECTORIA RECTILÍNEA UNIFORME

- **function** `tray=generar_trayectoria(v,npuntos)`:

**IN:** `v`—velocidad constante del movimiento.

`npuntos`— nº de puntos de control que va a poseer la trayectoria

**OUT:** `tray`— matriz que nos describe la posición del vehículo durante la trayectoria

`Figure1`—nos genera la representación gráfica de la trayectoria modelada en 2D

- **function** `tramo = generar_tramo (q0,q1,t0,t1,v)`

**IN:** `q0`—posición del vehículo al inicio del intervalo de cálculo.

`q1`— posición del vehículo al final del intervalo de cálculo.

`t0`— tiempo en el inicio del intervalo de cálculo.

`t1`— tiempo en el final del intervalo de cálculo

`v`— velocidad constante que posee el sólido a lo largo de la trayectoria.

**OUT:** `tramo`— posiciones del vehículo en el intervalo de cálculo entre dos puntos de control.

### A.2 TRAYECTORIA ACELERADA

- **function** `[tray, tiempo]= trayectoria_acelerada (puntos,velocidad)`

**IN:** `puntos`—vector de puntos de control de la trayectoria, dimensiones (2xN).

`Velocidad`—velocidad en cada punto de control, dimensiones (1xN).

**OUT:** `tray`— matriz que nos describe la posición del vehículo durante la trayectoria

`Tiempo`—vector de tiempos de paso por cada posición calculada de la trayectoria.

`Figure1`—representación gráfica en 2D de la trayectoria del vehículo.

- **function** `[tramo, t]= generar_tramoacelerado (q0,q1, v0,v1,t0,t1)`

**IN:** `q0`— posición del vehículo al inicio del intervalo de calculo

`q1`— posición del vehículo al final del intervalo de cálculo.

`v0`— velocidad del vehículo al inicio del intervalo de cálculo.

`V1`—velocidad del vehículo al final del intervalo de cálculo.

`T0`—tiempo de paso del vehículo por el punto inicio del intervalo.

`T1`—tiempo de paso del vehículo por el punto final del intervalo.

**OUT:** tramo— posición del vehículo entre cada par de puntos de control.  
t— tiempos de paso de cada posición calculada en el intervalo entre dos puntos de control

### A.3 B- SPLINES CÚBICOS SIN MATRIZ DE ROTACIÓN

- **function** tray= splines\_posicion2 (pos, t)

**IN:** pos— vector de puntos de control de la trayectoria. Dim (2xN)  
t— vector de tiempo de paso por cada uno de los puntos de control. Dim (1xN)

**OUT:** tray—matriz que nos describe la posición del vehículo durante la trayectoria  
Figure1 — representación gráfica de la trayectoria en 2D de B-Splines cúbicos sin matriz de rotación

### A.4 B-SPLINES CÚBICOS CON MATRIZ DE ROTACIÓN

- **function** [tray,vel,acel]= generate\_trayectoria2(Npoints, t)

**IN:** Npoints—nº de puntos de control.  
t— vector del tiempo de paso por cada uno de los puntos de control

**OUT:** tray— matriz que nos describe la posición del vehículo durante la trayectoria  
vel— matriz que nos describe la velocidad del vehículo durante la trayectoria  
acel— matriz que nos describe la aceleración del vehículo durante la trayectoria  
figure1— representación gráfica de la trayectoria que nos describe la orientación mediante los focos de control además de la trayectoria del sensor GPS, 3D  
figure2— representación de las velocidades en cada uno de los ejes de coordenadas  
figure3 — representación de las aceleraciones en cada uno de los ejes de coordenadas

- **function** points\_sphere = rand\_unit\_sphere (Npoints)

**IN:** Npoints—nº de puntos de control.

**Out:** points\_sphere— vector de dirección del giro (w) generado de manera aleatoria.

- **function** T=expMapSE3(w\_t,v\_t)

Realiza la operación exponencial en las matrices de traslación.

**IN:** w\_t—vector director del giro en el espacio se(3) de dimensiones 3x1  
v\_t — vector v de velocidad en el espacio se(3) dim 3x1.

**OUT:** T—matriz de traslación en el espacio SE(3) de dimensiones 4x4

- **function** `[omega_t, v_t] = logMapSE3(T)`

Esta función realiza la operación logaritmo para el cambio de espacio de las matrices de traslación

**IN:** T—matriz de traslación de dimensiones 4x4 en el espacio SE(3)

**OUT:** omega\_t— vector director del giro en el espacio se(3) de dimensiones 3x1  
V\_t— vector v de velocidad en el espacio se(3) dim 3x1

- **function** `[pw1,pw2,pw3,pw4,pw5]= saveCamera (Tnew, scale_cam)`

**IN:** Tnew—matriz de traslación, dim4x4.

Scale\_cam—constante que define las dimensiones del foco.

**OUT:** pw1— punto 1 del foco de orientación  
pw2— punto 2 del foco de orientación.  
pw3— punto 3 del foco de orientación.  
pw4— punto 4 del foco de orientación.  
pw5— punto 5 del foco de orientación

- **function** `W= skew(w)`

**IN:** w—vector director del giro.

**OUT**—matriz antisimétrica del vector director de giro

- **function** `drawcamera (P1, P2, P3, P4, P5)`

**IN:** P1—punto 1 del foco de orientación  
P2—punto 2 del foco de orientación  
P3—punto 3 del foco de orientación  
P4—punto 4 del foco de orientación  
P5—punto 5 del foco de orientación.

**OUT:** figure – a partir de los 5 puntos de entrada dibuja un foco de orientación de un punto en la trayectoria.

- **function** `[velocidad, aceleración]= generate_acelerationAndVelocity ( B, devB, ddevB, T10, T11, T12, T13)`

**IN:** B— vector de dimensiones 4x1, el cual nos viene definido por el tipo de spline  
devB— derivada del vector B, dim 4x1.  
ddevB— doble derivada del vector B, dim 4x1.  
T10— matriz de control anterior al intervalo de cálculo  
T11—matriz de control inicial del intervalo de cálculo  
T12—matriz de control final del intervalo de cálculo  
T13—matriz de control siguiente del intervalo de cálculo

**OUT:** velocidad— vector velocidad en la posición de cálculo, dim 3x1.  
aceleración— vector aceleración en la posición de cálculo, dim 4x1.

