

SISTEMA PORTÁTIL DE BAJO COSTE PARA LA MEDICIÓN Y REPRESENTACIÓN DE PARÁMETROS CINEMÁTICOS EN 3D

J.F. Navarro-Iribarne

Dpto. de Informática y Automática, UNED, C/ Juan del Rosal 16, 28040 Madrid, fnavarro@gmail.com

D. Moreno-Salinas

Dpto. de Informática y Automática, UNED, C/ Juan del Rosal 16, 28040 Madrid, dmoreno@dia.uned.es

J. Sánchez-Moreno

Dpto. de Informática y Automática, UNED, C/ Juan del Rosal 16, 28040 Madrid, jsanchez@dia.uned.es

Resumen

El empleo de la monitorización para el análisis deportivo se ha convertido en una herramienta casi indispensable en el entrenamiento profesional de alto nivel. El poder conocer con exactitud los movimientos realizados por un deportista durante un entrenamiento aportan una gran ventaja con respecto al entrenamiento convencional, ya que si se conoce de forma teórica el movimiento que mejor rendimiento ofrece, el entrenador pretenderá que su deportista aproxime su movimiento real al teórico. Este artículo trata sobre el desarrollo de un sistema de bajo coste para la medición y representación de parámetros cinemáticos en 3D. Para la captura de los movimientos del deportista se ha empleado una unidad de medición inercial (IMU), cuyos datos serán procesados en una placa Arduino y mediante Bluetooth se enviarán los parámetros cinemáticos del movimiento a un teléfono móvil donde serán visualizados en forma de gráficas.

Palabras clave: IMU, Bluetooth LE, Arduino, cuaterniones, sistemas de control embebidos o empotrados.

1 INTRODUCCIÓN

Uno de los aliados del entrenamiento técnico es la tecnología. Un ejemplo concreto es la videocámara. Cuando un deportista quiere mejorar su técnica, es muy útil conocer qué es lo que hace mal para poder corregirlo. La videocámara permite grabar la secuencia del movimiento realizado y estudiarlo para mejorarlo [1]. Este sistema tiene el inconveniente de que sólo ofrece información visual, además de hacerlo en un plano de 2 dimensiones. Se han realizado algunos estudios sobre esta materia empleando 2 cámaras para obtener una representación tridimensional, pero los resultados no han sido muy favorables, ya que, si bien se capta el movimiento del

deportista, no existe una cuantificación precisa de los parámetros que intervienen en la mejora de la técnica [2]. Estos parámetros son principalmente dos: trayectoria y velocidad del movimiento del deportista. Son precisamente estos 2 parámetros los que el dispositivo desarrollado calculará para su posterior visualización.

En este trabajo se describe el desarrollo y funcionamiento del dispositivo que permite la captación de variables físicas durante el movimiento de un deportista para, posteriormente, visualizarlas de forma gráfica en un teléfono móvil. Se presenta un caso práctico que consiste en su empleo en una disciplina deportiva concreta: el lanzamiento de disco.

2 COMPONENTES DEL SISTEMA

2.1. Unidad de medición inercial (IMU)

Las unidades de medición inercial (IMU) son dispositivos utilizados para adquirir variables físicas de un cuerpo en movimiento empleando una serie de sensores. Estos sensores se dividen en acelerómetros, giroscopios y magnetómetros. Una IMU incorpora 3 acelerómetros, 3 giroscopios y 3 magnetómetros, uno por cada eje (X, Y, Z).

Dado que estos dispositivos solo se encargan de captar variables físicas, es necesario procesar toda esa información para determinar la orientación y velocidad del cuerpo a monitorizar. Una IMU suele estar ubicada en equipos portátiles, por lo que es usual el empleo de microcontroladores para la gestión de la información que éste aporta.

2.2. Algoritmo de fusión

El principal objetivo de los algoritmos de fusión de datos es el de mejorar la calidad de la información obtenida de procesos con múltiples sensores, conocidos como sinérgicos, es decir que actúan

conjuntamente para un objetivo común [3]. Existen varios métodos de filtrado para fusionar los datos del sensor. El utilizado en este trabajo es el desarrollado por Sebastian Madgwich [4], el cual emplea una representación de la orientación por medio de cuaterniones.

