

ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

---

CAMPUS DI CESENA  
SCUOLA DI INGEGNERIA E ARCHITETTURA  
CORSO DI LAUREA MAGISTRALE IN  
INGEGNERIA E SCIENZE INFORMATICHE

SENSOR FUSION PER LA LOCALIZZAZIONE  
INDOOR IN APPLICAZIONI CONTEXT-AWARE

Elaborata nel corso di: Programmazione Avanzata e Paradigmi

*Tesi di Laurea di:*  
MICHELE ROSSI

*Relatore:*  
Prof. A. BEVILACQUA

*Co-relatori:*  
Prof. A. RICCI  
Ing. A. GHERARDI

---

ANNO ACCADEMICO 2015-2016  
SESSIONE III



# Abstract

Nel contesto della Realtà Aumentata e della tecnologia Wearable, la richiesta, e soprattutto la necessità, di applicazioni context-aware ha reso il concetto di *posizione* una informazione estremamente importante. Il presente documento mira a fornire una solida base per la progettazione e realizzazione di un sistema di localizzazione indoor attraverso l'uso di sensori inerziali e di tecniche di filtering, in particolare il filtro di Kalman e le varianti Extended e Unscented per la riduzione degli errori di stima, comparandone le performance.



Alla mia famiglia,  
per avermi sostenuto fino al raggiungimento  
di questo traguardo, con amore e ottimismo.



# Ringraziamenti

Ringrazio il mio relatore, il professor Alessandro Bevilacqua, per la disponibilità e la guida che mi ha fornito nell'elaborazione della tesi.

Ringrazio il professor Alessandro Ricci, per il supporto fornitomi non solo durante lo svolgimento della tesi, ma anche durante il mio percorso di studi.

Ringrazio il Dr. Ing. Alessandro Gherardi per il prezioso aiuto, la competenza e la supervisione che mi ha fornito per tutto lo svolgimento della tesi.

Ringrazio la mia famiglia, mia madre Antonella, mio padre Denis e mio fratello Simone, per l'amore, la comprensione e il sostegno mostratomi durante questo periodo.

Ringrazio i miei colleghi, in particolar modo Giacomo, Francesca e Filippo, per aver condiviso con me le difficoltà di questo percorso, ed avermi sostenuto fino al raggiungimento di questo traguardo.

Ringrazio i miei amici, per aver creduto in me ed avermi incoraggiato fino al conseguimento di questo obiettivo.

Un ringraziamento particolare va a tutte le splendide persone che ho conosciuto durante il mio periodo in Svezia, per avermi motivato e spinto a dare il meglio.



# Indice

<b>Elenco delle Tabelle</b>	<b>xiii</b>
<b>Elenco delle Figure</b>	<b>xvi</b>
<b>Elenco delle Abbreviazioni</b>	<b>xvii</b>
<b>Introduzione</b>	<b>xix</b>
<b>1 La Realtà Aumentata</b>	<b>1</b>
1.1 Mixed Reality . . . . .	1
1.1.1 Tassonomia a tre dimensioni . . . . .	2
1.2 Classificazione dei display . . . . .	4
<b>2 I Dispositivi Indossabili</b>	<b>7</b>
2.1 Concetti chiave . . . . .	7
2.2 Le questioni tecniche legate al progresso della tecnologia indossabile . . . . .	9
2.3 I dispositivi . . . . .	11
<b>3 I Filtri di Bayes</b>	<b>13</b>
3.1 Concetti chiave . . . . .	13
3.2 La proprietà di Markov . . . . .	14
3.3 L'algoritmo dei filtri Bayesiani . . . . .	15
3.3.1 Un esempio pratico . . . . .	16
3.4 Osservazioni . . . . .	18
3.5 Implementazione . . . . .	18
3.5.1 La distribuzione normale multivariata . . . . .	18
3.5.2 I filtri Gaussiani . . . . .	20

<b>4</b>	<b>Il filtro di Kalman</b>	<b>21</b>
4.1	Un primo sguardo ai filtri di Kalman: stima di uno stato ad una variabile . . . . .	21
4.1.1	Esempio: stima di un valore costante . . . . .	24
4.2	Il filtro di Kalman su modelli multidimensionali . . . . .	24
4.2.1	Fase di predizione . . . . .	25
4.2.2	Fase di correzione . . . . .	27
4.3	Il tuning dei parametri . . . . .	28
4.4	Approssimazioni del filtro di Kalman . . . . .	29
4.4.1	Extended Kalman Filter . . . . .	29
4.4.2	Unscented Kalman Filter . . . . .	30
<b>5</b>	<b>Posizione</b>	<b>35</b>
5.1	Magnetometer Measurement Difference . . . . .	36
5.1.1	Il magnetometro come strumento per la localizzazione	36
5.1.2	Approccio proposto . . . . .	37
5.1.3	Algoritmo . . . . .	38
5.1.4	Osservazioni . . . . .	40
5.2	GPS Indoor . . . . .	40
5.2.1	Introduzione ai sistemi GNSS . . . . .	40
5.2.2	Il calcolo della posizione . . . . .	41
5.2.3	Utilizzo dei segnali GNSS per la localizzazione in contesti indoor . . . . .	42
<b>6</b>	<b>Orientamento</b>	<b>45</b>
6.1	Rappresentazione dell'orientamento . . . . .	45
6.1.1	Angoli di Eulero . . . . .	45
6.1.2	Quaternioni . . . . .	47
6.2	I sensori inerziali . . . . .	48
6.2.1	Calcolo dell'orientamento in situazioni statiche . . . . .	49
6.2.2	Velocità angolare . . . . .	50
<b>7</b>	<b>Sensor Fusion</b>	<b>51</b>
7.1	Introduzione alla Sensor Fusion per la localizzazione . . . . .	51
7.2	Modello del sistema . . . . .	53
7.3	Osservazioni . . . . .	54

<b>8</b>	<b>Configurazione dell'esperimento</b>	<b>55</b>
8.1	Introduzione . . . . .	55
8.1.1	Decomposizione di Cholesky . . . . .	55
8.2	Acquisizione dei dati . . . . .	56
8.2.1	Android API . . . . .	56
8.2.2	Allineamento dei sensori . . . . .	59
8.2.3	Conversione tra coordinate geografiche e cartesiane . . . . .	60
8.2.4	Dataset . . . . .	60
8.3	Analisi dei dati . . . . .	61
8.3.1	Piattaforma . . . . .	61
8.3.2	Parametri iniziali . . . . .	61
<b>9</b>	<b>Risultati sperimentali</b>	<b>63</b>
9.1	Risultati preliminari . . . . .	63
9.2	Comparazione tra EKF e UKF . . . . .	65
9.2.1	Deviazione standard . . . . .	65
9.2.2	Complessità computazionale . . . . .	67
9.3	Outline . . . . .	68
<b>10</b>	<b>Conclusioni</b>	<b>69</b>
<b>A</b>	<b>Grafici delle simulazioni</b>	<b>71</b>
	<b>Bibliografia</b>	<b>79</b>



# Elenco delle tabelle

9.1	Le deviazioni standard degli angoli di roll, pitch e yaw ottenute sui dati non filtrati e filtrati con l'EKF e l'UKF in una situazione statica. . . . .	67
9.2	Le deviazioni standard degli angoli di roll, pitch e yaw ottenute sui dati non filtrati e filtrati con l'EKF e l'UKF durante un movimento rettilineo. . . . .	67
9.3	Tempo di computazione richiesto dall'EKF e dall'UKF, da fermo. . . . .	68
9.4	Tempo di computazione richiesto dall'EKF e dall'UKF per completare una singola iterazione del processo di stima. . . .	68



# Elenco delle figure

1.1	Il Reality-Virtuality Continuum di Milgram[16]. . . . .	2
1.2	La tassonomia tridimensionale di Milgram [16]. . . . .	3
1.3	Asse del Grado di Conoscenza del Mondo [16]. . . . .	3
1.4	Asse della Fedeltà di Riproduzione[16]. . . . .	4
1.5	Asse del Livello di Immersività Percepita [16]. . . . .	5
3.1	Rilevazione indiretta degli stati $x_k$ attraverso misurazioni rumorose $z_k$ (adattata da [23]). . . . .	14
3.2	Esempio di comportamento di un filtro Bayesiano in un con- testo uni-dimensionale (adattata da [27]). . . . .	17
3.3	Funzione densità di probabilità di una distribuzione normale nel caso unidimensionale (a) e multidimensionale (b). . . . .	20
4.1	Processo di stima del filtro di Kalman di un singolo valore [28].	22
4.2	Il processo di stima del filtro di Kalman Filter di un singolo valore. . . . .	24
6.1	Sistema di riferimento: convenzione di Tait-Bryan. . . . .	46
8.1	Sistemi di riferimento del dispositivo e dei sensori. . . . .	59
9.1	Grafico della stima dell'orientamento ottenuto grazie all'EKF in una situazione statica. . . . .	64
9.2	Grafico della stima dell'orientamento ottenuto grazie all'UKF in una situazione statica. . . . .	64
9.3	Grafico della stima dell'orientamento ottenuto grazie all'EKF mentre il dispositivo segue un movimento rettilineo. . . . .	65
9.4	Grafico della stima dell'orientamento ottenuto grazie all'UKF mentre il dispositivo segue un movimento rettilineo. . . . .	66

