



CAMPUS
DE EXCELENCIA
INTERNACIONAL



Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Facultad de Informática

TRABAJO FIN DE GRADO

Teller: Telemetría con GPS

AUTOR: Javier Bailón Villafranca

TUTOR: Ángel Herranz Nieva

Madrid, Junio 2014

Índice general

RESUMEN	v
SUMMARY	vii
1 INTRODUCCIÓN	1
1.1 Antecedentes	1
1.2 Motivación	2
1.3 Objetivos	2
2 ANÁLISIS	5
2.1 Datos del GPS	5
2.1.1 BT747	7
2.2 Datos a extraer	7
2.2.1 Información al piloto	9
3 MODELOS MATEMÁTICOS	11
3.1 Modelo usado	11
3.1.1 Tiempo	11
3.1.2 Radio de curvatura	12
3.1.3 Distancia real	13
3.1.4 Aceleración tangencial	13
3.1.5 Aceleración centrípeta	13

3.1.6	Aceleración total	14
3.1.7	Ángulo de inclinación	14
3.1.8	Velocidad de inclinación	15
3.2	Media móvil	15
3.3	Modelo descartado	16
4	IMPLEMENTACIÓN	19
4.1	Lectura del CSV	19
4.2	Aplicación de las fórmulas	20
4.3	Aplicación de la media móvil	21
4.4	Guardado en JSON	21
5	PRUEBAS REALIZADAS	23
5.1	Las fórmulas	23
5.1.1	Tiempo	24
5.1.2	Radio de curvatura	25
5.1.3	Distancia real	26
5.1.4	Aceleración tangencial	27
5.1.5	Aceleración centrípeta	28
5.1.6	Aceleración total	29
5.1.7	Ángulo de inclinación	30
5.1.8	Velocidad de inclinación	31
5.2	El programa	32
5.2.1	Recorrido en línea recta	32
5.2.2	Recorrido con curvas	32
5.2.3	Recorrido en circuito	33
6	CONCLUSIÓN	35

RESUMEN

En el campo del motociclismo y el automovilismo de competición se debe disponer de tecnología que ayude en la conducción y el aprendizaje del piloto. La telemetría juega un papel que es clave en este aspecto. Gracias a GPS precisos y que ofrecen una gran variedad de información, el piloto puede observar cualquier defecto en su conducción. Sin embargo, la mayor parte de los pilotos que se dedican al motociclismo y el automovilismo de manera amateur no puede permitirse la compra de estos dispositivos.

A lo largo de este documento se explica el trabajo realizado para crear Teller. Teller es una aplicación que recoge los datos de un GPS de bajo coste, genera información mediante cálculos físicos realizados con Erlang y visualiza dicha información para los pilotos que quieren mejorar. Con esta aplicación se pretende ofrecer una alternativa barata de telemetría sin necesidad de gastarse el dinero en GPS precisos pero demasiado caros.

SUMMARY

In the field of motorcycling racing and motorsport technology should be available to assist in driving and learning from the pilot. Telemetry plays a key role in this regard. Thanks to accurate GPS which offer various information, the pilot can see any fault in his driving. However, most of pilots who engage motorcycling racing and motorsport in the amater way can not afford to purchase these devices.

Throughout this document the work done to create Teller is explained. Teller is an application that collects data from a low cost GPS, it generates information by physical calculations made in Erlang and it displays this information for pilots who want to improve. This application aims to provide a cheap alternative of telemetry without spending money on accurate but expensive GPS.

1

INTRODUCCIÓN

1.1 Antecedentes

Dedicarse al motociclismo o al automovilismo de competición no requiere solamente de un vehículo y un duro entrenamiento, también se necesita tecnología punta. Esta tecnología puede servir para mejorar las prestaciones del vehículo o bien para mostrar detalles de la conducción del piloto. Esto último resulta ser una ayuda crucial ya que, gracias a la visualización de estos resultados, el piloto puede evaluar en qué está fallando y así rectificar el error en un futuro. Todo esto es posible gracias a los sistemas de telemetría de los que dispone el vehículo.

La telemetría se define como un proceso de comunicación mediante el cual se realizan mediciones y recogidas de datos en puntos de difícil acceso y se transmite a equipos de monitorización. Sus orígenes se remontan a mediados del siglo XIX siendo un método rudimentario para transmitir información entre el Zar de Rusia y su ejército. A partir de ahí ha ido evolucionando hasta aplicarse en un amplio abanico de campos, desde agencias espaciales y ejército hasta industrias y exploración científica.

Como ya se ha dicho, también es empleado en el campo del motociclismo y el automovilismo, donde es casi indispensable el envío de información mediante comunicación inalámbrica para proporcionar los datos en tiempo real. Nada

más recibir la información el sistema se encarga de recoger aquello que necesita y mostrar nuevos datos que sean de utilidad para el piloto. Dichos valores pueden ser desde datos simples como la velocidad en un punto o el tiempo que ha tardado en dar una vuelta a un circuito, hasta datos complejos como el rendimiento del motor o el desgaste de los frenos.

1.2 Motivación

Como es de suponer, este tipo de tecnología exige de equipos realmente complejos y caros que los pilotos amater no suelen poder permitirse. Es por este motivo que surge este trabajo.

El proyecto consiste en trabajar con un dispositivo GPS de alta frecuencia pero de bajo coste y extraer la máxima información posible para luego poder realizar un informe con datos que ayuden a evaluar la conducción de un piloto de motociclismo. Aunque de un GPS de mercado no se pueda obtener información tan compleja como la temperatura del motor o el nivel de aceite, sí que puede proporcionar datos que permitan calcular aceleraciones, fuerzas G o la inclinación de la moto. Todo ello se puede obtener mediante sencillos cálculos de física y trigonometría.