2.3. Cuaterniones

Con el fin de evitar el problema del *gimbal lock* en el dispositivo se trabaja con cuaterniones. Un cuaternión o cuaternio Q está constituido por cuatro componentes (q_0, q_1, q_2, q_3) que representan las coordenadas del cuaternio en una base $\{e, i, j, k\}$ [5]. Es frecuente denominar parte escalar del cuaternio a la componente en e (q_0), y parte vectorial al resto de componentes. De esta manera, un cuaternio se puede representar como

$$Q = [q_0, q_1, q_2, q_3] = [s, v] \tag{1}$$

donde s representa la parte escalar y v la vectorial. Para el empleo de los cuaterniones como metodología de representación de orientaciones se asocia el giro de un ángulo θ sobre el vector k al cuaternio definido por:

$$Q = rot(k, \theta) = \left(\cos \frac{\theta}{2}, k \cdot \sin \frac{\theta}{2} \right) \tag{2}$$

La composición de cuaterniones es tan sencilla como multiplicar cuaternios entre sí. De tal forma, que el resultado de rotar según el cuaternio Q_1 , para posteriormente rotar según Q_2 , es el mismo que el de rotar según Q_3 , obtenido por la expresión:

$$Q_3 = Q_2 \circ Q_1 \tag{3}$$

Es importante tener en cuenta que el producto de cuaterniones no es conmutativo. Se puede apreciar así que el empleo de cuaternios para la composición de rotaciones es un método computacionalmente muy práctico, pues basta con multiplicar cuaternios entre sí, lo que corresponde a una expresión de productos y sumas muy simples.

2.4. Bluetooth Low Energy (BLE)

Existen diversos protocolos wireless para uso en IOT (Internet Of Things), pero lo que hace que el BLE sea tan interesante es que es el más sencillo para implementar la comunicación entre pequeños dispositivos y una aplicación en cualquier plataforma móvil actual (iOS, Android, Windows Phone, etc.)

3 DESCRIPCIÓN DEL SISTEMA

La adquisición de los datos por parte del dispositivo (Figura 1) se basa en una placa Arduino Nano 33 BLE.

Esta placa incorpora, entre otros dispositivos, un sistema de medición inercial (IMU) para la toma de datos, un módulo de comunicación BLE para transferir las mediciones a un teléfono móvil y una batería que dota al dispositivo de la alimentación necesaria para su funcionamiento de forma portable. Todos estos componentes han sido ubicados en una carcasa fabricada para tal fin que será situada preferiblemente en la muñeca/tobillo del deportista.

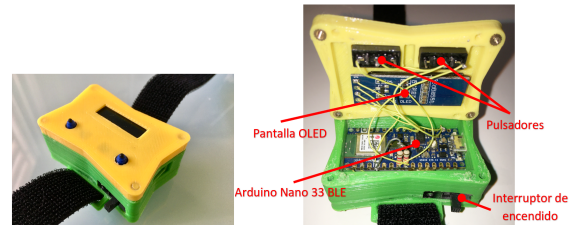


Figura 1: Imágenes del dispositivo portátil

Una vez concluida la fase de adquisición, procesado y envío de la información al teléfono móvil, el deportista podrá ver la trayectoria y velocidad del movimiento realizado en un teléfono.

4 METODOLOGÍA

El desarrollo software se ha realizado en dos entornos. Uno para la programación del microcontrolador de la placa Arduino que recibe datos de la IMU, los procesa y los envía al teléfono móvil, y otro para la programación del teléfono, que, tras recibir la información, la representará en forma de gráficas.

4.1 Programación del microcontrolador

El desarrollo del programa para el microcontrolador nRF52840 de la placa Arduino está escrito en C++ bajo el entorno *Visual Studio Code* empleando la extensión *PlatformIO*.

4.1.1 Estructura del programa

La elaboración del programa que recibe, procesa y envía la información de la IMU al teléfono se divide en dos bloques. Uno dedicado a la recepción y procesamiento de la información recibida de la IMU, y el otro al envío de la información ya procesada a través de BLE hacia el teléfono.

4.1.2 Procesamiento de la información de la IMU

La IMU empleada es la de la placa Arduino, en concreto se trata del chip LSM9DS1. Dispone de 9 grados de libertad y está conectado al microcontrolador mediante el protocolo I²C. Se ha utilizado para su programación la librería *Arduino_LSM9DS1.h*.