A.1	Filtro EKF, situazione statica, stima dell'angolo di roll. . . .	71
A.2	Filtro EKF, situazione statica, stima dell'angolo di pitch. . . .	72
A.3	Filtro EKF, situazione statica, stima dell'angolo di yaw. . . .	72
A.4	Filtro UKF, situazione statica, stima dell'angolo di roll. . . .	73
A.5	Filtro UKF, situazione statica, stima dell'angolo di pitch. . . .	73
A.6	Filtro UKF, situazione statica, stima dell'angolo di yaw. . . .	74
A.7	Filtro EKF, percorso a zigzag, stima dell'angolo di roll. . . .	74
A.8	Filtro EKF, percorso a zigzag, stima dell'angolo di pitch. . . .	75
A.9	Filtro EKF, percorso a zigzag, stima dell'angolo di yaw. . . .	75
A.10	Filtro UKF, percorso a zigzag, stima dell'angolo di roll. . . .	76
A.11	Filtro UKF, percorso a zigzag, stima dell'angolo di pitch. . . .	76
A.12	Filtro UKF, percorso a zigzag, stima dell'angolo di yaw. . . .	77

# Elenco delle Abbreviazioni

AR	Augmented Reality
AV	Augmented Virtuality
AWGN	Additive White Gaussian Noise
DR	Dead-Reckoning
EKF	Extended Kalman Filter
EPM	Extent of Presence Metaphor
EWK	Extended of World Knowledge
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GRV	Gaussian Random Variable
HMD	Head-Mounted Display
HUD	Head-Up Display
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LTIV	Linear Time-InVariant
KF	Kalman Filter
KG	Kalman Gain
MMD	Magnetometer Measurement Difference
MR	Mixed Reality
PDF	Probability Density Function
RFID	Radio-Frequency IDentification
RF	Reproduction Fidelity
RV	Reality-Virtuality
SdR	Sistema di Riferimento
SNR	Signal-to-Noise Ratio
UKF	Unscented Kalman Filter
UT	Unscented Transform

UWB  
WLAN

Ultra-Wide Band  
Wireless Local Area Network

# Introduzione

Nel contesto della Realtà Aumentata e della tecnologia Wearable, la richiesta, e soprattutto la necessità, di applicazioni context-aware ha reso il concetto di *posizione* una informazione estremamente importante. Nonostante il vasto utilizzo di tecniche di tracciamento outdoor, tra cui il tracciamento GPS, i contesti indoor richiedono un maggior grado di precisione e robustezza: tra le varie tecniche sviluppate ricordiamo l'impiego di beacon WiFi, di magnetometri o di segnali a ultrasuoni.

Il presente documento mira a fornire una solida base per la progettazione e realizzazione di un sistema di localizzazione indoor attraverso l'uso di sensori inerziali e di tecniche di filtering.

Le tecnologie di positioning prese in esame sono:

- *Magnetometer Measurement Difference*: tecnica che impiega il pattern matching sulle differenze di intensità del campo magnetico tra due rilevazioni successive per compensare l'errore introdotto dal *bias*;
- *GPS Indoor*: consiste nell'applicazione di tecniche di filtering e amplificazione su segnali GPS deboli.

Inizialmente, introduciamo brevemente i concetti di *Wearable Computing* e *Realtà Aumentata*, esponendone le principali caratteristiche e, successivamente, vediamo quali sono i risultati raggiungibili dalla combinazione di queste due tecnologie: vengono illustrate le tecniche basate sui filtri di Kalman e le varianti Extended e Unscented per la riduzione degli errori di stima, comparandone le performance.

La scelta di queste tecniche è giustificata dalla disponibilità sul mercato di un ampio numero di dispositivi, gli *smart glass*, poco eterogenei tra loro che, quindi, presentano caratteristiche simili; proprio per questa omogeneità, questa dissertazione getta le basi per lo sviluppo di un framework che

consenta agli sviluppatori di ottenere la posizione e l'orientamento per la realizzazione di applicazioni context-aware.

# Capitolo 1

## La Realtà Aumentata

Azuma (1997) definisce un'applicazione di realtà aumentata come [1]:

*One that combines the real world with the virtual world, is interactive and in real-time, and is registered in three dimensions.*

Sebbene esista una vasta gamma di tecnologie su cui poter implementare applicazioni di realtà aumentata, la piattaforma più diffusa è sicuramente l'*head-mounted display (HMD)*;

Di seguito illustriamo i concetti di *Mondo Reale* e *Mondo Virtuale* e come “sovrapporre” i due mondi.

### 1.1 Mixed Reality

Il concetto di *Mixed Reality* nasce dal tentativo di Milgram di definire la relazione che lega la realtà virtuale a quella aumentata. Supponiamo di avere un asse la cui dimensione indichi il grado di virtualità:

- l'estremo sinistro rappresenta un ambiente puramente reale, costituito da tutto ciò che può essere osse ato o percepito direttamente da una persona;
- l'estremo destro identifica un ambiente puramente virtuale, costituito solamente da oggetti virtuali, come simulazioni di computer-grafica o monitor a immersione totale.

L'idea proposta da Milgram considera lo spazio interposto tra le due realtà come un continuum, da qui il nome *Reality-Virtuality Continuum*: un certo valore di virtualità indica un ambiente con uno specifico rapporto di realtà-virtualità (Figura 1.1). Quindi, permette di considerare ambienti sottoposti sia a leggi del mondo reale che del mondo virtuale: da questo nasce il campo della Mixed Reality [16, 15].

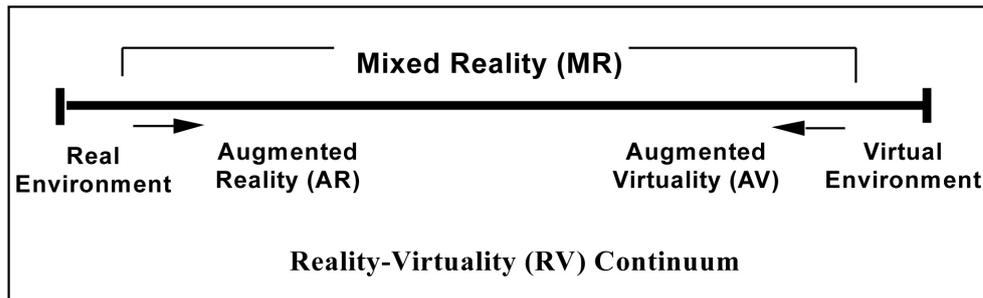


Figura 1.1: Il Reality-Virtuality Continuum di Milgram[16].

La Mixed Reality è a sua volta suddivisibile in *Realtà Aumentata (AR)* e *Virtualità Aumentata (AV)*.

Parlando di realtà aumentata si intende come riferimento una realtà non identica a quella fisica, ma che differisce per qualche particolare; ad esempio, possono essere stati aggiunti degli smart objects o delle entità computazionali che possono o meno interagire con l'ambiente.

Per quanto riguarda la virtualità aumentata, spesso l'utente si trova immerso in un ambiente virtuale, dove, però, possono essere presenti elementi dell'ambiente reale circostante, acquisiti attraverso appositi sensori.

### 1.1.1 Tassonomia a tre dimensioni

Sebbene il continuum espresso da Milgram esprima chiaramente i concetti di AR e AV, una dimensione non è sufficiente a rappresentare le varie possibilità all'interno della categoria dei sistemi di Mixed Reality. Milgram, Takemura, Utsumi e Kishino propongono una tassonomia basata su tre dimensioni (Figura 1.2): il *Grado di Conoscenza del Mondo (EWK)*, la *Fedeltà*

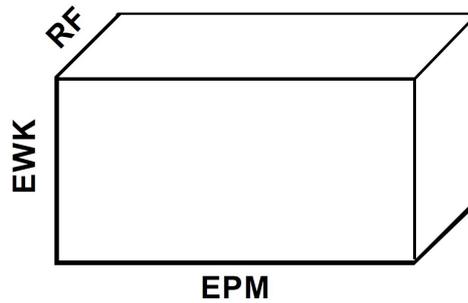


Figura 1.2: La tassonomia tridimensionale di Milgram [16].

di Riproduzione (*RF*) e il Livello di Immersività Percepita (*EPM*)[16]. Il Grado di Conoscenza del Mondo, in inglese Extent of World Knowledge (Figura 1.3), definisce la quantità di informazioni disponibili sugli oggetti e sul mondo visualizzato. L'estremo sinistro equivale ad una diretta acquisizione e poi presentazione all'utente di immagini raccolte, come se l'utente le stesse vedendo direttamente attraverso una videocamera: definisce quella classe di display impiegati, ad esempio, nelle situazioni di telemanipolazione in ambienti non strutturati o sconosciuti, come l'esplorazione di nuovi territori. L'estremo destro indica una completa conoscenza dell'ambiente e degli oggetti al suo interno, la loro posizione e il punto di vista dell'utente, requisiti fondamentali per la creazione di una realtà completamente virtuale.

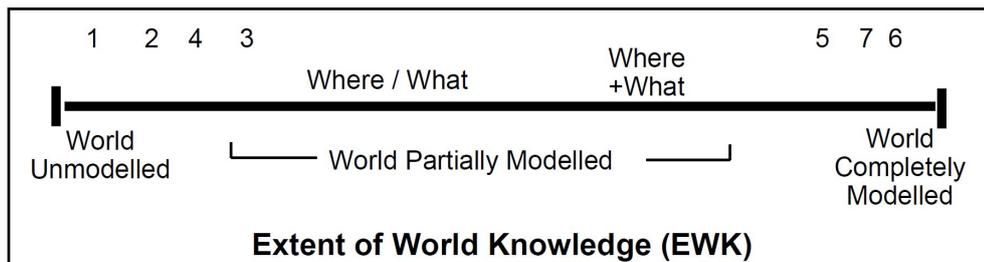


Figura 1.3: Asse del Grado di Conoscenza del Mondo [16].

La Fedeltà di Riproduzione, in inglese Reproduction Fidelity (Figura 1.4),

si riferisce alla qualità con cui le immagini vengono riprodotte e presentate all'utente, che siano esse la rappresentazione di oggetti reali o virtuali, dato che, solitamente, queste sono semplificazioni delle realtà di interesse, limitate dall'hardware, dalle tecniche di acquisizione o dalla risoluzione del display.