Este tipo de GPS tampoco suele ofrecer el envío de datos mediante comunicación inalámbrica para poder visualizar en tiempo real los resultados, por ello está orientado a motociclistas que realizan sus vueltas de entrenamiento y luego quieren observar los resultados.

1.3 Objetivos

Hasta ahora se ha explicado en qué consiste el proyecto al completo, sin embargo, para este trabajo solo se realiza una parte de dicho proyecto. Este trabajo está destinado a realizar todos los cálculos necesarios para que más tarde, una interfaz pueda mostrar al piloto los datos que le interesen. Por tanto, los objetivos que se desean alcanzar en este trabajo son:

- Recoger los datos de un dispositivo GPS que sean necesarios para obtener información relevante para el piloto.

- Generar un modelo con las fórmulas necesarias para llegar a obtener dicha información.
- Implementar el modelo de manera que por cada ristra de datos que recoja en un instante devuelva información de dicho instante en otra ristra de datos.
- Realizar las pruebas necesarias para validar dicha implementación comparando los resultados obtenidos con resultados previamente calculados.

2

ANÁLISIS

En este punto de la documentación del proyecto se realiza un estudio de todos los datos y la información que interviene en el programa. Primero se explican los diferentes datos que ofrece el GPS y, a partir de estos datos, se detallan nuevos datos que se quieren extraer como resultado final para mostrar al piloto que haga uso de la aplicación.

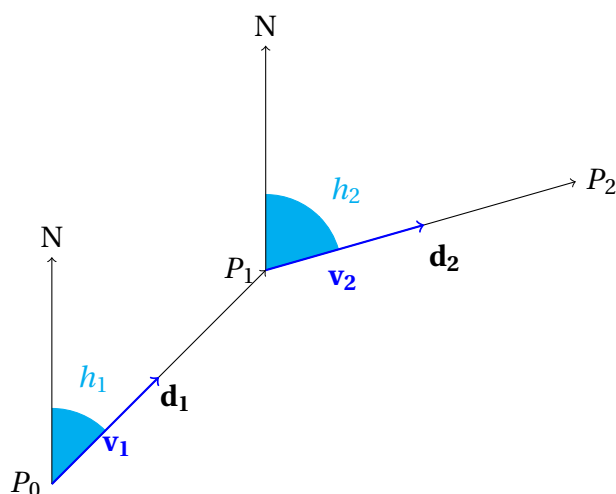
2.1 Datos del GPS

Dado que la aplicación requiere de un GPS ha sido necesario utilizar uno para realizar el trabajo. El GPS que se ha empleado ha sido el iBlue 747A+ cuya frecuencia de toma de datos es de 5Hz (es decir, que cada 200ms se realiza una toma de datos, cinco cada segundo). Lo ideal sería tener un GPS con una frecuencia muy alta, de manera que haga un mayor número de tomas de datos en el mismo tiempo y, en consecuencia, sea más preciso.

En cada toma de datos el GPS consigue diversa información, sin embargo, no todos los datos interesan para mostrar información al piloto. Los datos que se necesitan para llevar a cabo todos los cálculos y visualizar lo que se desea son los siguientes:

- **Índice (Index):** Este dato es el número de identificación de cada instante en el que el GPS hace una recopilación de datos, es decir, que es la identificación de cada punto del recorrido.
- **Fecha (Date):** Este dato es el día, mes y año en el que se recopila la información para un punto del recorrido.
- **Hora (Time):** Este dato es la hora, minuto y segundo en el que se recopila la información para un punto del recorrido.
- **Latitud (Latitude):** Este dato indica la distancia desde el ecuador hasta el punto del recorrido y sirve, junto a la longitud, para determinar la posición del punto en la Tierra que luego se visualizará en el mapa de la aplicación. El GPS lo representa en grados.
- **Longitud (Longitude):** Este dato indica la distancia desde el meridiano de Greenwich hasta el punto del recorrido y sirve, junto a la latitud, para determinar la posición del punto en la Tierra que luego se visualizará en el mapa de la aplicación. El GPS lo representa en grados.
- **Heading:** Este dato indica el sentido del movimiento y corresponde con los grados respecto al norte de la Tierra en sentido horario. Se representa tanto en las fórmulas como en las figuras como h .
- **Distancia (Distance):** Este dato es la distancia en línea recta desde el punto anterior. El GPS lo representa en metros. Se representa tanto en las fórmulas como en las figuras como d .
- **Velocidad (Speed):** Este dato es la velocidad a la que se mueve en cada punto del recorrido. El GPS lo representa en kilómetros por hora. Se representa tanto en las fórmulas como en las figuras como v .

Con estos datos se puede realizar una representación gráfica del recorrido que realiza el GPS visualizando como un punto cada captura que realiza. A continuación se muestra una figura en la que se pueden observar tres puntos de un recorrido (P_i) con sus respectivos headings (h_i), distancias (d_i) y velocidades (v_i).



2.1.1 BT747

Para extraer los datos del GPS al ordenador se ha optado por emplear el programa GPS Data Logger Software - BT747. Con esta aplicación se pueden seleccionar los datos que interesan del GPS y guardarlos en distintos formatos de salida. De entre los distintos formatos que se ofrece se puede devolver en un archivo CSV (Comma-Separated Value). Este formato es con el que Teller trabaja para leer los datos de entrada.

Además de que el BT747 es un software sencillo de manejar, la razón que lleva a utilizar este programa es que puede extraer los datos de una gran variedad de GPS de bajo coste entre los que se encuentra el iBlue 747A+ (el GPS con el que se han realizado las pruebas). De esta modo, al trabajar con las salidas generadas por el BT747, Teller es compatible con toda esa variedad de GPS.

