

Relazione progetto di sensori

Ronzoni Simone - Papallazi Xhorxho

1 Introduzione

Si propone in questo breve articolo lo sviluppo di un controller che si comporti come un controller per un headset VR attraverso l'uso della piattaforma Arduino Nano 33 BLE Sense. A tale scopo si è provato ad utilizzare vari sensori di questa piattaforma per stimare orientamento e spostamento della stessa in uno spazio tridimensionale, per permettere il controllo tramite gesture o pulsanti e dare un feedback all'utente attraverso i led integrati.

Il lavoro è stato svolto in varie fasi:

- Analisi delle prestazioni di alcuni algoritmi per la stima dell'orientamento e dello spostamento
- Calibrazione dei sensore IMU
- Integrazione delle funzionalità di Bluetooth Low Energy
- Assemblaggio di un piccolo circuito con 2 push button per permettere l'interazione con l'utente

2 Orientation tracking

Per la stima dell'orientamento sulle piattaforme Arduino™ si utilizzano solitamente dei filtri complementari che sono computazionalmente più leggeri e hanno convergenza più veloce rispetto ad un filtro di Kalman. Questi filtri teoricamente sono meno accurati di un filtro di Kalman ma per sistemi con dinamiche veloci possono andare bene. In letteratura si possono trovare due filtri in particolare:

- Madgwick (scelta se si utilizza il magnetometro)¹: viene inizializzato con un parametro β che rappresenta quanto le misure di accelerometro e magnetometro vanno a correggere la stima dell'orientamento dalle misure del giroscopio per rimuovere i disturbi da traslazioni e distorsioni magnetiche.
Sono stati provati vari valori di β rispetto al valore predefinito 0.1: riducendo questo valore la stima dell'orientamento risulta più precisa ma meno reattiva e la velocità di convergenza rispetto alla posizione reale subisce molti ritardi. Provando vari valori, anche ricercandoli in altre applicazioni con la stessa piattaforma², si è trovato in 0.615 un buon compromesso per ritardo e precisione della stima;
- Mahony (scelta se non si utilizza il magnetometro)³: viene inizializzato con due parametri che rappresentano il fattore proporzionale e integrativo dello schema di sensor fusion.

È presente una libreria⁴ che mette a disposizione le implementazioni di questi due filtri che restituiscono l'orientamento della piattaforma rispetto al sistema di riferimento della terra.

La piattaforma utilizzata mette a disposizione un IMU a 9 gradi di libertà⁵ con accelerometro, giroscopio e magnetometro integrati in esso.

2.1 Osservazioni

Si riportano di seguito una lista di osservazioni:

- La scelta di β influenza il funzionamento della stima dello spostamento in quanto se la stima dell'orientamento non è abbastanza veloce, la rimozione della componente gravitazionale non è efficace e porta a spostamenti anche semplicemente inclinando la piattaforma. Questo potrà sicuramente essere oggetto di studio per un possibile miglioramento ma per ora non si è fatto nulla a riguardo;
- Utilizzare le misurazioni del magnetometro non sempre migliora la stima, viste le interferenze di altri campi magnetici (nelle prove fatte anche tenendo la piattaforma accanto ad un laptop si avevano delle differenze) che risulta invece buona anche utilizzando solamente i dati dall'accelerometro e dal giroscopio, previa calibrazione;
- La frequenza di aggiornamento della funzione *loop* con la stima dell'orientamento è di circa 100Hz mentre potremo vedere poi come le frequenze di aggiornamento dei sensori sono molto più alte.