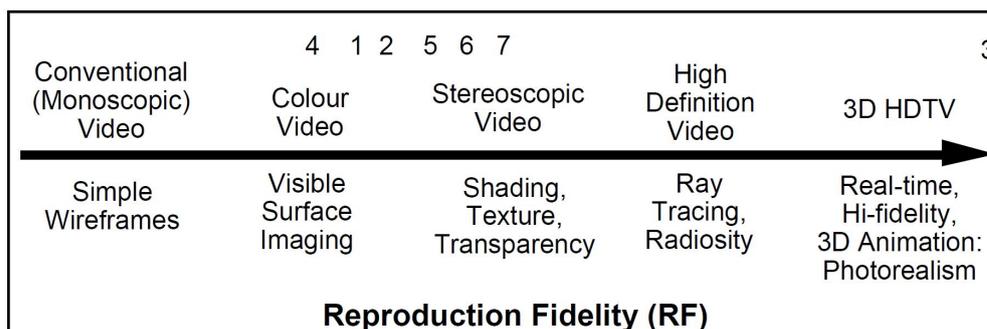


Figura 1.4: Asse della Fedeltà di Riproduzione[16].

Il Livello di Immersività Percepita, in inglese Extent of Presence Metaphor (Figura 1.5), è il indica quanto il soggetto si senta parte del mondo che gli viene mostrato. Può essere ricondotto a concetti di esocentrismo e egocentrismo: nel primo caso, l'utente è riconducibile a un osservatore esterno che guarda il mondo da una "finestra", mentre, nel secondo caso, l'utente percepisce le immagini come se fossero acquisite dal suo punto di vista.

La combinazione di RF e EPM indica il realismo dei display di Mixed Reality.

In conclusione, questo framework permette di mettere a confronto diverse tecnologie ed evidenziarne similitudini e differenze sotto gli aspetti principali.

## 1.2 Classificazione dei display

Gli HMD possono essere classificati in base alla tecnica con cui la percezione del mondo reale e del mondo virtuale vengono combinate e presentate all'utente [1, 2]:

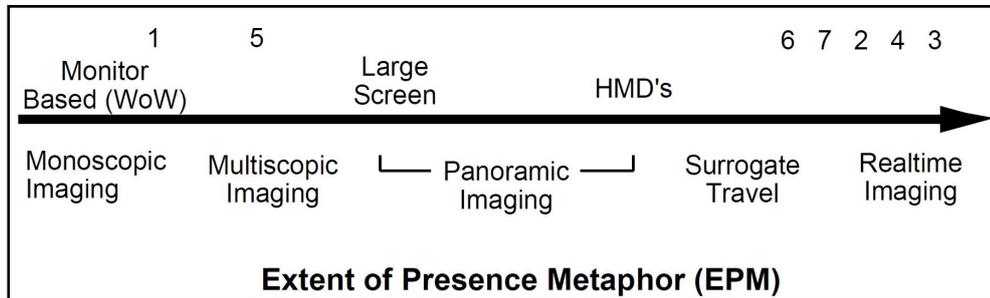


Figura 1.5: Asse del Livello di Immersività Percepita [16].

- *See-through*: secondo questo approccio, l'utente percepisce direttamente il mondo reale, mentre gli elementi virtuali sono sovrapposti a quelli reali attraverso tecniche ottiche o video. Di norma, il display è fissato all'utente.
- *Closed-view*: il sistema combina le informazioni acquisite sul mondo reale con quelle virtuali, mostrando all'utente il risultato pre-computato. Questo è prevalentemente un approccio non immersivo per l'utente, dato che non mostra la realtà aumentata dal punto di vista dell'utente, ma da quello di videocamere esterne (che potrebbero anche essere posizionate su un robot, ad esempio).

Dati i due approcci, i display più diffusi applicano una tra le seguenti alternative:

- *Monitor-based*: questa configurazione è solitamente basata su una serie di videocamere (fisse o mobili), un monitor, un generatore di immagini digitali e un combinatore; il monitor è in una posizione fissa e l'utente è vincolato a restare di fronte al sistema.
- *Optical see-through*: il componente chiave di questo approccio è un combinatore ottico che riduce la luce proveniente dal mondo reale, permettendo di riflettere parte della luce proveniente dal monitor agli occhi dell'utilizzatore.

- *Video see-through*: in questo caso, la visione del mondo reale è acquisita attraverso delle videocamere che, successivamente, verrà combinata con le immagini virtuali e mostrata sul monitor.

L'impiego della tecnica ottica, rispetto alla tecnica video, offre alcuni evidenti vantaggi:

- dato che deve trattare solo un flusso di immagini alla volta, la sovrapposizione ottica è più semplice ed economica;
- sebbene la risoluzione del flusso di immagini virtuali sia limitata da quella del display, la visione del mondo reale non ne risulta influenzata;
- in caso di guasti o esaurimento dell'alimentazione, l'utente si ritroverebbe comunque con una visione diretta dell'ambiente reale, mentre, utilizzando un dispositivo video, l'utente otterrebbe uno schermo nero, che lo renderebbe cieco a ciò che lo circonda;
- l'uso di dispositivi video presuppone l'uso di videocamere che introducono un disallineamento tra la visione acquisita dalla telecamera e quella che sarebbe percepita dalla posizione degli occhi.

D'altro canto, anche la tecnologia video introduce notevoli vantaggi:

- l'approccio video permette di combinare più efficacemente due flussi di immagini, operando a livello di pixel;
- data un'immagine digitalizzata, è facile ridurre le distorsioni attraverso tecniche di image processing, mentre, nel caso ottico, si rende necessario l'utilizzo di sistemi ottici costosi e pesanti;
- in caso di ritardi nella generazione dei due stream, si possono applicare tecniche per sincronizzare e ridurre il disallineamento temporale delle immagini;
- la posizione della testa, nel caso ottico, è data esclusivamente dall'head-tracker, mentre, attraverso le videocamere, si può fare affidamento anche sulla digitalizzazione della scena reale;

Non esiste una tecnica migliore dell'altra: l'impiego di una specifica tecnologia deve sempre essere giustificata dal contesto applicativo di interesse.

# Capitolo 2

## I Dispositivi Indossabili

Con l'avvento di internet e delle tecnologie web e mobile, la maggior parte dei processi, aziendali e non, sta subendo una informatizzazione radicale: ciò rende i computer la base per una vasta gamma di funzionalità, anche se confinati a una scrivania, il che ne limita notevolmente l'utilizzo in diversi settori. Il concetto di *indossabile* nasce proprio dalla necessità di poter disporre delle stesse funzionalità che avremmo stando seduti di fronte ad un computer in un contesto più dinamico, senza che ciò interferisca con le azioni e interazioni quotidiane; la tecnologia indossabile mira a combinare l'ampia disponibilità di funzionalità dei computer e i requisiti di mobilità e flessibilità di un utente spesso in movimento [26].

### 2.1 Concetti chiave

Il concetto di Computazione Indossabile, in inglese *Wearable Computing*, è spesso associato a dispositivi elettronici indossabili, come piccole tastiere a una mano e HMD: tuttavia, non esiste una singola definizione condivisa da tutto il mondo accademico. Di seguito citiamo tre definizioni che, complessivamente, racchiudono i concetti chiave di questa tecnologia:

Rhodes (2000) [24, 21]

*Wearables provide portability during operation; enable hands-free or hands-limited use; can attract the user's attention, even*

*when not in active use; can run continuously; and attempt to sense the user's current context.*

Kortuem (1998) [24, 10]

*The user interface technique that allows focusing the user's attention and presenting information in an unobtrusive, context-dependent manner.*

Mann (1997) [24]

*Wearables are constant and always ready, unrestrictive, not monopolizing of user attention, observable and controllable by the user, attentive to the environment, useful as a communication tool, and personal.*

Dalle definizioni precedenti individuiamo quelli che possono essere considerati gli attributi principali di un computer indossabile [10, 24]:

- *Applicazioni hands-free o one-handed*: un computer indossabile permette al suo utilizzatore di avere una o entrambe le mani libere per poter svolgere compiti manuali mentre interagisce con il sistema; questo può richiedere metodi di input alternativi (come le interfacce vocali, ad esempio);
- *Accesso costante e persistente ai servizi*: i dispositivi indossabili sono pensati per un uso quotidiano e continuo; inoltre deve essere garantito il corretto funzionamento in situazioni dette “on the go” nelle quali l'utente si trova in movimento o impegnato in altri compiti: detto ciò il sistema deve interagire con l'utente senza distrarlo e allo stesso tempo supportarlo nello svolgimento dei suoi compiti;
- *Monitoraggio dell'ambiente*: il dispositivo deve essere in grado di percepire l'ambiente circostante per poter reagire prontamente e coerentemente con ciò che succede intorno all'utente. La maggior parte dei dispositivi dispone di una serie di sensori atti all'identificazione delle caratteristiche dell'ambiente circostante (come videocamere per l'individuazione di specifici oggetti, o sistemi GPS per la navigazione);
- *Realtà Aumentata*: l'interfaccia utente non deve risultare ingombrante o limitante, permettendo interazioni rapide con un minimo sforzo da parte dell'utente. Inoltre, l'utente ottiene una serie di informazioni *context-dependent* aggiuntive.