2.2 Datos a extraer

Para poder realizar la visualización de información la aplicación requiere de una serie de datos que, por sí solo, no los aporta el GPS. Por ello, lo primero que se ha realizado en el proyecto ha sido establecer los datos que se quieren obtener como resultado. En la siguiente lista figuran los datos a extraer:

- **Índice (Index):** Este dato es el número de identificación de cada instante en el que el GPS hace una recopilación de datos, es decir, que es la identificación de cada punto del recorrido. Coincide con el Índice que se recoge

del GPS.

- **Fecha (Date):** Este dato es el día, mes y año en el que se recopila la información para un punto del recorrido. Coincide con la Fecha que se recoge del GPS.
- **Hora (Time):** Este dato es la hora, minuto y segundo en el que se recopila la información para un punto del recorrido. Coincide con la Hora que se recoge del GPS.
- **Latitud (Latitude):** Este dato indica la distancia desde el ecuador hasta el punto del recorrido y sirve, junto a la longitud, para determinar la posición del punto en la Tierra que luego se visualizará en el mapa de la aplicación. Se va a representar en grados. Coincide con la Latitud que se recoge del GPS.
- **Longitud (Longitude):** Este dato indica la distancia desde el meridiano de Greenwich hasta el punto del recorrido y sirve, junto a la latitud, para determinar la posición del punto en la Tierra que luego se visualizará en el mapa de la aplicación. Se va a representar en grados. Coincide con la Longitud que se recoge del GPS.
- **Heading:** Este dato indica el sentido del movimiento y corresponde con los grados respecto al norte de la Tierra en sentido horario. Se representa tanto en las fórmulas como en las figuras como h . Coincide con el Heading que se recoge del GPS.
- **Distancia real (Real distance):** Este dato es la distancia real recorrida desde el punto anterior. Se va a representar en metros. Se representa tanto en las fórmulas como en las figuras como D .
- **Distancia total (Total distance):** Este dato es la distancia real recorrida desde el inicio. Se va a representar en kilómetros.
- **Velocidad (Speed):** Este dato es la velocidad a la que se mueve en cada punto del recorrido. Se va a representar en kilómetros por hora. Se representa tanto en las fórmulas como en las figuras como v . Coincide con la Velocidad que se recoge del GPS.
- **Aceleración tangencial (Speedup):** Este dato es la variación de rapidez de un punto al siguiente respecto al tiempo transcurrido en el intervalo. Se va a representar en metros por segundo al cuadrado. Se representa tanto en las fórmulas como en las figuras como a^t .

- **Radio de curvatura (Curvature radius):** Este dato es el radio de la circunferencia que se genera al realizar el recorrido entre dos puntos. Se va a representar en metros. Se representa tanto en las fórmulas como en las figuras como r .
- **Aceleración centrípeta (Centripetal acceleration):** Este dato es el cambio de dirección de la velocidad al recorrer una trayectoria. Se va a representar en metros por segundo al cuadrado. Se representa tanto en las fórmulas como en las figuras como a^c .
- **Aceleración total (Total acceleration):** Este dato es el cambio de velocidad de un punto al siguiente respecto al tiempo transcurrido en el intervalo. Se va a representar en metros por segundo al cuadrado. Se representa tanto en las fórmulas como en las figuras como a .
- **Ángulo de inclinación (Lean angle):** Este dato indica los grados de inclinación que adquiere una moto al tumbarla para realizar un giro. Se va a representar en grados. Se representa tanto en las fórmulas como en las figuras como I .
- **Velocidad de inclinación (Lean speed):** Este dato indica la velocidad a la que cambia el ángulo de inclinación de un punto al siguiente respecto al tiempo transcurrido en el intervalo. Se va a representar en grados por segundo.

2.2.1 Información al piloto

Como se ha mencionado con anterioridad esta aplicación está orientada a motoristas, aunque también pueden hacer uso de la misma pilotos de automóvil. En la conducción de un piloto de motocicleta sólo influyen dos elementos:

- **El acelerador (y freno):** Durante el recorrido que realiza el piloto, éste puede acelerar o frenar. Estas acciones tienen su efecto en el speedup (aceleración tangencial). Si el piloto acelera el speedup aumenta y si frena el speedup disminuye.
- **El manillar:** Además de poder acelerar o frenar el piloto puede realizar giros. A esta acción se le llama contramanillar y al realizarla se produce un cambio en el radio de curvatura del tramo.

A partir del speedup y del radio de curvatura se pueden alterar los demás datos. Por lo tanto, el resto de datos dependen del acelerador o del manillar:

- Los cambios producidos en el speedup provocan cambios en la velocidad de la moto, por tanto, de manera indirecta, el acelerador influye en la velocidad.
- A partir de la velocidad y el radio de curvatura se obtiene la aceleración centrípeta. En consecuencia, la aceleración centrípeta varía con el acelerador y con el manillar.
- El ángulo de inclinación se produce debido a la aceleración centrípeta, por tanto también depende de ambos elementos.

Resumiendo, con estos dos elementos se puede controlar todos los datos que aparecen por pantalla. Por ello, viendo los datos en los que se falla, el motorista puede saber en que elemento está cometiendo el error y rectificar la próxima vez.

3

MODELOS MATEMÁTICOS

Una vez conocidos los datos de entrada y los de salida se procedió a diseñar un modelo matemático que satisficiera las necesidades del proyecto. Para encontrar dicho modelo se pensaron diferentes maneras de alcanzar los datos de salida a partir de los datos de entrada y, a partir de ello, se idearon diferentes modelos.

3.1 Modelo usado

Tras descartar varios modelos al final se decidió usar el siguiente modelo que ahora se explica con sus respectivas fórmulas.