4.1.3 Estructura de control

El proceso de captura y procesado de la información se controla mediante una estructura de control “switch...case”, quedando el programa dividido en 4 fases. El objetivo de dividir el programa en varias fases es, además del orden y claridad de éste, permitir el mayor tiempo de muestreo posible en la fase de captura de datos de la IMU. Es importante que en la fase de captura el programa solo se dedique a leer información de la IMU y no gaste tiempo en tareas innecesarias. Las fases son:

- a) Fase I: Modo espera. Sucede cuando el dispositivo se pone en marcha mediante el interruptor de encendido.
- b) Fase II. Acondicionamiento. Dura 20 segundos que es el tiempo necesario para que los cálculos realizados por el algoritmo de fusión se estabilicen. Una vez concluida la temporización, y justo antes de que el programa pase a la siguiente fase, se acciona brevemente el motor vibrador durante 500 ms para indicar que la fase de captura va a comenzar.
- c) Fase III. Captura. Es la recogida de datos procedentes de la IMU. En este caso se realiza una captura de 3000 muestras.
- d) Fase IV. Procesado. Se realizan los cálculos para transformar las mediciones de la IMU obtenidas en la fase de captura (acelerómetro, giroscopio, magnetómetro) en velocidad y posición. Esta fase se divide en varias etapas:

I. Aplicación del algoritmo de fusión. Mediante el algoritmo de fusión todas las mediciones recogidas en un instante dado se emplean para mejorar la calidad de la información de salida. En la Figura 2 se muestra un esquema de su funcionamiento.

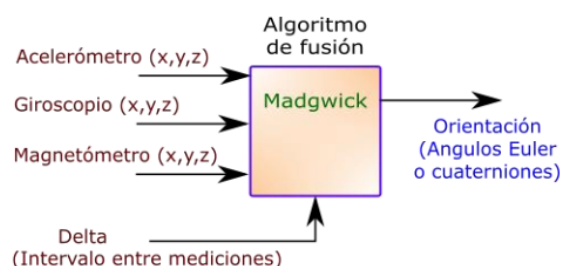


Figura 2: Algoritmo de fusión

II. Empleo de cuaterniones. Tal y como se ha indicado, se trabaja con cuaterniones para orientar la IMU.

III. Eliminación de la gravedad. Un acelerómetro está sujeto a aceleraciones dinámicas (a las que sometemos al dispositivo de manera voluntaria) y las estáticas

(fuerza de la gravedad). Es evidente que para conseguir unas mediciones correctas es necesario eliminar la aceleración estática. Con el cuaternión se calcula la dirección esperada de la gravedad y luego se resta de las lecturas del acelerómetro (Figura 3). El resultado es la aceleración dinámica [6].

```

g[0] = 2 * (q[1] * q[3] - q[0] * q[2]);
g[1] = 2 * (q[0] * q[1] + q[2] * q[3]);
g[2] = q[0] * q[0] - q[1] * q[1] - q[2] * q[2] + q[3] * q[3];

aT_x = 9.80665 * (ax - g[0]);
aT_y = 9.80665 * (ay - g[1]);
aT_z = 9.80665 * (az + g[2]);
    
```

Figura 3: Empleo de cuaterniones

IV. Matriz de rotación. Si la IMU se desplazara con sus ejes paralelos a los ejes de nuestro sistema de referencia, los cálculos para determinar su posición serían bastantes sencillos. Pero eso nunca va a ocurrir, porque la IMU siempre tendrá alguno de sus ejes (sistema de referencia de la IMU) rotado con respecto al nuestro (sistema de referencia de la Tierra). En la Figura 4 se muestra un ejemplo de los dos sistemas de referencia.

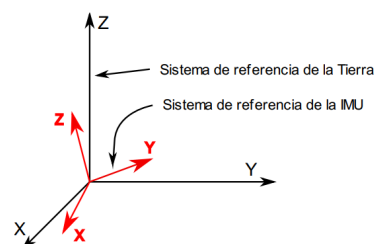


Figura 4: Rotación en 2 sistemas distintos

Para llevar el sistema de referencia de la IMU al de la tierra, es necesario rotarlo. Para ello se emplea la matriz de rotación sobre los cuaterniones mostrada en (4) [7].

$$R(q) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (4)$$

El valor de las aceleraciones llevadas al sistema de referencia de la Tierra se calcula multiplicando la inversa de la matriz de rotación generada con los cuaterniones por las aceleraciones a las que está sometido el dispositivo. De esta forma la aceleración gravitacional queda eliminada.

V. Filtrado de la aceleración. La señal de la aceleración obtenida por la IMU es una señal con bastante ruido y algún *glitch* que puede generar errores en los cálculos posteriores. Para obtener una señal más suavizada se aplica un filtro de reducción de ruido del tipo EWMA (*Exponentially Weighted Moving Average*) [8]. Este filtro tiene un bajo coste computacional y se basa en tomar N muestras,

sumarlas y dividir las por N. El promedio es "móvil" cuando se recalcula cada vez que se obtiene una nueva muestra tomando las N muestras anteriores. Este filtrado se emplea usando la librería "ewma.h" para Arduino. En las Figuras 5 y 6 se muestran una señal en bruto y otra suavizada respectivamente.