## 2.2 Le questioni tecniche legate al progresso della tecnologia indossabile

I dispositivi indossabili, grazie alle loro particolari caratteristiche, possono essere impiegati in diversi settori, che si ampliano di pari passo al progresso tecnologico. In particolare, si rende necessario affrontare alcune questioni che, come possiamo notare, hanno permesso (e continuano ad influenzare) una diffusione sempre maggiore di questa tipologia di dispositivi [2, 24, 25]:

- *Alimentazione*: come abbiamo già evidenziato precedentemente, i dispositivi wearable sono pensati per un uso continuo e quotidiano, quindi devono fornire una certa durata; inoltre, dato che fanno parte di componenti indossabili, il loro peso non deve superare una certa soglia. La ricerca tecnologica deve portare ad un trade-off tra queste due grandezze, senza trascurare il consumo dei componenti elettronici:
  - Tra le soluzioni più diffuse troviamo la sostituzione della batteria principale con una secondaria, ma questo presuppone che l'utente ricordi di caricarle entrambe a fine giornata;
  - potrebbe risultare sufficiente una batteria con durata adatta ad alimentare il dispositivo per le ore di attività dell'utente, per esempio 16 ore, mentre le restanti 8 sono utilizzate per ricaricarlo. Se parliamo di dispositivi integrati negli indumenti, possiamo pensare ad un metodo di ricarica wireless integrato nel guardaroba, oppure il recupero di energia attraverso meccanismi che sfruttano le attività motorie;
  - sebbene i recenti studi siano stati guidati dalla necessità di batterie sempre più piccole con un'elevata capacità, il progresso tecnologico in questo campo non cresce di pari passo alle altre tecnologie;
  - inoltre, il costo delle batterie rimane tra i più alti, se paragonato agli altri componenti dei dispositivi;
- *Networking*: per essere in grado di fornire servizi come la geolocalizzazione, ricerche sul web o computazioni cloud, il dispositivo indossabile deve poter comunicare attraverso diverse tipologie di rete, dalle reti cellulari, alle wireless e persino a quelle bluetooth. Un indossabile

deve fornire diverse tipologie di interazione, date le diverse caratteristiche dei vari standard di comunicazione, e non deve permettere che le trasmissioni interferiscano tra loro:

- Off-body: comprende tutte quelle trasmissioni provenienti dal dispositivo ed indirizzate ad una infrastruttura di rete fissa. Alcune problematiche legate a questo tipo di comunicazioni sono dovute alla mobilità dell'utente (il dispositivo potrebbe perdere il segnale) o alla condivisione della stessa banda di trasmissione da più trasmettitori (infatti sono stati sviluppati diversi standard per gestire gli accessi multipli). Nonostante i notevoli progressi nel campo del networking, nessun sistema di comunicazione può essere considerato ubiquitos: perciò è necessario prevedere degli appositi protocolli che gestiscano l'accesso remoto a risorse attualmente non raggiungibili;
  - Interoperabilità: come abbiamo anticipato, i dispositivi indossabili devono essere in grado di comunicare attraverso diversi standard; ciò implicherebbe un incremento notevole dei componenti adibiti alla comunicazione: possiamo risolvere parzialmente il problema gestendo tutte le trasmissioni attraverso una *software radio* che permetta di “emulare” i vari componenti hardware.
  - Near-body: identifica quel tipo di trasmissioni indirizzate a oggetti esterni nelle immediate vicinanze (ad esempio, le trasmissioni con tecnologia RFID): grazie a dei tag posizionati in determinate posizioni è possibile trasmettere e ricevere informazioni (utile nel caso si renda necessario effettuare operazioni context-dependent, dove la posizione rappresenta un'informazione molto importante);
- *Compromesso tra prestazioni e attrattività*: durante la progettazione di un indossabile, non si può non tenere conto della forma e delle dimensioni: spesso le persone preferiscono oggetti con fatture eleganti o che richiamino l'aspetto di oggetti di uso quotidiano; tuttavia, stiamo parlando di piccoli computer che consumano energia e che, quindi, producono calore, proporzionalmente al task eseguito. Quindi, va tenuto in considerazione anche il fattore di dissipazione termica durante la progettazione; tuttavia, spesso non è facile trovare un buon compro-

messo tra prestazioni, dissipazione, e appetibilità (in termini estetici). Alcune strategie di risparmio energetico possono ridurre parzialmente il problema del surriscaldamento: ad esempio, nel contesto delle comunicazioni wireless, se il segnale è debole è richiesta più potenza per la trasmissione, mentre si potrebbe attendere di trovarsi in una zona con una ricezione migliore;

- *Privacy*: per permettere al sistema di fornire certe funzionalità, il wearable deve trasmettere alcune informazioni che possono essere confidenziali o semplicemente private per l'utente; ad esempio, dato l'argomento della tesi, la posizione del dispositivo è un'informazione necessaria all'utente, ma non deve diventare di dominio pubblico. Si rendono necessari appositi accorgimenti come la cancellazione di tutti i dati quando non più necessari oppure permettere all'utente di specificare in modo esplicito chi possa avere accesso a tali dati.

## 2.3 I dispositivi

Le tipologie di indossabili più diffuse sono certamente gli Smartwatch e gli SmartGlass: i primi hanno, normalmente, una forma che richiama quella degli orologi da polso, ma forniscono funzionalità nettamente superiori, mentre i secondi mirano a fornire una visione dell'ambiente circostante "aumentata" rispetto a ciò che percepirebbe l'utente autonomamente, grazie alle informazioni aggiuntive raccolte dal dispositivo in modo autonomo o ricevute dalle infrastrutture di supporto collocate nell'ambiente (tag visivi e beacon ad esempio).



# Capitolo 3

## I Filtri di Bayes

Un concetto chiave della Robotica Probabilistica è quello della *belief*: con questo termine si identifica la *conoscenza* che il robot ha sullo *stato reale* dell'ambiente; spesso tale stato, infatti, differisce da quello inferito dai dati, visto che non è misurabile direttamente.

Il filtro di Bayes, o Recursive Bayesian Estimation, denota un approccio probabilistico per la stima della *belief* in modo ricorsivo, date in ingresso rilevazioni “rumorose”. In questo caso, per *filtro* si intende il processo di stima dello stato corrente di un sistema dinamico.

### 3.1 Concetti chiave

Le interazioni tra un robot e l'ambiente possono avvenire in due modi [27]:

- Misurazioni dei sensori: il robot utilizza i sensori a sua disposizione per rilevare lo stato dell'ambiente;
- Azioni di Controllo: interazioni atte alla modifica dello stato del *mondo*, come ad esempio il movimento del robot o la manipolazione di oggetti circostanti.

Le rilevazioni forniscono informazioni allo scopo di aumentare la conoscenza del mondo, mentre le azioni introducono una perdita di conoscenza dovuta alla stocasticità dell'ambiente e alla accuratezza di esecuzione degli attuatori.

Interagendo con l'ambiente, quindi, il sistema riceve in ingresso due tipi di flussi di dati:

- *Dati di misurazione*: sono i dati riguardanti lo stato dell'ambiente in uno specifico momento. Il termine  $z_t$  denota una misurazione al tempo  $t$ ;
- *Dati di controllo*: forniscono informazioni sulla transizione da uno stato all'altro. Un esempio evidente di dato di controllo è la velocità: dato un robot in una posizione iniziale, una velocità di movimento e un'intervallo di tempo è possibile ottenere approssimativamente la sua posizione finale. Il dato di controllo al tempo  $t$  è denotato come  $u_t$ .

Tuttavia, nei problemi di filtraggio ottimale, il valore osservato  $z_k$  è dato dalla somma tra il valore di stato  $x_k$  (stato reale in cui si trova il sistema) e un *rumore di misurazione*. Lo scopo è quindi quello di stimare lo stato nascosto  $x_k$  (Figura 3.1) [23].

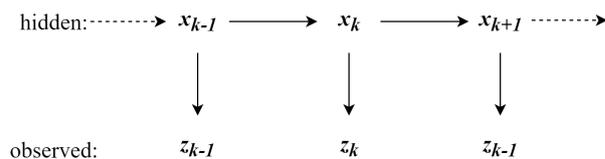


Figura 3.1: Rilevazione indiretta degli stati  $x_k$  attraverso misurazioni rumorose  $z_k$  (adattata da [23]).

La robotica Probabilistica rappresenta le belief attraverso *distribuzioni di probabilità condizionata*, assegnando un valore di densità di probabilità ad ogni possibile stato, condizionato dai dati attualmente rilevati:

$$bel(x_t) = p(x_t | z_{1:t}, u_{1:t}). \quad (3.1)$$

## 3.2 La proprietà di Markov

I filtri di Bayes sono stati ideati per stimare la belief dello stato  $x_t$ , rappresentato da una *variabile casuale*, attraverso tutte le informazioni raccolte sequenzialmente dai sensori: perciò, la distribuzione di probabilità  $bel(x_t)$

su  $x_t$  ne rappresenta l'incertezza. La belief definita come nell'Equazione 3.1 implica che la complessità computazionale del processo di stima cresca in modo esponenziale con l'aumento del numero di rilevazioni provenienti dai sensori. Per rendere la stima *computazionalmente trattabile*, si assume che il modello sia *Markoviano* [5]. Assumere che il modello descriva un sistema Markoviano implica che lo stato  $x_t$  sia uno stato *completo*, quindi [23]:

- *Proprietà Markoviana degli stati*:  $x_t$  dipende solamente dallo stato precedente  $x_{t-1}$ , rendendo valida la seguente uguaglianza

$$p(x_t | x_{t-1}, x_{t-2}, \dots, x_0) = p(x_t | x_{t-1}). \quad (3.2)$$

Gli stati  $\{x_t : t = 0, 1, 2, \dots\}$ , dunque, formano una *sequenza Markoviana*.

- *Indipendenza condizionale delle misurazioni*: le rilevazioni  $z_t$  dipendono solamente dallo stato corrente al tempo  $t$ , quindi

$$p(z_t | x_{0:t}, z_{1:t}, u_{1:t}) = p(z_t | x_t) \quad (3.3)$$

La proprietà di Markov permette di computare efficientemente la belief evitando perdite di informazioni. Tuttavia, spesso i modelli non riproducono dettagliatamente le dinamiche del sistema: in certi casi può non essere possibile documentare in modo preciso la realtà di interesse; altre volte si sceglie una riproduzione approssimata della realtà per ridurre la complessità computazionale. Nonostante ciò, i filtri di Bayes hanno dimostrato una certa robustezza di fronte a queste violazioni [27].