3.1.1 Tiempo

Para realizar cálculos como las aceleraciones se necesita el incremento de tiempo, por tanto es necesario tener el tiempo en cada punto. En un principio el GPS ofrece un tiempo, no obstante, se ha decidido calcularlo a través de la velocidad y la distancia recorrida, ya que el del GPS no se ha considerado un valor de fiar.

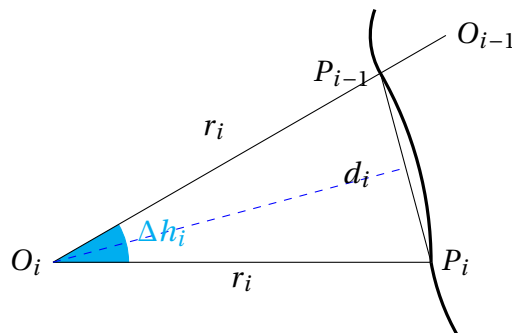
Al ser un GPS de 5Hz en teoría realiza una captura de datos cada 200ms, sin embargo, la toma de datos podría haber llevado otro tiempo, pero el GPS siempre devuelve que ha durado 200ms exactos.

El tiempo en un punto i (t_i) se calcula dividiendo la distancia recorrida desde el punto anterior (d_i) entre la velocidad en dicho punto (v_i).

$$t_i = \frac{d_i}{v_i}$$

3.1.2 Radio de curvatura

Si el vehículo realiza un recorrido curvilíneo entre dos puntos dicho recorrido se puede representar como una parte de una circunferencia. Para el modelo elegido interesa conocer el radio de curvatura de dicha circunferencia, así que para obtenerlo se necesita conocer la distancia entre los dos puntos y el incremento del heading del segundo punto respecto al primero. Se puede ver más claro en la siguiente representación:



En la figura se observa que los radios de curvatura (r) junto con la distancia (d) forman un triángulo isósceles cuyo ángulo (el del centro de la circunferencia) coincide con el incremento de heading (Δh). Si se divide por la mitad quedan dos triángulos rectángulos y cada uno tiene de hipotenusa el radio de curvatura, uno de los catetos es la mitad de la distancia y un ángulo es la mitad del incremento de heading. Conociendo esto el radio de curvatura se puede sacar por trigonometría.

El radio de curvatura en un punto i (r_i) se calcula dividiendo la distancia recorrida (d_i) entre dos veces el seno de la mitad del incremento de heading respecto al punto anterior (Δh_i).

$$r_i = \frac{d_i}{2 \cdot \sin\left(\frac{\Delta h_i}{2}\right)}$$

3.1.3 Distancia real

Un dato que se desea conocer es la distancia real. El GPS informa de la distancia en línea recta entre dos puntos. Este dato sirve si el recorrido es en línea recta, pero cuando el vehículo toma una curva se está perdiendo información y se muestran menos metros de los que se están recorriendo realmente. Es por esto que se necesita calcular a parte la distancia real. Puesto que se está representando una curva como una parte de la circunferencia que forman dos puntos, la distancia real será el arco que se forma.

La distancia real recorrida en un punto i (D_i) se calcula multiplicando 2π veces el radio en dicho punto (r_i) por el incremento de heading respecto al punto anterior (Δh_i) y dividiendo entre 360.

$$D_i = \frac{2\pi r_i \cdot \Delta h_i}{360}$$

3.1.4 Aceleración tangencial

También se quiere mostrar la aceleración tangencial. Dicha aceleración es sencilla de obtener ya que es la variación de rapidez en el tiempo.

La aceleración tangencial en un punto i (a_i^t) se calcula dividiendo el incremento de velocidad respecto al punto anterior (Δv_i) entre el incremento de tiempo respecto al punto anterior (Δt_i).

$$a_i^t = \frac{\Delta v_i}{\Delta t_i}$$

3.1.5 Aceleración centrípeta

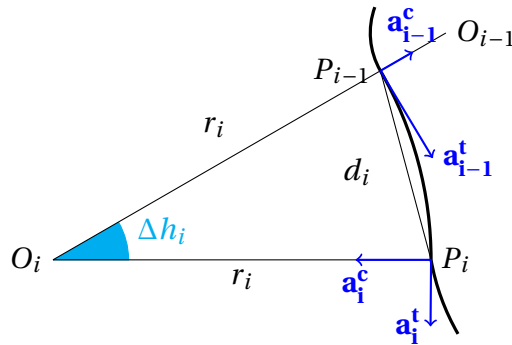
Al igual que se pretende visualizar la aceleración tangencial también se pretende visualizar la aceleración centrípeta. Esta aceleración se puede obtener de dos modos. Mediante la velocidad angular (ω) o mediante la velocidad (v). En este caso se ha optado por calcularlo a partir de la velocidad.

La aceleración centrípeta en un punto i (a_i^c) se calcula elevando al cuadrado la velocidad en dicho punto (v_i) y dividiendo entre el radio de curvatura (r_i).

$$a_i^c = \frac{v_i^2}{r_i}$$

3.1.6 Aceleración total

La aceleración tangencial y la aceleración centrípeta se representan del siguiente modo:



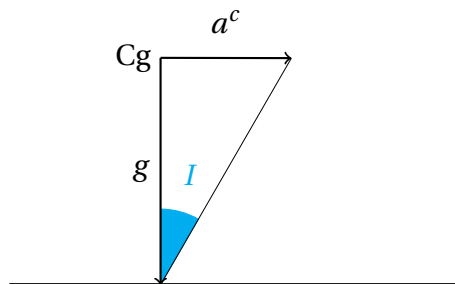
Viendo la figura se puede sacar que el vector de aceleración tangencial (a^t) forma un ángulo de 90° con el vector de la aceleración centrípeta (a^c). Puesto que la aceleración total es la suma de estas dos aceleraciones perpendiculares entre sí, la aceleración total se puede simplificar en un vector cuyas coordenadas X e Y son la aceleración tangencial y centrípeta.