¹[Madgwick](#)

²[Reefwing AHRS](#)

³[Mahony](#)

⁴[SensorFusion](#)

⁵[LSM9DS1](#)

3 Motion tracking

Per stimare lo spostamento è stato utilizzato l'accelerometro all'interno dell'IMU. L'accelerometro misura le accelerazioni lineari proprie della piattaforma e dalle sue misure integrando due volte si può ottenere una stima dello spostamento percorso dalla piattaforma stessa. Sono note le seguenti relazioni:

$$a(t) = \frac{dv(t)}{dt},$$

$$v(t) = \frac{ds(t)}{dt}$$

Da cui si ottengono le seguenti relazioni:

$$\Delta v = v(t_2) - v(t_1) = \int_{t_1}^{t_2} a(t) dt,$$

$$\Delta s = s(t_2) - s(t_1) = \int_{t_1}^{t_2} v(t) dt$$

La piattaforma invierà le variazioni di spostamento e non lo spostamento totale. L'integrazione dell'accelerazione avviene considerando l'accelerazione costante nell'intervallo di misura oppure usando la tecnica di quadratura del trapezio:

$$v(t) = (a(t-1) + a(t)) * \frac{h}{2}$$

Dove:

- $a(t-1)$ è l'accelerazione misurata nel loop precedente;
- a è l'accelerazione misurata nel loop corrente;
- h il passo di discretizzazione pari a $\frac{1000}{sampleFreq}$.

Una volta ottenuta la velocità da questa prima integrazione, possiamo ottenere lo spostamento utilizzando le stesse formule di quadratura che si possono usare per trovare la velocità. In particolare per quella del trapezio risulta ancora:

$$s(t) = (v(t-1) + v(t)) * \frac{h}{2}$$

3.1 Osservazioni

Si riporta di seguito una lista di osservazioni:

- Prima di procedere con il calcolo dello spostamento bisogna rimuovere la componente gravitazionale, che viene sempre misurata da un accelerometro, dalle misure di accelerazione. Utilizzando la stima dell'orientamento della piattaforma rispetto al sistema di riferimento terra si calcola la matrice di rotazione in modo da poter ruotare il vettore delle accelerazioni misurate e sottrarvi il vettore $\vec{g} = [0, 0, 9.81]$;
- Per cercare di migliorare la stima dello spostamento è stato applicato un filtraggio passa-basso alle misure dell'accelerometro per provare ad eliminare il rumore nella misure usando una libreria per arduino⁶;
- Integrando nei modi descritti, le misure di spostamento risultavano essere molto basse quindi sono state denormalizzate le misure di accelerazione che prima erano normalizzate rispetto all'accelerazione di gravità e le stime di spostamento sono state moltiplicate per un fattore di scala;

⁶[LibFilter](#)

- Se si ruota la piattaforma troppo velocemente si deve attendere che la stima del filtro converga: con il parametro β scelto il tempo massimo di convergenza è di 2 secondi e in questo intervallo tutti gli spostamenti sono sbagliati visto che l'orientamento della piattaforma è sbagliato e quindi la rimozione della componente gravitazionale comporta uno spostamento;
- Si è cercato di mimare la misura di spostamento di un sensore per realtà aumentata ma studiando il loro funzionamento si è scoperto che le misure di spostamento sono effettuate facendo un tracking, tramite una o più fotocamere, di emettitori luminosi posti sui controller (il sensore VR di PlayStation™ ad esempio utilizza due luci visibili e una fotocamera⁷).

⁷PlayStation VR Tracking

4 Calibrazione

Durante le prove per la misura dell'orientamento e dello spostamento si è dovuto affrontare un processo di calibrazione dei sensori all'interno dell'IMU che presentavano una deviazione dai valori reali. Per fare ciò è stata utilizzata una libreria⁸, che si occupa anche dell'interfacciamento per il setup e la lettura dei dati dai sensore, per poter calibrare accelerometro, giroscopio e magnetometro.

Sono state impostate le frequenze di campionamento dei sensori come segue:

- 476Hz per l'accelerometro e il giroscopio;
- 400Hz per il magnetometro visto che le frequenze utilizzabili per questo non sono uguali a quelle utilizzabili per gli altri sensori.

Sono stati inoltre impostati i valori di fondo scala dei sensori come segue:

- $\pm 4g$ per l'accelerometro;
- $\pm 1000 \frac{\text{deg}}{\text{s}}$ per il giroscopio;
- $\pm 1200 \mu\text{T}$ per il magnetometro.