### 3.3 L'algoritmo dei filtri Bayesiani

L'algoritmo permette di ottenere la stima  $bel(x_t)$  al tempo  $t$  in due fasi [23, 27]:

- *Predizione*: il filtro di Bayes è *ricorsivo*, quindi l'algoritmo "prevede" che  $bel(x_t)$  assuma un certo valore  $\overline{bel}(x_t)$ , calcolato attraverso il parametro di controllo  $u_t$  e la stima  $bel(x_{t-1})$  ottenuta all'iterazione precedente;

$$\overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1} \quad (3.4)$$

- *Aggiornamento*: data in ingresso una nuova misurazione  $z_t$ , per ogni possibile  $x_t$  raggiungibile dallo stato corrente, l'algoritmo moltiplica la belief  $\overline{bel}(x_t)$  per la probabilità che la misurazione  $z_t$  sia osservata dal sistema; il valore ottenuto viene normalizzato moltiplicando il risultato ottenuto per la costante di normalizzazione  $\eta$ , ottenendo infine la belief  $bel(x_t)$

$$bel(x_t) = \eta p(z_t|u_t) bel_p(x_t). \quad (3.5)$$

Questa fase è chiamata anche *passo di correzione*.

Riassumendo, l'Algoritmo dei Filtri Bayesiani calcola il valore a posteriori dello stato  $x_t$  condizionato dalla rilevazione e dal valore del parametro di controllo al tempo  $t$ , assumendo che il mondo sia Markoviano e lo stato sia completo.

Come si evince dalle Equazioni 3.4 e 3.5, la belief  $bel(x_t)$  al tempo  $t$  è calcolata dalla belief  $bel(x_{t-1})$  al tempo  $t-1$ : questo è ciò che rende il filtro di Bayes *ricorsivo*.

### 3.3.1 Un esempio pratico

Per illustrare il comportamento e le proprietà di un filtro Bayesiano ci serviamo di un esempio tratto dall'articolo *Bayesian Filtering for Location Estimation, 2003* [5].

In un mondo unidimensionale, una persona cammina lungo un corridoio trasportando un sensore che invia un segnale ogni qualvolta si trovi di fronte a una porta; il sensore non fa distinzione tra porte diverse. Nella Figura 3.2) troviamo le diverse posizioni assunte dalla persona lungo il corridoio e la belief corrente:

1. Dato che la posizione iniziale della persona non è determinabile, la distribuzione è uniforme su tutte le possibili posizioni (Figura 3.2a);
2. dopodiché, dato che la persona si trova di fronte ad una porta, il sensore lancia un segnale di "porta identificata". Di conseguenza, la belief che ne risulta attribuisce una alta probabilità alle posizioni in prossimità delle porte, dando luogo a tre picchi, sebbene indistinguibili tra loro. Inoltre, la probabilità assegnata alle posizioni distanti dalle porte è comunque diversa da zero, in modo da tenere conto di eventuali errori di lettura da parte dei sensori (Figura 3.2b);

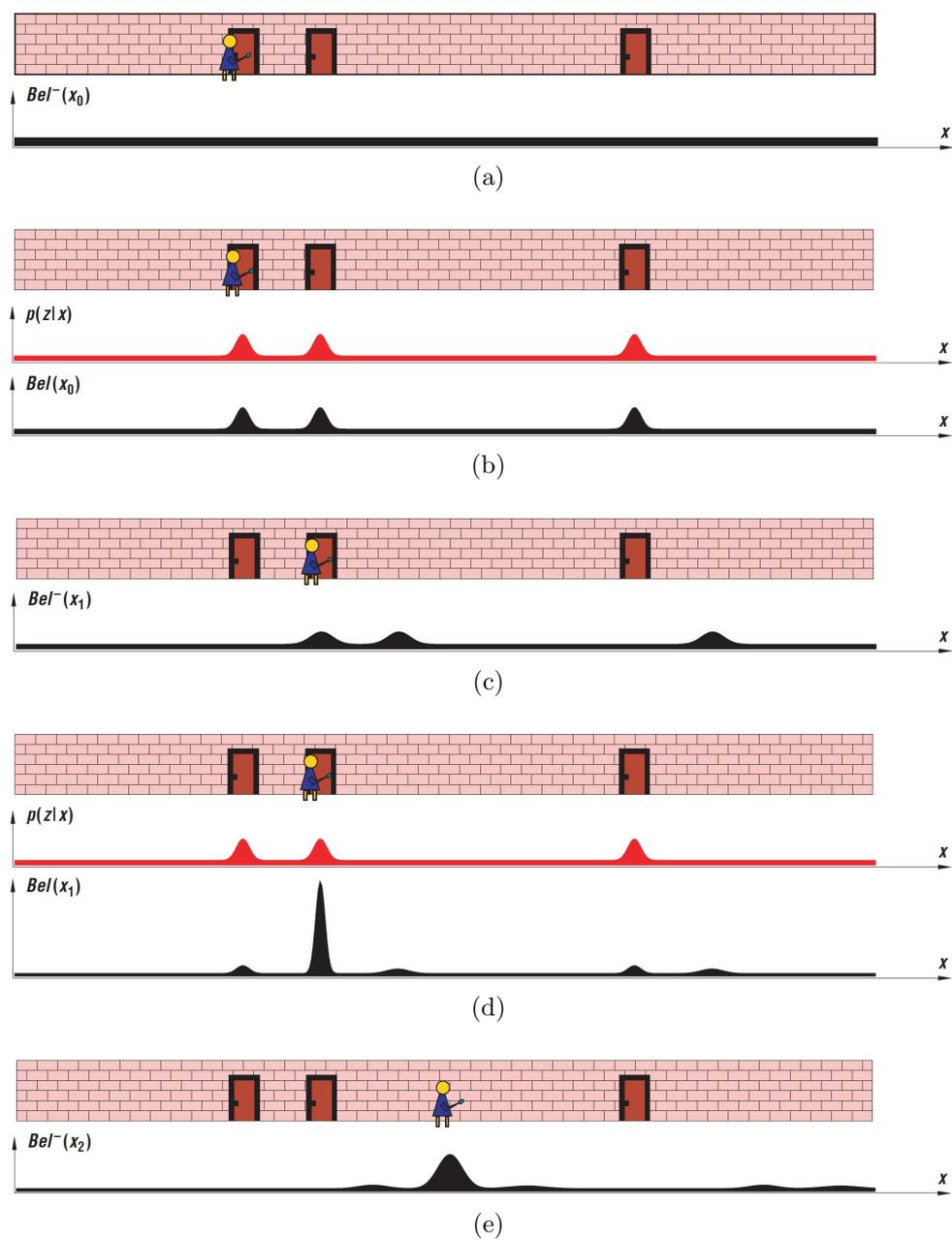


Figura 3.2: Esempio di comportamento di un filtro Bayesiano in un contesto uni-dimensionale (adattata da [27]).

3. considerando la persona in movimento (verso destra), il filtro di Bayes trasla la belief nella stessa direzione e la appiana (smoothing) per tenere conto delle incertezze nella stima del movimento (Figura 3.2c);
4. successivamente, il sensore invia un altro segnale e l'utente si sposta nuovamente, mostrando una alta probabilità in una posizione prossima a quella reale. (Figure 3.2d) e 3.2e).

In sintesi, ogni qualvolta un sensore lancia un segnale rappresentante una rilevazione, il filtro effettua la fase di predizione attraverso la Equazione 3.4 ottenendo  $\overline{bel}(x_t)$  ( $Bel^-$  nella Figura 3.2) e ne corregge la stima grazie al valore osservato  $z_t$  e alla Equazione 3.5, ottenendo  $bel(x_t)$ .

## 3.4 Osservazioni

Sebbene il filtro di Bayes esprima nel modo più generale possibile il processo di stima per poter ottenere belief statisticamente consistenti, le varie implementazioni mostrano diversi comportamenti sulla base delle differenti assunzioni sul modello o lo stato iniziale del sistema. Come abbiamo già anticipato nella Sezione 3.2, tutto ciò è dovuto alle diverse necessità specifiche del caso di interesse che ha come risultato una vasta gamma di tecniche e algoritmi *ad hoc* che generano delle *approssimazioni* del reale valore della belief. Nel momento di scegliere una particolare approssimazione, è necessario, quindi, trovare un *trade-off* adatto tra l'*efficienza computazionale*, la *precisione della approssimazione* e la *facilità di implementazione* dell'algoritmo.

## 3.5 Implementazione

### 3.5.1 La distribuzione normale multivariata

La più comune funzione di densità di probabilità (PDF) è data dalla *distribuzione normale* unidimensionale, dove si assume che, quindi,  $x$  sia un valore scalare. La PDF di una distribuzione normale con media  $\mu$  e varianza  $\sigma^2$  è esprimibile matematicamente come

$$p(x) = (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2} \frac{(x - \mu)^2}{\sigma^2}\right\}. \quad (3.6)$$

La distribuzione normale multivariata generalizza il concetto di distribuzione normale a uno spazio multidimensionale, dove  $x$  rappresenta un vettore e la sua PDF è esprimibile attraverso l'equazione

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right\}, \quad (3.7)$$

dove  $\mu$  è il vettore media e  $\Sigma$  una matrice positiva chiamata *matrice di covarianza*. La matrice di covarianza è una matrice quadrata i cui elementi nelle posizioni  $i$  e  $j$  rappresentano la covarianza tra l'elemento  $i$ -esimo e  $j$ -esimo di un vettore casuale multivariato, ovvero

$$\Sigma_{i,j} = \text{cov}(X_i, X_j), \quad (3.8)$$

dove la covarianza tra due variabili casuali è definita come

$$\text{cov}(X_i, X_j) = \text{E}[(X_i - \text{E}[X_i])(X_j - \text{E}[X_j])]. \quad (3.9)$$

Una matrice quadrata è una matrice di covarianza valida se e solo se [23]:

- gode della proprietà di simmetria, ovvero

$$A^T = A; \quad (3.10)$$

- è una matrice definita positiva, ovvero se, dato un vettore  $y \in \mathbb{C}$  e  $y^\dagger$  il suo trasposto complesso coniugato definito come

$$y^\dagger = (y^T)^*, \quad (3.11)$$

allora la seguente relazione risulta valida:

$$y^\dagger A y \geq 0. \quad (3.12)$$

Tuttavia, considerando che operiamo nel campo reale e, quindi,  $y \in \mathbb{R}$ , l'Equazione 3.12 può essere sostituita dalla seguente:

$$y^T A y \geq 0. \quad (3.13)$$

Nel caso  $x$  sia un valore scalare, l'Equazione 3.7 risulta equivalente alla 3.6. In Figura 3.3 possiamo osservare la diversa distribuzione di probabilità della distribuzione normale e della sua variante generalizzata.