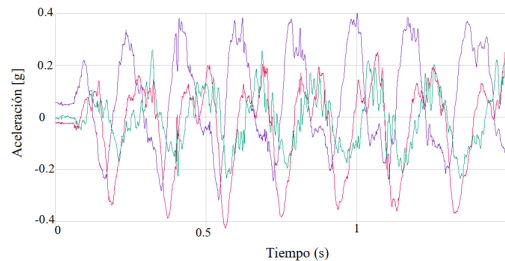


Figura 5: Aceleraciones sin suavizar

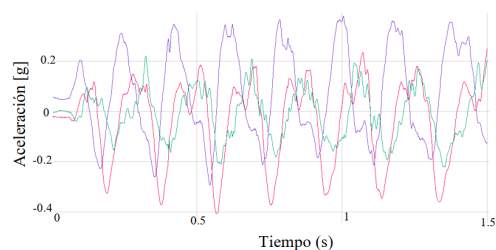


Figura 6: Aceleraciones suavizadas

VI. Cálculo de la velocidad. Empleando las aceleraciones filtradas se calcula la velocidad instantánea en cada punto mediante la integración de dichas aceleraciones. El cálculo integral se realiza de manera que se divide el área bajo la curva de la aceleración en pequeños rectángulos y se suma el área de todos ellos. El resultado de esta operación es la velocidad [9]. Esta operación para calcular la velocidad se realiza para cada uno de los ejes (X, Y, Z).

VII. Filtrado DC. El cálculo de la velocidad a partir de la integración de la aceleración conlleva un problema. Dado que el resultado final se calcula como la suma acumulativa de áreas parciales, si en cualquiera de estas áreas hay un error, provocado, por ejemplo, por ruido en la señal, el error afectará al resultado final. Si el ruido origina muchos errores en diversas áreas, el resultado diferirá bastante del esperado.

Este hecho se aprecia fácilmente cuando se integra la aceleración de un móvil para calcular su velocidad. Si se monitoriza un movimiento del tipo paro-marcha-paro, es posible que la velocidad final calculada no sea exactamente 0 (aunque el móvil realmente esté parado). Si no se corrigen o minimizan estos errores, en los posteriores cálculos de las posiciones, los resultados serían bastantes incoherentes. Una forma de reducir este problema es eliminar la componente continua de la señal de la velocidad obtenida.

La eliminación de la componente continua se consigue aplicando un filtro paso alto a la señal de la velocidad. El valor de la frecuencia de corte ha sido determinado de forma empírica. Es conveniente que la frecuencia de corte no sea muy alta, ya que esto provoca la eliminación total de la componente continua, anulando cualquier posible representación del desplazamiento del dispositivo. Para la aplicación del filtro paso alto se ha empleado la librería "filters.h" para Arduino.

En las gráficas de las Figuras 7 y 8 se aprecia la efectividad de este filtro, donde se muestra la velocidad del eje X, tanto sin filtrar como filtrada.

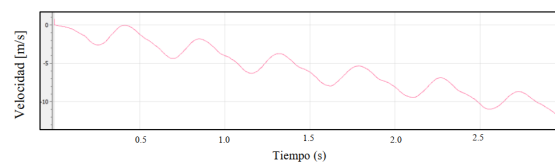


Figura 7: Velocidad en eje X sin filtrar

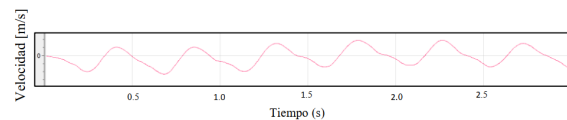


Figura 8: Velocidad en eje X filtrada

VIII. Aproximación a 0 de la velocidad. La aceleración leída por la IMU, aunque ésta se encuentre totalmente detenida, nunca será 0, sino que tiene un pequeño valor. Este valor, por pequeño que sea, termina afectando a la velocidad calculada.

Con el fin de que la velocidad calculada sea 0 cuando el dispositivo se encuentre prácticamente detenido, se aplica un pequeño ajuste. Si la aceleración medida se encuentra dentro de una ventana establecida durante un número de muestras consecutivas también establecido, la velocidad durante todo ese periodo valdrá 0 (Figura 9).