Mentre la calibrazione dell'accelerometro non è particolarmente importante ai fini della stima dell'orientamento aiuta nella stima dello spostamento. È vitale invece la calibrazione del giroscopio e del magnetometro in quanto:

- per la stima dell'orientamento bisogna integrare le misure di velocità angolari del giroscopio ma quindi ad ogni ciclo di misura si sommano gli errori che anche se piccoli ad un certo punto saturano la misura;
- senza le misure del magnetometro si ha una rotazione continua, spesso anche veloce, sull'asse z della piattaforma quindi è necessaria una corretta misura dal magnetometro.

La libreria utilizzata permette la calibrazione di due parametri:

- offset: la deviazione dal valore fisico di zero quando la piattaforma è ferma ma viene comunque misurato del movimento;
- slope: un fattore che compensa la sensitività dei trasduttori visto che “*non tutto il silicio è uguale*“ e il trasduttore della quantità fisica non sempre ha la sensitività teorica.

La calibrazione di accelerometro e giroscopio è abbastanza semplice e diretta in quanto basta calibrare in ogni verso i vari assi dei sensori raccogliendo un certo numero di misurazioni e calcolando i due fattori.

Per la calibrazione del magnetometro bisogna andare a capire come è orientato il campo magnetico terrestre nel punto in cui stiamo calibrando il sensore e poi anche per questo calibrare i vari assi. Per poter capire come è orientato il campo magnetico terrestre si possono utilizzare vari siti online che restituiscono i valori dell'angolo di declinazione e di quello di inclinazione del campo magnetico terrestre in base alle proprie coordinate geografiche di latitudine, longitudine ed elevazione rispetto al livello del mare⁹. In questo caso il sensore è stato calibrato a Brusaporto. (Lat 45° 40' 3" N, Lon 9° 46' 2" E, 255.0 m Mean Sea Level)

⁸[LSM9DS1 library](#)

⁹[National Center For Environmental Information](#)

5 Interazione con l'utente

Per permette l'interazione con l'utente si è pensato a due modi:

- gestures;
- pulsanti.

5.1 Riconoscimento delle gestures

Grazie al sensore APDS9960 è possibile implementare un sistema di riconoscimento delle gesture. Tale sensore è costituito da quattro fotodiodi che captano un segnale infrarosso emesso da un trasmittitore. Quando avviene un movimento, il segnale infrarosso viene riflesso indietro al sensore che capta direzione e velocità, in un range dai 10 ai 20 cm.

L'obiettivo è riuscire a determinare la direzione lungo quattro direzioni cardinali del movimento del pollice sopra il sensore mentre si impugna il controller costituito dall'Arduino.

Esiste la libreria *Arduino_APDS9960* che integra già tale funzionalità con ottimi risultati: in particolare è in grado di determinare la direzione del movimento rispetto al sistema di riferimento dell'Arduino. L'unico parametro configurabile è il livello di sensibilità su una scala da 1 a 100, con valori alti che rendono il riconoscimento più sensibile ma meno accurato (di default è 80 se non specificato). Sperimentalmente si è osservato come un valore pari a 60 sia più adatto al contesto in questione, riducendo la possibilità di false letture.

```
1 #include <Arduino_APDS9960.h>
2
3 void setup() {
4     Serial.begin(9600);
5
6     while (!Serial);
7     if (!APDS.begin()) {
8         Serial.println("Error initializing APDS9960 sensor!");
9     }
10    APDS.setGestureSensitivity(60);
11    Serial.println("Detecting gestures ...");
12}
13 void loop() {
14     if (APDS.gestureAvailable()) {
15         // a gesture was detected, read and print to serial monitor
16         int gesture = APDS.readGesture();
17         switch (gesture) {
18             case GESTURE_UP:
19                 Serial.println("Detected UP gesture");
20                 break;
21             case GESTURE_DOWN:
22                 Serial.println("Detected DOWN gesture");
23                 break;
24             case GESTURE_LEFT:
25                 Serial.println("Detected LEFT gesture");
26                 break;
27             case GESTURE_RIGHT:
28                 Serial.println("Detected RIGHT gesture");
29                 break;
30             default:
31                 break;
32         }
33     }
34 }
```

Listing 1: Metodo setup plugin

Tuttavia può essere necessario personalizzare e calibrare il riconoscimento delle gesture in base ad uno scenario e utilizzo particolare. Questo è possibile utilizzando modelli di Machine Learning con TinyML.