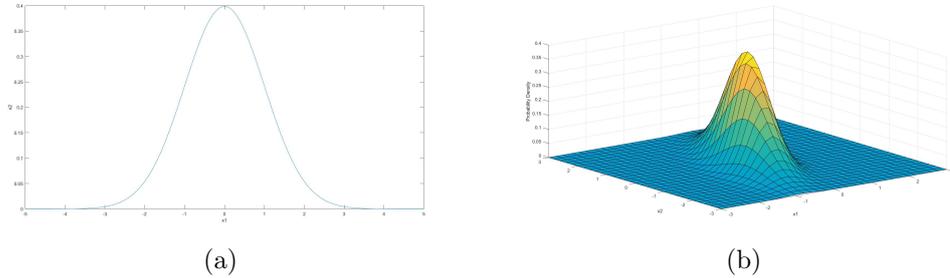


Figura 3.3: Funzione densità di probabilità di una distribuzione normale nel caso unidimensionale (a) e multidimensionale (b).

### 3.5.2 I filtri Gaussiani

La prima implementazione del filtro di Bayes nel tempo continuo, computazionalmente trattabile, è data dai cosiddetti *Filtri Gaussiani*; ciò che sta alla base di ogni tecnica di filtraggio Gaussiano è l'ipotesi che la belief possa essere rappresentata attraverso una *distribuzione normale multivariata* [23]. Assumere che  $bel(x_t)$  segua una distribuzione normale comporta che la belief possa essere rappresentata semplicemente grazie ai valori di media e di covarianza: questa rappresentazione è chiamata *moments representation*.

Nel Capitolo 4 illustriamo le caratteristiche e il comportamento dell'algoritmo del *filtro di Kalman*, il quale implementa il filtro di Bayes e assume che le belief siano rappresentate da delle distribuzioni Gaussiane.

# Capitolo 4

## Il filtro di Kalman

Il filtro di Kalman è stato ideato da Rudolph Emil Kalman negli anni 50; è uno stimatore ricorsivo ottimale dello stato di un sistema che minimizza l'errore introdotto da misurazioni incomplete o soggette a rumore, dove il modello di stato e il modello di misura sono approssimati a modelli lineari Gaussiani [23]; implementa i concetti del filtro di Bayes e descrive la distribuzione Gaussiana in forma di moments representation (vedi Sottosezione 3.5.2). Il filtro di Kalman è in grado di computare solamente la belief di *stati continui*. Non può essere utilizzato su spazi di stato discreti o ibridi[27]. Nonostante le condizioni di ottimalità spesso non siano verificate, il filtro di Kalman fornisce risultati più che soddisfacenti per diverse tipologie di applicazioni.

### 4.1 Un primo sguardo ai filtri di Kalman: stima di uno stato ad una variabile

Una definizione più immediata del concetto di filtro di Kalman può essere la seguente [28]:

*It is an iterative mathematical process that use a set of equations and consecutive data inputs to quickly estimate the true value, position, velocity, etc. of the object being measured, when the measured values contain unpredicted or random error, uncertainty or variation.*

Per comprendere meglio il funzionamento di un filtro di Kalman possiamo fare affidamento alla Figura 4.1, che raffigura il comportamento del processo di stima nel caso di uno stato formato da una singola variabile.

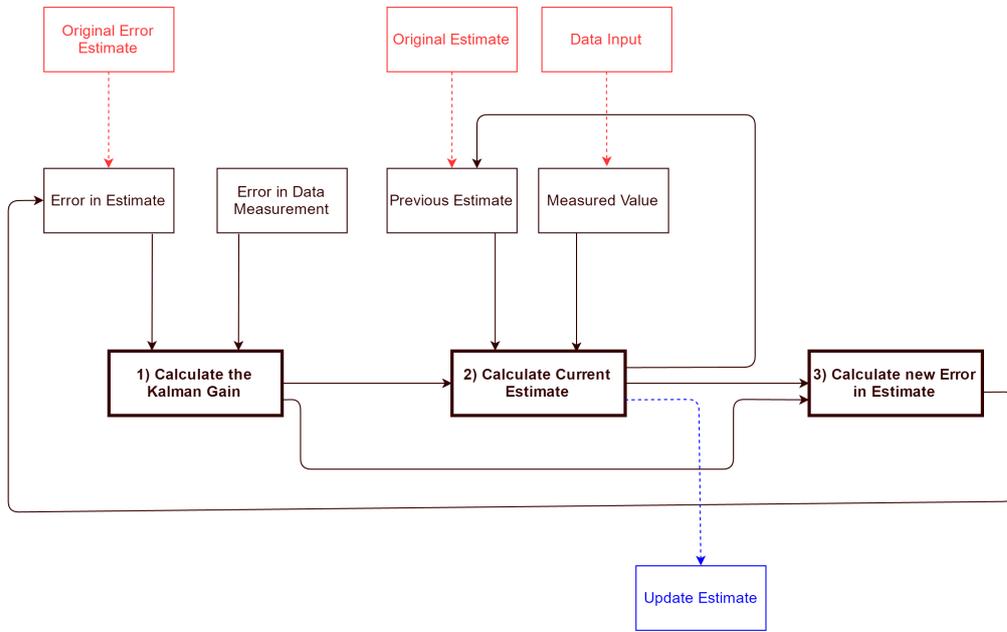


Figura 4.1: Processo di stima del filtro di Kalman di un singolo valore [28].

Se assumiamo che  $EST_t$  sia la nostra *stima corrente*,  $EST_{t-1}$  la *stima precedente* (quindi la stima all'iterazione precedente),  $MEA$  la misurazione all'iterazione corrente,  $\varepsilon_{EST}$  l'incertezza del processo di stima e  $\varepsilon_{MEA}$  l'incertezza della rilevazione, il processo di stima è, quindi, composto principalmente da 3 passi:

1. Calcolo del *guadagno di Kalman*;
2. Calcolo della stima corrente;
3. Calcolo del nuovo errore di stima.

Il primo passo è il calcolo del guadagno di Kalman: dati in ingresso l'errore di stima  $\varepsilon_{EST}$  o, nel caso della prima iterazione, l'errore di stima originale,

e l'errore di misura  $\varepsilon_{MEA}$ , il guadagno di Kalman è dato dall'Equazione 4.1:

$$KG = \frac{\varepsilon_{EST}}{\varepsilon_{EST} + \varepsilon_{MEA}}. \quad (4.1)$$

Il guadagno di Kalman (KG) indica il peso assegnato alla stima corrente e alla misurazione, quindi stabilisce quanto ciascuno dei due valori andrà a incidere sulla stima finale: se l'errore di stima è molto basso possiamo ritenere più attendibile il risultato del processo di stima, mentre, in caso contrario, verrà data più importanza ai valori rilevati. Il KG è uno scalare compreso tra 0 e 1.

Il secondo passo consiste nel calcolo della stima relativa all'iterazione corrente. Oltre al KG, per calcolare la nuova stima sono necessari la stima precedente e la rilevazione corrente: la stima si ottiene grazie all'Equazione 4.2:

$$EST_t = EST_{t-1} + KG [MEA - EST_{t-1}]. \quad (4.2)$$

Nel caso la rilevazione risulti molto attendibile, KG sarà più elevato, dato che l'errore di misura tenderà a 0; in caso contrario, l'errore di misurazione sarà superiore, risultando in un KG prossimo allo 0.

L'ultimo passo aggiorna l'errore di stima grazie alla stima corrente e al guadagno di Kalman:

$$E_{EST_t} = \frac{(\varepsilon_{MEA})(\varepsilon_{EST_{t-1}})}{(\varepsilon_{MEA}) + (\varepsilon_{EST_{t-1}})} \Rightarrow EST_t = [1 - KG](\varepsilon_{EST_{t-1}}). \quad (4.3)$$

L'errore di stima risulterà sempre inferiore a quello dell'iterazione precedente, dato che, in base al valore del KG, il comportamento differisce solo per la velocità con cui converge a zero:

- un valore di KG elevato deriva da un ridotto valore di  $\varepsilon_{MEA}$ ; l'errore di stima  $\varepsilon_{EST}$ , quindi, convergerà velocemente a 0 dato che avremo piccole variazioni nei valori misurati;
- un KG ridotto, al contrario, simboleggia un ampio errore di misurazione, portando a una lenta convergenza della stima al valore reale, a causa di rilevazioni poco attendibili.

### 4.1.1 Esempio: stima di un valore costante

Per cominciare, scegliamo un semplice problema di stima di un valore di temperatura dato un insieme di rilevazioni rumorose: prendendo un valore reale di temperatura di 72 gradi, una stima iniziale di 68 gradi e una successione di 119 rilevazioni con un errore di rilevazione di 4 gradi, distribuito uniformemente, possiamo notare dalla Figura 4.2 che la stima del filtro di Kalman converge al valore reale.