La aceleración total en un punto i (a_i) se calcula realizando la raíz cuadrada de la suma del cuadrado de la aceleración tangencial en dicho punto (a_i^t) con el cuadrado de la aceleración centrípeta en dicho punto (a_i^c).

$$a_i = \sqrt{(a_i^c)^2 + (a_i^t)^2}$$

3.1.7 Ángulo de inclinación

Para obtener el ángulo de inclinación de una moto se ha procedido a realizar un cálculo trigonométrico también. Para comprenderlo mejor se ofrece la siguiente representación:



En esta figura se muestra cómo se obtiene la inclinación (I) de una moto con su propio centro de gravedad (Cg) a partir de la aceleración centrípeta (a^c) y la fuerza de gravedad (g).

El ángulo de inclinación en un punto i (I_i) en grados se calcula realizando la arcotangente de la aceleración centrípeta en dicho punto (a_i^c) dividida por la gravedad (g).

$$I_i = \arctan\left(\frac{a_i^c}{g}\right)$$

3.1.8 Velocidad de inclinación

Una vez se tiene la inclinación se puede obtener la velocidad de inclinación. Del mismo modo que la velocidad, la velocidad de inclinación son los grados inclinados en el tiempo que se recorre un trayecto entre dos puntos.

La velocidad de inclinación en un punto i (v_i^I) se calcula dividiendo el incremento de inclinación respecto al punto anterior (ΔI_i) entre el incremento de tiempo respecto al punto anterior (Δt_i).

$$v_i^I = \frac{\Delta I_i}{\Delta t_i}$$

3.2 Media móvil

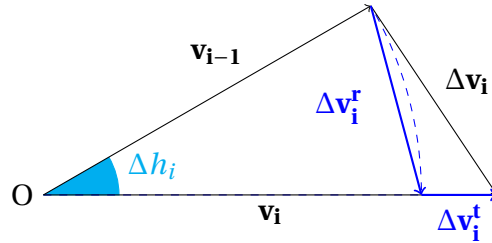
Puesto que los resultados que se obtienen del GPS no son exactos al cien por cien es necesario aplicar un filtro que elimine el ruido y la falta de precisión que se genera. Por ello se ha optado por emplear lo que se conoce como media móvil.

La media móvil se utiliza en conjuntos de datos para crear promedios de diferentes subconjuntos de dichos datos. Con esto se consigue suavizar el resultado evitando picos de información que no deberían existir. En este caso se ha decidido emplear la media móvil con subconjuntos de tres datos, de tal modo que el punto de en medio tomará como nuevo valor la media de dicho subconjunto.

3.3 Modelo descartado

Como ya se ha dicho, para obtener la información a extraer con los datos que se obtienen del GPS se habían propuesto diferentes modelos como alternativas. Para todos los modelos propuestos se había ideado obtener el incremento de velocidad radial y tangencial para un instante a partir de las velocidades en dicho instante y el anterior.

En uno de los modelos se proponía, como se muestra en la siguiente figura, que el incremento de velocidad radial (Δv_i^r) y el incremento de velocidad tangencial (Δv_i^t) para un instante i se obtiene a partir del incremento de heading (Δh_i) y del vector de velocidad para dicho instante (v_i) y para el instante anterior (v_{i-1}).



De esta manera, las fórmulas para el incremento de velocidad radial (Δv_i^r) y para la aceleración radial (a_i^r) quedaban del siguiente modo:

$$\Delta v_i^r = 2v_{i-1} \sin\left(\frac{\Delta h_i}{2}\right)$$

$$a_i^r = \frac{\Delta v_i^r}{\Delta t_i}$$

En este caso, la aceleración con la que se trabajaba era la radial, que al final tendría que dar el mismo resultado que trabajar con la aceleración centrípeta. Para poder validar dicho modelo se realizó una igualación entre dichas aceleraciones y se supuso que el vehículo realizaba un giro circular de 90° a una velocidad constante:

$$a^r = a^c$$

$$\frac{\Delta v^r}{\Delta t} = \frac{v^2}{r}$$

$$\frac{2v \sin\left(\frac{\Delta h}{2}\right)}{\Delta t} = \frac{v^2}{r}$$

$$\frac{2 \sin\left(\frac{90}{2}\right)}{\Delta t} = \frac{v}{r}$$

Al ser un recorrido circular de 90° el vehículo va a una velocidad a la que recorre un cuarto de circunferencia por cada incremento de tiempo:

$$\frac{2 \sin(45)}{\Delta t} = \frac{\frac{2\pi r}{4\Delta t}}{r}$$

$$\frac{\frac{1}{2}}{\Delta t} = \frac{\pi}{4\Delta t}$$

$$1 = \frac{\pi}{2}$$

El resultado final indicaba que el modelo no era válido al ser una inconsistencia. Para el resto de modelos se hizo la misma igualación y también aparecieron inconsistencias, por tanto, también fueron descartados.

4

IMPLEMENTACIÓN

La aplicación ha sido implementada en el lenguaje de programación Erlang. El programa principal se lanza con la función main que recibe dos parámetros de entrada, el fichero de entrada en formato CSV y el fichero de salida que se guardará en JSON. Para hacer la conversión se siguen los siguientes pasos:

1. Una función que lee el CSV y guarda los datos extraídos.
2. Una función que genera la nueva información aplicando las fórmulas físicas.
3. Una función que aplica el filtro que suaviza los resultados.
4. Una función que guarda el resultado final en un documento JSON.