5.2 Pulsanti

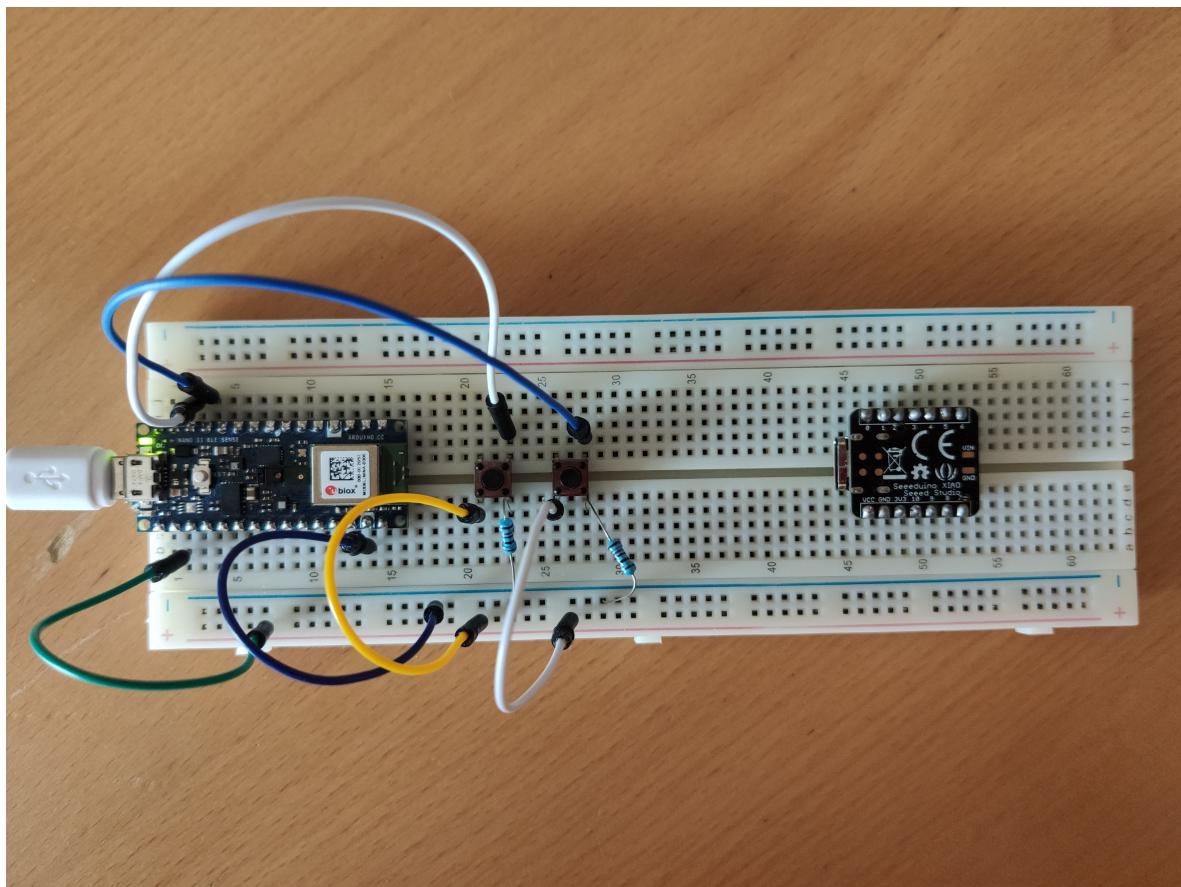
È stato progettato un piccolo circuito assemblando su una breadboard con due pulsanti e scritto poi un codice per la piattaforma che in base ai pulsanti premuti accendesse di un colore diverso il led nelle seguenti combinazioni:

- pulsante 1 premuto: led di colore verde;
- pulsante 2 premuto: led di colore blu;
- entrambi i pulsanti premuti: led di colore rosso.