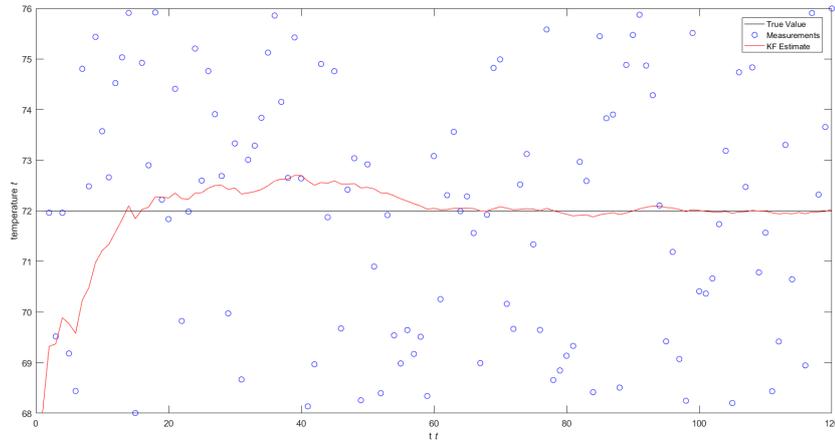


Figura 4.2: Il processo di stima del filtro di Kalman Filter di un singolo valore.

## 4.2 Il filtro di Kalman su modelli multidimensionali

La formulazione originale del filtro di Kalman presuppone uno stato formato da due elementi [30]:

- *Matrice di stato*: il vettore  $x_k$ , con  $x \in \mathbb{R}^n$  racchiude tutte le variabili di stato da stimare;
- *Matrice di covarianza dello stato*: denotata con  $P_k$ , rappresenta la stima dell'accuratezza del valore dello stato ottenuto al passo  $k$ .

Il processo di stima avviene in due fasi ben distinte:

1. *Time update*: questa fase, chiamata anche *fase di predizione*, permette di ottenere una stima dello stato futuro proiettando nel tempo la stima dello stato corrente e della matrice di covarianza dell'errore, ovvero una stima *a priori*;
2. *Measurement Update*: la seconda fase, detta anche *fase di correzione*, regola il valore della stima a priori incorporando la nuova rilevazione, ottenendo così una stima *a posteriori*.

Alla prima iterazione lo stato equivale allo stato iniziale, quindi  $x_{k-1} = x_0$  e  $P_{k-1} = P_0$ . Scegliere un  $x_0$  distante dal valore reale influisce solamente sul tempo necessario all'algoritmo per convergere. La scelta di  $P_0$ , invece, risulta più cruciale: scegliere un valore prossimo allo zero significa assumere di conoscere lo stato iniziale con molta precisione, dando poco peso al valore delle rilevazioni; in caso contrario, utilizzare un valore molto alto presuppone una stima pessimistica dei valori iniziali, dando molta importanza alle rilevazioni [17]. Spesso  $P_0$  è una matrice diagonale.

## 4.2.1 Fase di predizione

### Predizione dello stato

La prima equazione permette di “predire” lo stato futuro propagando lo stato corrente in base al modello fisico del sistema; in formule

$$x_k^- = Ax_{k-1} + Bu_k + w_k, \quad (4.4)$$

dove:

- $x_k^-$ : rappresenta la predizione dello stato calcolata sul modello fisico e lo stato precedente;
- $u_k$ : la matrice delle variabili di controllo, con  $u \in \mathbb{R}^l$ , raccoglie le informazioni relative ai parametri di controllo conosciuti come può esserlo, in molti casi, l'accelerazione (ad esempio, il caso del lancio di un sasso o di un grave in caduta libera, dove il parametro di controllo racchiude l'effetto delle forze che agiscono sul corpo, come la forza di gravità). A volte i dati di controllo sconosciuti possono essere modellati come un rumore di processo casuale [17];

- $w_k$ : il rumore di processo è una sorgente di rumore bianco con distribuzione normale di media 0 e varianza data dalla matrice di covarianza  $Q$ , quindi

$$p(w) \sim N(0, Q); \quad (4.5)$$

- $A$ : rappresenta il modello di transizione dello stato attraverso una matrice  $n \times n$ . In assenza di una funzione di controllo o di rumori di processo, la matrice  $A$  mette in relazione lo stato al tempo  $t - 1$  con quello al tempo corrente  $t$ ;
- $B$ : la matrice  $B$  ha dimensioni  $n \times l$  e rappresenta il modello di controllo, mettendo in relazione i parametri di controllo con lo stato;  $Bu_k$ , quindi, modifica il valore  $Ax_{k-1}$  tenendo conto dell'influenza della variabile di controllo;
- $Q$ : = la *matrice di covarianza del rumore di processo* riflette il grado di incertezza imputabile al processo di stima e alle caratteristiche non modellabili nello stato; spesso, dato che non è possibile osservare il comportamento del processo direttamente, si possono ottenere buoni risultati anche con modelli di processo relativamente semplici. Il valore di  $Q$  deve essere abbastanza piccolo da non vanificare il processo di rilevazione e, al tempo stesso, non troppo grande da introdurre maggior incertezza, rendendo il processo di stima inutile [17]. La matrice  $Q$  è spesso approssimata a una matrice diagonale e, partendo da valori prossimi allo zero dell'ordine di  $10^{-9}$ , si effettua una fase di tuning fino ad ottenere il comportamento desiderato.

### Predizione dell'errore di stima

Questa fase del processo di predizione permette di ottenere una stima a priori dell'errore del valore di stato stimato:

$$P_k^- = AP_{k-1}A^T + Q, \quad (4.6)$$

dove:

- $P_k^-$ : rappresenta la matrice di covarianza dell'errore a priori, quindi calcolata solamente attraverso le informazioni del passo precedente e il modello fisico.

## 4.2.2 Fase di correzione

Le equazioni costituenti la fase di correzione sono responsabili dell'incorporazione delle informazioni derivate dalle rilevazioni con la stima a priori ottenuta dal processo di predizione, ottenendo una stima a posteriori migliore.

### Equazione di misurazione

La  $k$ -esima rilevazione viene indicata come  $z_k$ , con  $z \in \mathbb{R}^m$ , e espressa attraverso la formula

$$z_k = Hx_k + v_k, \quad (4.7)$$

dove:

- $v_k$ : il rumore di misurazione è una sorgente di rumore bianco con distribuzione normale di media 0 e varianza data dalla matrice di covarianza  $R$ :

$$p(v) \sim N(0, R). \quad (4.8)$$

In base al tipo di applicazione, può essere assunto come costante, dipendente dalla dinamica del sistema o modellato come funzione del rapporto tra segnale e rumore (SNR);

- $H$ : la matrice  $H$ , di dimensioni  $m \times n$ , rappresenta il modello di osservazione, e mette in relazione lo stato con la rilevazione  $z_k$ ;
- $R$ : la *matrice di covarianza della misurazione* indica l'errore intrinseco di una rilevazione. Spesso si assume  $R$  come una matrice diagonale indicando che le rilevazioni dei vari componenti del vettore di stato siano indipendenti tra loro [31].

### Guadagno di Kalman

Il calcolo del guadagno di Kalman si ottiene applicando la formula

$$K_k = \frac{P_k^- H^T}{HP_k^- H^T + R}, \quad (4.9)$$

il cui valore, comparando l'errore di stima  $P_k^-$  con l'errore di rilevazione  $R$ , stabilisce quanto ritenere affidabili la stima predetta e la rilevazione corrente.

### Calcolo del nuovo stato

Questa fase permette, infine, di calcolare la stima dello stato al tempo corrente, correggendo il valore precedentemente previsto con le informazioni aggiuntive fornite dalla rilevazione:

$$x_k = x_k^- + K[z_k - Hx_k^-]. \quad (4.10)$$

### Aggiornamento della matrice di covarianza dello stato

Come ultimo passo, viene calcolata la matrice di covarianza dello stato dell'iterazione corrente definitiva.

$$P_k = (I - KH)P_k^-. \quad (4.11)$$

### Lo stato corrente diventa lo stato precedente

I risultati di questo step,  $x_k$  e  $P_k$ , ora diventano gli input di una nuova iterazione del filtro di Kalman, assunto il ruolo di stato precedente:

$$k \rightarrow k - 1. \quad (4.12)$$

## 4.3 Il tuning dei parametri

La fase di implementazione è seguita da una fase di tuning dei parametri per minimizzare l'errore di stima.

Per quanto riguarda i parametri iniziali di stato  $x_0$  e covarianza dell'errore di stima  $P_0$ , principalmente influenzano solamente la convergenza dell'algoritmo, soprattutto per sistemi LTIV (Linear Time Invariant Systems). In assenza di informazioni a priori, la stima iniziale viene scelta in modo casuale o con valore 0, mentre la matrice di covarianza dell'errore viene impostata come matrice diagonale con grandi valori sulla diagonale. Nel caso dei sistemi lineari, questa scelta influenza la convergenza dell'algoritmo di solo alcune iterazioni; per quanto riguarda i sistemi non lineari, è preferibile disporre di informazioni più precise [17].

Per quanto concerne la matrice di covarianza dei rumori di processo  $Q$ , i cui valori di riferimento sono le deviazioni standard ottenute da prove e verifiche sperimentali con i dati reali, aumentarne i valori comporta un incremento

dell'errore di processo, dove il filtro assegna sempre una maggior importanza alle informazioni fornite dalle misure. Idealmente, con una matrice  $Q$  composta da valori tendenti all'infinito, il filtro restituirebbe le misure ricevute come input.