4.1 Lectura del CSV

Como se ha explicado con anterioridad, Teller trabaja a partir de un archivo CSV de entrada. En dicho documento se encuentran todos los datos que se necesitan para realizar los cálculos pertinentes. Comienza con una línea con los nombres de los diferentes datos y las siguientes líneas corresponden con cada captura de datos que ha realizado el GPS ordenando la información igual que en la primera

línea. Este sería un ejemplo de CSV para tres puntos en los que se da la velocidad y la distancia:

```
INDEX,SPEED(km/h),DISTANCE(m),
1,91.234,5.01,
2,91.567,5.12,
3,90.987,4.98,
```

Para realizar la lectura del CSV, esta función hace uso de la librería `ecsv`. Esta librería se encarga de leer un documento y aplicar la función interna que se quiere a cada una de las líneas del documento. En este caso, la función interna que se ha creado se encarga de construir tuplas de cada dato y guardarlas en una lista por cada línea. Dichas tuplas se componen del nombre del dato y su valor. Para el ejemplo anterior, el resultado de realizar esta lectura sería:

```
[{INDEX,1},{SPEED(km/h),91.234},{DISTANCE(m),5.01}},
[{INDEX,2},{SPEED(km/h),91.567},{DISTANCE(m),5.12}},
[{INDEX,3},{SPEED(km/h),90.987},{DISTANCE(m),4.98}]
```

Tras realizar el procesado del fichero se devuelve el resultado al programa principal en una lista compuesta de cada una de las listas correspondientes a cada toma de datos.

4.2 Aplicación de las fórmulas

Una vez leído el archivo CSV se procede a aplicar las fórmulas a los datos leídos. Dado que muchos de los datos dependen de incrementos, es necesario leer un punto junto con el anterior y así obtener los resultados para el punto actual. Esto provoca que en el resultado final haya un punto menos, puesto que para el primer punto no se podrían calcular ciertos datos.

Dado que la entrada de esta función tiene los elementos divididos por tuplas, se puede obtener el valor de cada dato llamando a la tupla por su nombre (el primer elemento) con una función propia de Erlang. Una vez se tienen los datos necesarios se llama a las funciones que se necesitan del módulo Erlang donde están las fórmulas y se obtienen los nuevos resultados con ellas.

Para guardar los nuevos resultados se utiliza el mismo procedimiento. Para cada punto se genera una lista de tuplas que contiene todos los datos que son necesarios para mostrar al usuario la información que se desea.

4.3 Aplicación de la media móvil

Tras aplicar todas las fórmulas se necesita aplicar un filtro que suavice los resultados obtenidos. Este filtro es necesario ya que al no ser un GPS con una precisión exacta la información puede contener ligeros picos que pueden despistar al usuario cuando la visualice. Como se ha explicado anteriormente, el filtro elegido es la media móvil en subconjuntos de tres puntos.

Para llevar a cabo la aplicación del filtro se recorre punto por punto leyendo los datos a los que se les aplicará el filtro correspondientes al punto actual, al anterior y al posterior. A continuación se calcula la media ponderada de los tres y se le asigna ese valor en las nuevas listas que se van generando para cada punto.

Al terminar de aplicar los filtros a todos los puntos se devuelve el resultado del mismo modo que se ha estado haciendo en las otras dos funciones, es decir, en una lista de tuplas para cada punto.

4.4 Guardado en JSON

Finalmente, los resultados deben ser almacenados en un fichero JSON para su posterior visualización. Sin embargo, los resultados no llegan en formato JSON, sino que llegan con el siguiente formato:

```
{INDEX,1},{SPEED(km/h),91.234},{DISTANCE(m),5.01},  
{INDEX,2},{SPEED(km/h),91.567},{DISTANCE(m),5.12},  
{INDEX,3},{SPEED(km/h),90.987},{DISTANCE(m),4.98}
```

Para convertir dicho formato a JSON se necesita de una librería. Para este proyecto se ha utilizado la librería Jiffy, la cuál es bastante simple de utilizar. El resultado ya viene preparado para no tener que realizar grandes cambios y poder ser codificado directamente con Jiffy. El resultado tras la codificación es el siguiente:

```
{"index":1,"speed":91.234,"distance":5.01},  
{"index":2,"speed":91.567,"distance":5.12},  
{"index":3,"speed":90.987,"distance":4.98}
```

Una vez convertido en formato JSON todo está listo para guardarse en un documento. Para ello se hace una llamada a la función que ofrece Erlang para escribir en documentos y se le introduce el resultado en JSON.

5

PRUEBAS REALIZADAS

Tras realizar la implementación es necesario realizar un mínimo de pruebas que den por correcto el funcionamiento del programa. Para ello se han realizado dos tipos de pruebas:

- Pruebas con las fórmulas.
- Pruebas con el programa.

5.1 Las fórmulas

En este tipo de pruebas se ha comprobado que cada una de las fórmulas empleadas funciona correctamente. El modo en el que se han realizado estas pruebas ha sido insertando unos valores y comparando si el resultado coincide con el que debe ser correcto.

5.1.1 Tiempo

Debido a que es una división primero se realizaron pruebas cuyos resultados deberían dar números enteros, a continuación se probó a operar con signos, luego se realizaron divisiones con decimales y por último con 0. Para evitar divisiones entre 0 se obliga a dividir al programa entre 0.0000001 cuando recibe un 0.

Tiempo			
Velocidad(m/s)	Distancia(m)	Esperado(s)	Resultado(s)
4	2	2	2.0
625	25	25	25.0
-625	25	-25	-25.0
625	-25	-25	-25.0
-625	-25	25	25.0
5	2	2.5	2.5
2	5	0.4	0.4
630	25	25.2	25.2
2.5	2.5	1	1.0
2.5	5	0.5	0.5
0	2	0	0.0
5	0	50000000	5.0e7

5.1.2 Radio de curvatura

Puesto que también es una división se han realizado las mismas pruebas que con el tiempo. Para evitar divisiones entre 0 también se obliga a dividir al programa entre 0.0000001 cuando recibe un 0.