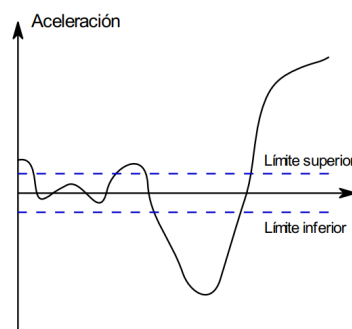


Figura 9: Límites de la aceleración

En el caso de este dispositivo, los límites de la ventana son +1.0 y -1.0, y el número de muestras es de 60. En las Figuras 10 y 11 se muestra el funcionamiento de este algoritmo.

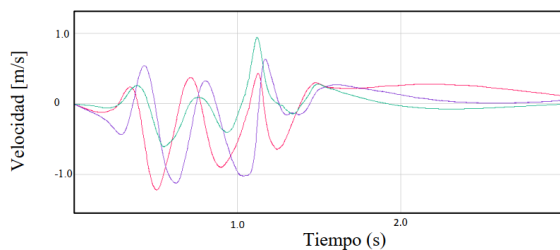


Figura 10: Velocidad sin ajuste

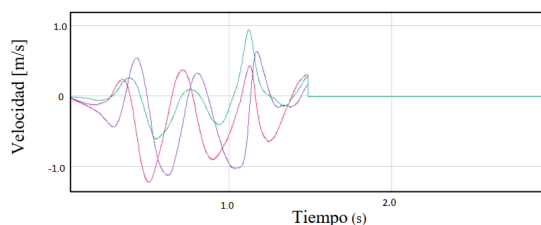


Figura 11: Velocidad ajustada

IX. Cálculo del módulo de la velocidad. En el paso previo se ha calculado la velocidad de cada uno de los componentes, así que el valor del módulo de la velocidad para cada instante es:

$$V(t) = \sqrt{Vx(t)^2 + Vy(t)^2 + Vz(t)^2} \quad (5)$$

X. Cálculo de las posiciones. Al igual que se hizo para las velocidades a partir de las aceleraciones, las posiciones se calculan mediante la integración de la velocidad. Este proceso se realiza en cada instante y para cada eje.

Fase V. Espera de la conexión Bluetooth. La siguiente fase tras el proceso de cálculo del desplazamiento es el envío mediante BLE. Pero antes de realizar el envío es necesario configurar la comunicación.

4.1.4 Configuración BLE

a) Esperando la conexión de la central. El programa quedará detenido en este punto hasta que el teléfono móvil no establezca comunicación con el dispositivo.

b) Empaquetado de datos. Una de las limitaciones a tener en cuenta con el BLE Energy es que solo permite la transmisión de 20 bytes en cada trama. Con el fin de aprovechar al máximo cada trama se ha realizado el empaquetado de la Tabla 3.

Tabla 1. Empaquetado de datos

Dato	Posición X					Posición Y					Posición Z					Velocidad				
Bytes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Formato	x	x	.	x	x	x	x	.	x	x	x	x	.	x	x	x	x	.	x	x

c) Envío de datos al teléfono móvil. Las tramas son enviadas al teléfono móvil según se van generando. El total de tramas enviadas se corresponde con el número de muestras tomadas por la IMU, en este caso 3000.

Cada vez que se recibe un paquete de datos desde el dispositivo, se está recibiendo la trama de datos de 20 bytes que formamos previamente. Mediante un simple acotado de ésta, se separan y almacenan en arrays independientes cada una de las 4 variables recibidas. El código que realiza esta operación se muestra en la Figura 12.

```
# La trama recibida de 20 bytes se divide en las variables de posición y velocidad
pos_x[index_x] = c.value[:5] # Parseo de la posición en X
float(pos_x[index_x]) # Se convierte a float la cadena recibida
pos_y[index_y] = c.value[5:10] # Parseo de la posición en X
float(pos_y[index_y]) # Se convierte a float la cadena recibida
pos_z[index_z] = c.value[10:15] # Parseo de la posición en X
float(pos_z[index_z]) # Se convierte a float la cadena recibida
velocidad[index_v] = c.value[15:20] # Parseo de la posición en X
float(velocidad[index_v]) # Se convierte a float la cadena recibida
```

Figura 12. Descomposición de la trama de 20 bytes

4.2 Programación del teléfono móvil

El programa encargado de recibir la información enviada por el dispositivo vía BLE y representar de manera gráfica el desplazamiento y velocidad, ha sido escrito en Python. Para los dispositivos iOS existe un entorno de programación llamado *Pythonista*, que permite la edición y ejecución de cualquier programa escrito en Python. En concreto, este programa ha sido escrito y ejecutado en un iPhone 6S. El programa se divide en 2 bloques.