La combinazione di entrambi i pulsanti premuti è stata pensata per un eventuale sviluppo futuro in cui tramite questa configurazione la piattaforma entra in uno stato di calibrazione per memorizzare la posizione in cui vogliamo punti il sensore.

Si potrebbe pensare di aggiungere anche un joystick analogico ma vista la frequenza di aggiornamento della piattaforma con gli algoritmi stima dell'orientamento e posizione (100Hz) sarebbe meglio utilizzare una seconda scheda che si occupi solo di pulsante e joystick e connessa alla piattaforma Arduino Nano tramite I2C per la gestione della calibrazione con interrupt.

Nell'immagine viene mostrato il circuito e una possibile seconda scheda che si potrebbe utilizzare (una piccola scheda Seeeduino), senza però il collegamento I2C.



6 Invio dei dati tramite Bluetooth Low Energy

Per l'invio dei dati si è deciso di utilizzare il bluetooth low energy di cui ne è dotato l'Arduino Nano BLE grazie al modulo NINA-B306.

Supporta il bluetooth 5.0 a 2.4 GHz con 40 canali. La capacità di invio può variare da 125 kbps fino a 1Mbps.

Per l'invio di tutte le informazioni di interesse, si è deciso di adottare l'approccio di edge computing, effettuando tutte le elaborazioni possibili su Arduino e inviare i dati già pronti per essere utilizzati con i seguenti vantaggi:

- Minore latenza: elaborando i dati direttamente sul dispositivo si eliminano i tempi di latenza di un'eventuale elaborazione remota.
- Minore banda di rete richiesta: vengono inviati solamente i dati strettamente necessari nel formato richiesto.
- Maggiore sicurezza: poiché i dati sono elaborati direttamente nel nodo che li raccoglie, non c'è rischio di perdita o compromissione dei dati quando inviati per l'elaborazione.
- Maggiore affidabilità: anche in caso di perdita di connettività di rete, i dati possono comunque essere elaborati e raccolti.

Per l'utilizzo della connessione Bluetooth sulla piattaforma è stato necessario generare un UUID per identificare il servizio che si sarebbe esposto che espone 6 caratteristiche:

- roll
- pitch
- yaw
- sx
- sy
- sz

Una volta avviata una connessione verso la piattaforma questa comincia a pubblicare i dati che sono visibili dal device che ha avviato la connessione.

7 Codice

Il codice per il controller e un piccolo visualizzatore creato utilizzando la libreria Processing¹⁰ si possono trovare alla seguente repository GitHub: [sensori_controller](#)

Le cartelle presenti sono:

- *controller_fusion*: la versione finale del controller utilizzando la libreria SensorFusion;
- *controller_buttons*: la parte di gestione dei pulsanti (2) e del builtin led della piattaforma che non ‘ stata integrata nella versione finale per mancanza di tempo;
- *controller_bluetooth*: la versione finale del controller inviando i dati tramite bluetooth e non tramite seriale;
- *extras*: cartella contenente il visualizzatore in Processing;
- *putty*: cartella contenente un log catturato con Putty per alcune analisi fatte in MATLAB;
- *calibration_data*: cartella contenente i dati di calibrazione per il magnetometro dal sito citato nella sezione [4](#)

¹⁰[Processing](#)

8 Conclusioni

Ci sono stati dei buoni risultati e sicuramente con una calibrazione migliore e della analisi più fini degli algoritimi si possono migliorare ancora i risultati.

La parte più problematica resta comunque la stima dello spostamento che difficilmente può raggiungere i risultati dei visori attualmente in commercio senza qualche algoritmo più sofisticato o l'integrazione di altri sensori.