Radio de curvatura			
Distancia(m)	Δheading($^{\circ}$)	Esperado(m)	Resultado(m)
4	180	2	2.0
625	60	625	625.0
-625	60	-625	-625.0
625	-60	-625	-625.0
-625	-60	625	625.0
5	2	143.2467	143.24672124637547
1	180	0.5	0.5
625	25	1443.8207	1443.8207235454058
1	60	1	1.0
0.5	180	0.25	0.25
0	2	0	0.0
5	0	2864788975.6541	2864788975.654116

5.1.3 Distancia real

Para la distancia real se han propuesto como pruebas diferentes casos. Para empezar, al multiplicar por π , no saldrán números enteros salvo que el resultado sea 0. Sin embargo se prueban las diferentes combinaciones de signo, operaciones con decimales y valores con 0.

Distancia real			
Radio de curvatura(m)	Δ heading($^{\circ}$)	Esperado(m)	Resultado(m)
4	180	12.5664	12.566370614359172
625	60	654.4985	654.4984694978735
-625	60	-654.4985	-654.4984694978735
625	-60	-654.4985	-654.4984694978735
-625	-60	654.4985	654.4984694978735
1	45	0.7854	0.7853981633974483
0.5	0.5	0.0044	0.004363323129985824
0.5	45	0.3927	0.39269908169872414
0	2	0	0.0
5	0	0	0.0

5.1.4 Aceleración tangencial

Al igual que el tiempo, la aceleración tangencial es una división sencilla, por ello sus pruebas son las mismas.

Aceleración tangencial			
Velocidad(m/s)	Δ Tiempo(s)	Esperado(m/s ²)	Resultado(m/s ²)
4	2	2	2.0
625	25	25	25.0
-625	25	-25	-25.0
625	-25	-25	-25.0
-625	-25	25	25.0
5	2	2.5	2.5
2	5	0.4	0.4
630	25	25.2	25.2
2.5	2.5	1	1.0
2.5	5	0.5	0.5
0	2	0	0.0
5	0	50000000	5.0e7

5.1.5 Aceleración centrípeta

En este caso ocurre algo parecido a la aceleración tangencial, la diferencia es que esta vez el numerador está elevado al cuadrado.

Aceleración centrípeta			
Velocidad(m/s)	Radio de curvatura(m)	Esperado(m/s ²)	Resultado(m/s ²)
4	2	8	8.0
25	5	125	125.0
-25	5	125	125.0
25	-5	-125	-125.0
-25	-5	-125	-125.0
5	2	12.5	12.5
2	5	0.8	0.8
628	25	15775.36	15775.36
2.5	6.25	1	1.0
2.5	2.5	2.5	2.5
0	2	0	0.0
5	0	250000000	2.5e8

5.1.6 Aceleración total

Dado que la aceleración total es una raíz de suma de cuadrados no hay posibilidad de que se haga la raíz de un número negativo ni muchas posibilidades de que el resultado sea entero. Sin embargo sí se contemplan casos en el que las entradas sean negativas o números decimales.

Aceleración total			
Ac. centrípeta(m/s ²)	Ac. tangencial(m/s ²)	Esperado(m/s ²)	Resultado(m/s ²)
4	3	5	5.0
-4	3	5	5.0
4	-3	5	5.0
-4	-3	5	5.0
5	2	5.3852	5.385164807134504
630	25	15775.36	630.4958366238433
1	1	1.4142	1.4142135623730951
2	1.5	2.5	2.5
0.5	0.5	0.7071	0.7071067811865476
0	2	2	2.0
5	0	5	5.0

5.1.7 Ángulo de inclinación

Para el ángulo de inclinación se han propuesto como pruebas diferentes casos. Para empezar, al dividir por π , no saldrán números enteros salvo que el resultado sea 0. Además, sólo se introduce un valor. Sin embargo se prueban las diferentes combinaciones de signo, operaciones con decimales y valores con 0.

Ángulo de inclinación		
ΔAceleración centrípeta(m/s²)	Esperado(°)	Resultado(°)
4	22.1899	22.189883550608762
-4	-22.1899	-22.189883550608762
625	89.1011	89.1010663166456
2.5	14.3017	14.301735017245004
0.5	2.9187	2.9187443726355857
9.80665	45	45
0	0	0

5.1.8 Velocidad de inclinación

Nuevamente la velocidad de inclinación es otra división sencilla, por ello sus pruebas son las mismas que para el tiempo y que para la velocidad tangencial.

Velocidad de inclinación			
ΔInclinación($^{\circ}$)	ΔTiempo(s)	Esperado($^{\circ}$/s)	Resultado($^{\circ}$/s)
4	2	2	2.0
625	25	25	25.0
-625	25	-25	-25.0
625	-25	-25	-25.0
-625	-25	25	25.0
5	2	2.5	2.5
2	5	0.4	0.4
630	25	25.2	25.2
2.5	2.5	1	1.0
2.5	5	0.5	0.5
0	2	0	0.0
5	0	50000000	5.0e7

Puesto que todos los resultados coinciden con los esperados, se dan por correctas todas las fórmulas implementadas.

5.2 El programa

Las pruebas con el programa han consistido en convertir un archivo CSV en un archivo JSON. Para verificar que se ha pasado la prueba se comprobaba que el documento era un JSON redactado correctamente y que los valores eran los correctos. Los CSV con los que se ha trabajado han sido generados a partir de recorridos reales del GPS iBlue 747A+. Puesto que dichos documentos contienen cientos de puntos, se ha comprobado que los datos sean correctos para los primeros y los últimos puntos. Además, como ya se ha comprobado que el cálculo de los nuevos datos es correcto se han comprobado únicamente aquellos datos que se copian del CSV.