4.2.1 Recepción de datos

El teléfono móvil hace de central en la transmisión BLE. Por este motivo ha de hacer un escaneo de los dispositivos visibles (periféricos) que están a su alcance y emparejarse con uno en concreto, nuestro dispositivo, que lo conocerá mediante el nombre que le fue asignado, “*Arduino*”. Posteriormente, la central ha de buscar el servicio con el UUID (Universally Unique Identifier) asignado en nuestro dispositivo y una vez encontrado, buscará la característica cuyo UUID también fue establecido en el dispositivo.

Cada vez que se recibe un paquete de datos desde el dispositivo, se está recibiendo la trama de datos de 20 bytes que se formó previamente según la Tabla 3.

4.2.2 Representación gráfica

Para la representación gráfica se ha empleado el módulo denominado “*matplotlib*”. Este módulo permite la representación de valores almacenados en arrays tanto en 2D como en 3D.

Para representar la trayectoria seguida por el dispositivo se utiliza una gráfica en 3D. El programa muestra la misma trayectoria desde distintos puntos de vista, con el fin de poder seleccionar y guardar la que ofrezca una mejor perspectiva. El comienzo del movimiento está en el punto (0,0,0). En la Figura 13 se muestran dos series de representaciones del mismo movimiento.

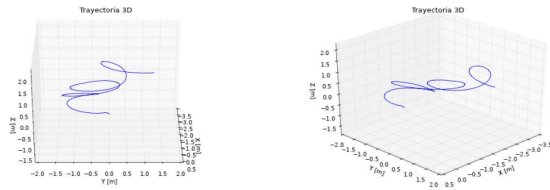


Figura 13: Representación de la trayectoria en 3D

También se representa la trayectoria del movimiento en 2 dimensiones, para que sea más fácil su visualización. En concreto se visualiza en una vista superior (plano XY) (Figura 14) y en una vista lateral (plano YZ) (Figura 15).

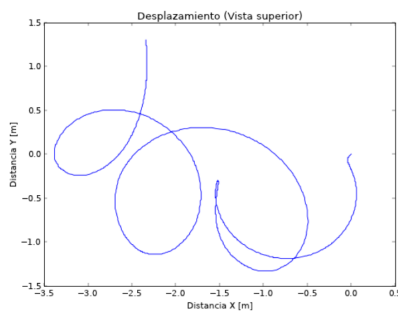


Figura 14: Vista superior del movimiento

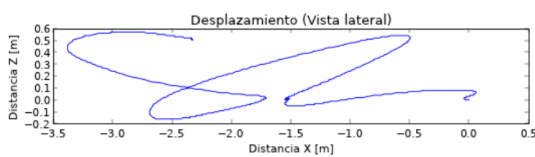


Figura 15: Vista lateral del movimiento

Para representar la velocidad se utiliza una gráfica en 2D expresada en m/s.

5 MODO DE EMPLEO

5.1 Colocación del dispositivo

El deportista se ha de sujetar con firmeza el dispositivo empleando la cinta que incorpora. Si no queda bien sujeto se pueden originar vibraciones que generen lecturas incorrectas en la IMU (Figura 16).

5.2 Encendido

Es necesario mover el interruptor situado en el lateral hacia la derecha (ver Figura 2). Al encenderlo aparece en la pantalla el texto MOLADIS.

5.3 Inicio monitorización

Para que la IMU comience la captura de datos se pulsa el botón de inicio del dispositivo.



Figura 16: Botón de inicio

En la pantalla aparece el texto “Estabilización” junto a una cuenta atrás de 20 segundos. Este valor numérico es el tiempo de estabilización que necesita el algoritmo de fusión.

Una vez terminada la cuenta atrás, el dispositivo realizará una pequeña vibración para indicar que comienza la adquisición de datos por la IMU, además de aparecer en la pantalla el texto “Capturando movimiento”. En este momento, el deportista debe comenzar a realizar el movimiento que desea monitorizar. El proceso de captura de datos tiene una duración de 6 segundos y en ese transcurso recoge 3000 mediciones.

Cuando termina, el dispositivo realiza otra vibración con un patrón distinto al anterior para indicar que ha finalizado la captura de datos. Una vez finalizada la captura, el microcontrolador procesa toda la información recogida. Durante esta fase aparece en la pantalla el texto “Calculando velocidad y posiciones”.

La fase siguiente es el envío de los datos desde Arduino al teléfono móvil. Esta fase se indica en la pantalla con el texto “Esperando conexión Bluetooth”. Cuando aparece este mensaje es el momento de ejecutar el programa de recepción y visualización de datos en el teléfono móvil.

En el teléfono móvil es necesario ejecutar el programa creado para recibir la información. Para ello se pulsa el botón ejecutar dentro del programa Pythonista (Figura 17).

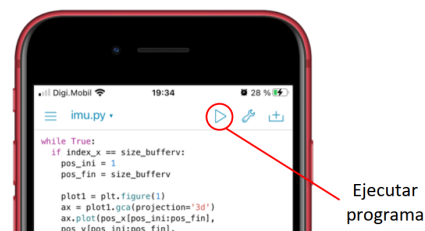


Figura 17: Ejecutar programa

El programa se ejecuta en modo consola. Si la conexión con el dispositivo se realiza de manera correcta, comienza la transmisión de los datos que son

recibidos por el teléfono. Cuando la información es recibida e interpretada por el programa, van apareciendo las gráficas generadas, según se muestra en las capturas de la Figura 18.

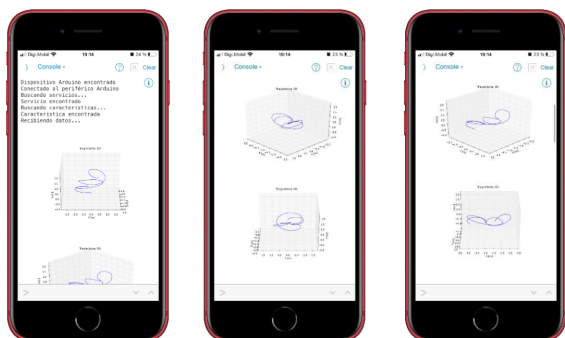


Figura 18: Recibiendo información del dispositivo

Cuando ha terminado el proceso, en el dispositivo se muestra el mensaje “Transmisión finalizada”. Llegado a este punto es posible volver a realizar otra captura pulsando el botón derecho del dispositivo. Aparecerá de nuevo una cuenta atrás para que el deportista se vaya preparando y el ciclo volverá a comenzar.

6 VALIDACIÓN DEL SISTEMA

El empleo de este dispositivo para la monitorización deportiva es válido para todos aquellos deportes en los que se efectúa un movimiento concreto y repetitivo. Por ejemplo, se podría emplear para monitorizar un golpeo de balón durante el saque en voleibol o en diversas disciplinas de lanzamiento en atletismo. En concreto, el chequeo de este dispositivo se ha realizado para un lanzamiento de disco.

6.1 Simulación de lanzamiento

Tras efectuar la simulación del lanzamiento de disco, en el teléfono móvil se generan una serie de gráficas. Entre las gráficas, las correspondientes a la trayectoria 3D son las que permiten visualizar de una manera más adecuada el movimiento. Las gráficas del movimiento en vista superior, lateral y velocidad se muestran en las Figura 19 y 20.

En la gráfica de la velocidad se observan 3 picos de velocidad bastante definidos. El primero de ellos se corresponde con el movimiento de medio arco de circunferencia realizado en sentido opuesto. El algoritmo de “Aproximación a 0 de la velocidad” detectó correctamente el cambio de sentido (ya que durante un instante la aceleración en este punto se aproxima a 0), forzando a que la velocidad fuese 0 en este punto. Como consecuencia de esto, el cálculo de la trayectoria es muy preciso y se genera el cambio de sentido que se observa en la vista aérea.

La trayectoria realizada, la distancia de los radios de giro (visualizado en la vista aérea) y la de la elevación del brazo durante el lanzamiento (vista lateral) son muy precisos (Figura 19). En la gráfica de la velocidad (Figura 20) se observa como el valor más alto coincide con el momento exacto del lanzamiento del disco.

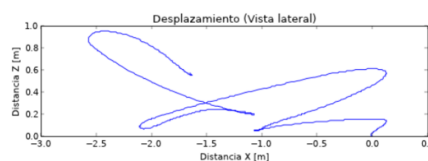
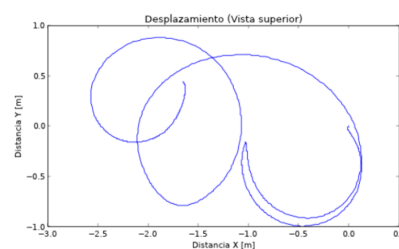
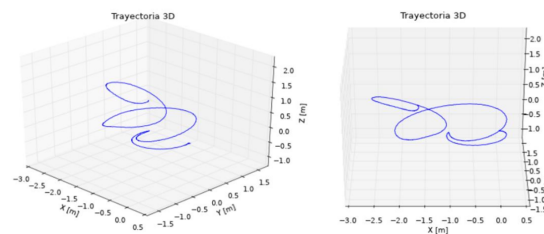