5.2.1 Recorrido en línea recta

Este recorrido consiste en un recorrido de pocos kilómetros por una autopista de Madrid. El tramo capturado es una línea recta con una velocidad constante y sin cambios bruscos.

CSV de la Recta					
Index	Time	Latitude	Longitude	Heading	Speed
14117	20:02:33.400	40.3651892796	-3.9455537766	91.991	331.675232
14118	20:02:33.600	40.3652301773	-3.9455818614	91.781	332.415924
14229	20:02:55.800	40.3697427120	-3.9487390154	91.632	331.392242
14230	20:02:56.000	40.3697834377	-3.9487683917	91.520	331.679443
JSON de la Recta					
Index	Time	Latitude	Longitude	Heading	Speed
14117	"20:02:33.400"	40.3651892796	-3.9455537766	91.991	331.675232
14118	"20:02:33.600"	40.3652301773	-3.9455818614	91.781	332.415924
14229	"20:02:55.800"	40.3697427120	-3.9487390154	91.632	331.392242
14230	"20:02:56.000"	40.3697834377	-3.9487683917	91.52	331.679443

5.2.2 Recorrido con curvas

Este recorrido consiste en un recorrido por una carretera secundaria de Madrid. El tramo capturado está lleno de curvas cerradas y con muchos cambios de velocidad.

CSV de las Curvas					
Index	Time	Latitude	Longitude	Heading	Speed
31994	18:51:34.800	40.3967186384	-4.0442374889	27.173	280.991974
31995	18:51:35.600	40.3967323244	-4.0443165558	36.701	283.327820
32606	18:53:44.000	40.4251194713	-4.0564322262	51.204	22.708427
32607	18:53:44.400	40.4251626989	-4.0564019732	50.902	27.253201
JSON de las Curvas					
Index	Time	Latitude	Longitude	Heading	Speed
31994	"18:51:34.800"	40.3967186384	-4.0442374889	27.173	280.991974
31995	"18:51:35.600"	40.3967323244	-4.0443165558	36.701	283.327820
32606	"18:53:44.000"	40.4251194713	-4.0564322262	51.204	22.708427
32607	"18:53:44.400"	40.4251626989	-4.0564019732	50.902	27.253201

5.2.3 Recorrido en circuito

Este recorrido consiste en una vuelta al circuito de Jarama. El tramo capturado también tiene curvas cerradas y cambios de velocidad.

CSV del Circuito					
Index	Time	Latitude	Longitude	Heading	Speed
3091	14:39:37.800	40.6145842131	-3.5831080133	145.306	162.666275
3092	14:39:38.000	40.6145147263	-3.5830801603	144.557	162.687408
3851	14:42:16.000	40.6147108644	-3.5831534091	131.063	161.701309
3852	14:42:16.200	40.6146470373	-3.5831243671	130.551	161.298172
JSON del Circuito					
Index	Time	Latitude	Longitude	Heading	Speed
3091	"14:39:37.800"	40.6145842131	-3.5831080133	145.306	162.666275
3092	"14:39:38.000"	40.6145147263	-3.5830801603	144.557	162.687408
3851	"14:42:16.000"	40.6147108644	-3.5831534091	131.063	161.701309
3852	"14:42:16.200"	40.6146470373	-3.5831243671	130.551	161.298172

Puesto que el comienzo y el final del JSON coinciden con los del CSV y el formato es el correcto, se han dado por buenas las pruebas y, en consecuencia, el programa.

6

CONCLUSIÓN

Tras todo el trabajo realizado cabe destacar el juego que da un dispositivo GPS sencillo como con el que se ha trabajado. Es interesante ver cómo un GPS que ofrece una mínima información de un recorrido sirve para obtener tanta variedad de información. Además, los dispositivos GPS son una tecnología que está en constante evolución y en poco tiempo se podrá obtener una mayor cantidad de datos con una mayor precisión gracias a estos aparatos.


Con este proyecto se demuestra que la telemetría no es una tecnología cara que esté al alcance de sólo unos pocos. Cualquier piloto amateur puede permitirse comprar un GPS compatible con Teller y utilizar el programa para ver detalles de su conducción o mejorar en los próximos entrenamientos. Sin duda, esta aplicación es una gran ayuda para aquellos pilotos que no disponen de un equipo en condiciones.

Obviamente, este proyecto sólo es un comienzo. Teller podría mejorarse en un futuro en diferentes aspectos con el fin de ofrecer al piloto un servicio de mayor calidad. Las posibilidades de mejora son muchas, desde obtener nuevos datos que aún no se han considerado en el proyecto hasta realizar un filtrado mejor para el ruido.

Bibliografía

- [1] BT747. Gps data logger software - bt747. <http://www.bt747.com/>.
- [2] Fred Hebert. *Learn You Some Erlang for Great Good!* Editorial No Starch, 2013.
- [3] Ángel Herranz. Curvas, velocidad e inclinación. <http://contramanillar.wordpress.com/2011/10/23/curvas-velocidad-e-inclinacion/>.
- [4] Manuel R. Ortega Girón. Cinemática de la partícula. <http://www.uco.es/~falorgim/fisica/archivos/Lecciones/LFM04.PDF>.
- [5] TutorVista. Tangential acceleration. <http://physics.tutorvista.com/motion/tangential-acceleration.html>.
- [6] Wikipedia. Circunferencia osculatriz. http://es.wikipedia.org/wiki/Circunferencia_osculatriz.
- [7] Wikipedia. Telemetry. <http://en.wikipedia.org/wiki/Telemetry>.

Este documento esta firmado por

	Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
	Fecha/Hora	Fri Jun 06 21:37:34 CEST 2014
	Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
	Numero de Serie	630
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)