Figura 19: Distintas perspectivas de la trayectoria

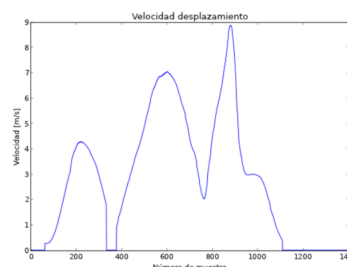


Figura 20. Velocidad del movimiento

7 CONCLUSIONES

En este trabajo se ha desarrollado un dispositivo que es capaz de captar las variables físicas originadas por el movimiento de un deportista, y posteriormente representar la trayectoria de éste en un teléfono móvil. Las pruebas realizadas para su verificación, a pesar de no poder ser realizadas por deportistas profesionales en un entorno real, han obtenido en general buenos resultados.

Como conclusión general se puede establecer que el empleo de este tipo de dispositivos para representar el desplazamiento de un móvil conlleva una serie de

limitaciones que han de tenerse en cuenta a la hora de validar los resultados obtenidos. La principal limitación es que no se pueden realizar registros prolongados en el tiempo sin algún mecanismo de reseteo (*dead reckoning*) [10], o bien, tener un elemento externo (señal GPS, visión artificial) para ir corrigiendo los errores en las medidas. Sin estos mecanismos, la deriva en la medida provocada por el giroscopio y/o el error acumulado durante la integración, primero de la aceleración y después de la velocidad, harían inservibles la adquisición de un largo trayecto. Para este trabajo, al tratarse de adquisiciones de movimientos cortos, no es necesario emplear mecanismos auxiliares.

Agradecimientos

Este trabajo fue financiado en parte por el Ministerio de Ciencia e Innovación con los proyectos PID2020-112658RBI00/10.13039/501100011033 y PID2020-112502RB-C44 y por la UNED a través del proyecto 096-034095 (ILTraDos).

English summary

LOW-COST PORTABLE SYSTEM FOR MEASUREMENT AND REPRESENTATION OF KINEMATIC PARAMETERS IN 3D

Abstract

The use of monitoring for sports analysis has become another tool in high-level professional training. Being able to know exactly the movements performed by an athlete during training provides an advantage over conventional training, since if the movement that offers the best performance is known theoretically, the trainer will expect his athlete to approximate his movement real to theoretical. This article deals with the development of a low-cost system for the measurement and representation of kinematic parameters in 3D. To capture the athlete's movements, an inertial measurement unit has been used, whose data will be processed in an Arduino-based microcontroller and via Bluetooth the kinematic parameters of the movement will be sent to a mobile phone where they will be displayed in form of graphs.

Keywords: IMU, Bluetooth LE, Arduino, cuaternions, embedded control systems.

Referencias

- [1] Bartlett, R. M. (1992). The biomechanics of the discus throw: A review. *J Sports Sci*, 10(5), 467-510.
- [2] Finanger, K.E. (1969). An EMG study of the function of selected muscles involved in the throwing of the discus. Unpublished doctoral dissertation, University of Iowa.
- [3] Chérigo, C., Rodríguez, H. (2017). Evaluación de algoritmos de fusión de datos para estimación de la orientación de vehículos aéreos no tripulados. *Proceedings of the 15th LACCEI International Multi-Conference for Engineering, Education and Technology*, Boca Raton, FL, USA.
- [4] Madgwick, S. (2010). An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Internal report.
- [5] Barrientos, A., Peñín, L.F., Balaguer, C., Aracil, R. (2007). *Fundamentos de robótica*. Mc Graw Hill.
- [6] Vareano, F. (2011). An EMG study of the function of selected muscles involved in the throwing of the discus. *MoS Thesis*.
- [7] Kong, X. (2004). INS algorithm using quaternion model for low cost IMU. *Robotics and Autonomous Systems* 46, 221–246.
- [8] Verrastro, C., Gómez, J.C., Folino, P., Alberino, S. (2009). Control PID con filtro dinámico de promedios móviles ponderados exponencialmente. *XIII Reunión de Trabajo en Procesamiento de la Información y Control*.
- [9] Olivares Garcés, D. (2018). Estudio e implementación de algoritmos para la estimación de la posición mediante sistemas inerciales con arduino. *Universidad Politécnica de Valencia*.
- [11] H. Yin, H. Guo, X. Deng. (2014). Pedestrian dead reckoning indoor positioning with step detection based on foot-mounted IMU. *Proceedings of the 2014 International Technical Meeting of The Institute of Navigation*, San Diego, CA, USA.



© 2022 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC-BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0>)