Infine, la matrice dell'errore di misura  $R$ , i cui elementi diagonali rappresentano la varianza di ogni misura, e gli altri elementi, che rappresentano la correlazione tra i diversi componenti del rumore di misurazione, è spesso ricondotta ad una matrice diagonale, dato che ogni componente del vettore di misura è solitamente indipendente dalle altre. Questi valori possono essere ottenuti ponendo il dispositivo in diverse situazioni statiche, calcolando la varianza sulle diverse misurazioni [31].

## 4.4 Approssimazioni del filtro di Kalman

Il filtro di Kalman assume che le equazioni di stato denotino funzioni lineari e che le misurazioni siano influenzate solamente da rumore Gaussiano; queste ipotesi limitano l'applicabilità del filtro di Kalman, dato che la maggior parte dei sistemi operano nella non linearità. Di seguito, approfondiamo due tecniche basate sul filtro di Kalman che risolvono, in parte, questo problema, rinunciando alla proprietà di ottimalità e fornendo, quindi, stime approssimate, seppur valide: l'*Extended Kalman Filter* (EKF) e l'*Unscented Kalman Filter* (UKF).

### 4.4.1 Extended Kalman Filter

L'Extended Kalman Filter supera l'assunzione di linearità rilassando i requisiti del filtro di Kalman originale: le funzioni di stato e di misura devono essere funzioni *differenziabili*, rendendo l'EKF applicabile a sistemi non lineari. L'EKF, quindi, ottiene una approssimazione di quella che sarebbe la vera belief.

L'EKF *linearizza* il sistema non lineare approssimando il sistema in modo analogo alla *espansione di Taylor del primo ordine* [11]. L'espansione di Taylor di una funzione  $f$  in  $a$  è data dalla formula:

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \dots + \frac{f^{(n)}(a)}{n!}(x-a)^n + \dots, \quad (4.13)$$

quindi, applicando un'espansione del primo ordine, otteniamo

$$f(x) = f(a) + f'(a)(x - a). \quad (4.14)$$

Assumendo che il nostro processo abbia un vettore di stato  $x \in \mathfrak{R}^n$ , in questo caso definito da un'equazione non lineare

$$x_k = f(x_{k-1}, u_k, w_{k-1}), \quad (4.15)$$

e delle rilevazioni  $z_k \in \mathfrak{R}^m$  modellate come

$$z_k = h(k_k, v_k), \quad (4.16)$$

la linearizzazione avviene attraverso le seguenti equazioni

$$F_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}, \quad (4.17)$$

$$G_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}, \quad (4.18)$$

dove  $F$  e  $G$  rappresentano le matrici Jacobiane delle funzioni  $f$  e  $h$  rispettivamente. Il processo di linearizzazione approssima  $f$  e  $h$  a funzioni lineari tangenti alle stesse, quindi le meccaniche di propagazione della belief sono equivalenti a quelle del filtro di Kalman per sistemi lineari [33, 17].

#### 4.4.2 Unscented Kalman Filter

L'Unscented Transform (UT) è un metodo utilizzato in analisi numerica per stimare l'effetto dell'applicazione di una trasformazione non lineare ad una distribuzione di probabilità: l'idea dietro l'UT consiste nello scegliere deterministicamente un certo numero di punti campione, chiamati *punti sigma*, che rappresentino la distribuzione dello stato  $x$ ; dopodichè, questi punti verranno propagati attraverso il sistema non lineare, ottenendo la stima della media e della covarianza della variabile trasformata [23].

L'Unscented Kalman Filter (Julier et al. 1997) implementa la UT all'interno di un processo di filtraggio di Kalman: la distribuzione dello stato è ancora approssimata a una GRV; tuttavia, non utilizza la momentum representation, come nel caso dell'EKF, ma una rappresentazione basata su *punti sigma*. Nel momento in cui questi punti sono propagati sul sistema

non lineare reale, permettono di calcolare il vero valor medio e la vera covarianza della distribuzione fino al terzo ordine dello sviluppo in serie di Taylor [29, 17]. Da notare che, a differenza dell'EKF, non tenta di approssimare la funzione non lineare, ma calcola direttamente una approssimazione della media e della covarianza della distribuzione obiettivo.

Il processo di stima segue i seguenti passi:

### Valori iniziali

Il primo passo viene eseguito solamente una volta, e permette di calcolare i valori costanti dei pesi  $\mathbf{W}_i^{(m)}$  e  $\mathbf{W}_i^{(c)}$ :

$$\mathbf{W}_0^{(m)} = \frac{\lambda}{n + \lambda}, \quad (4.19)$$

$$\mathbf{W}_0^{(c)} = \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta), \quad (4.20)$$

$$\mathbf{W}_i^{(m)} = \frac{\lambda}{2(n + \lambda)}, \quad (4.21)$$

$$\mathbf{W}_i^{(c)} = \frac{\lambda}{2(n + \lambda)}, \quad (4.22)$$

dove  $i = 0, 1, \dots, 2n$ . Il parametro  $\alpha$  determina la propagazione dei punti sigma attorno alla media della distribuzione ed è solitamente settato con un basso valore positivo come  $10^{-3}$ . Il parametro  $\beta$ , invece, è utilizzato per incorporare la prior knowledge della distribuzione di  $x$ ; per le distribuzioni Gaussiane il valore ottimale è  $\beta = 2$ .

### Predizione - Formazione dei punti sigma

Durante questa fase viene generato un insieme di  $2n + 1$  punti sigma:

$$\boldsymbol{\chi}_{k-1}^{(0)} = \mathbf{m}_{k-1}, \quad (4.23)$$

$$\boldsymbol{\chi}_{k-1}^i = \mathbf{m}_{k-1} + (\sqrt{(n + \lambda)\mathbf{P}_{k-1}})_i, \quad (4.24)$$

$$\boldsymbol{\chi}_{k-1}^{i+n} = \mathbf{m}_{k-1} - (\sqrt{(n + \lambda)\mathbf{P}_{k-1}})_i, \quad (4.25)$$

con  $\mathbf{m} \in \mathbb{R}^n$ ,  $i = 1, 2, \dots, n$  e dove  $\lambda$  è definito come

$$\lambda = \alpha^2(n + \kappa) - n. \quad (4.26)$$

Nella Equazione 4.26, il parametro  $\kappa$  è un altro parametro di scala spesso settato a 0.

### Predizione - Propagazione dei punti sigma attraverso il modello dinamico

La propagazione dei punti sigma attraverso la funzione non-lineare  $\mathbf{f}$  restituisce l'insieme dei punti trasformati, denotato come  $\hat{\boldsymbol{\chi}}_k^{(i)}$ :

$$\hat{\boldsymbol{\chi}}_k^{(i)} = \mathbf{f}(\boldsymbol{\chi}_{k-1}^i), \quad (4.27)$$

dove  $i = 0, 1, \dots, 2n$ .

### Predizione - Calcolo del valor medio e della covarianza predetti

La ricombinazione dei sigma points produce la media e la covarianza “predette” dello stato:

$$\mathbf{m}_k^- = \sum_{i=0}^{2n} \mathbf{W}_i^{(m)} \hat{\boldsymbol{\chi}}_k^{(i)}, \quad (4.28)$$

$$\mathbf{P}_k^- = \sum_{i=0}^{2n} \mathbf{W}_i^{(c)} (\hat{\boldsymbol{\chi}}_k^{(i)} - \mathbf{m}_k^-) (\hat{\boldsymbol{\chi}}_k^{(i)} - \mathbf{m}_k^-)^T + \mathbf{Q}_{k-1}. \quad (4.29)$$

### Correzione - Formazione dei punti sigma

$$\boldsymbol{\chi}_k^{-(0)} = \mathbf{m}_k^- \quad (4.30)$$

$$\boldsymbol{\chi}_k^{-(i)} = \mathbf{m}_k^- + (\sqrt{(n+\lambda)\mathbf{P}_k})_i \quad (4.31)$$

$$\boldsymbol{\chi}_k^{-(i+n)} = \mathbf{m}_k^- - (\sqrt{(n+\lambda)\mathbf{P}_k})_i \quad (4.32)$$

con  $i = 1, 2, \dots, n$

### Correzione - Propagazione dei punti sigma attraverso il modello di misura

I sigma points vengono proiettati sulla measurement function  $\mathbf{h}$ :

$$\hat{\boldsymbol{\gamma}}_k^{(i)} = \mathbf{h}(\boldsymbol{\chi}_k^{-(i)}) \quad (4.33)$$

dove  $i = 0, 1, \dots, 2n$ .

**Correzione - Calcolo della media prevista, della matrice di covarianza della misura prevista  $S_k$ , e la matrice di covarianza incrociata tra lo stato e la misurazione  $C_k$**

La ricombinazione dei punti sigma produce la misura prevista e la covarianza di misura prevista:

$$\boldsymbol{\mu}_k = \sum_{i=0}^{2n} \mathbf{W}_i^{(m)} \hat{\boldsymbol{\gamma}}_k^{(i)}, \quad (4.34)$$

$$\mathbf{S}_k = \sum_{i=0}^{2n} \mathbf{W}_i^{(c)} (\hat{\boldsymbol{\gamma}}_k^{(i)} - \boldsymbol{\mu}_k) (\hat{\boldsymbol{\gamma}}_k^{(i)} - \boldsymbol{\mu}_k)^T + \mathbf{R}_k. \quad (4.35)$$

Inoltre, la matrice di covarianza incrociata è data da:

$$\mathbf{C}_k = \sum_{i=0}^{2n} \mathbf{W}_i^{(c)} (\boldsymbol{\chi}_k^{-(i)} - \mathbf{m}_k^-) (\boldsymbol{\gamma}_k^{-(i)} - \boldsymbol{\mu}_k)^T. \quad (4.36)$$

**Correzione - Calcolo del guadagno del filtro, della media e della covarianza filtrate, condizionate dal valore della rilevazione**

Come nel caso del filtro di Kalman per sistemi lineari, il guadagno di Kalman è dato da:

$$\mathbf{K}_k = \mathbf{C}_k \mathbf{S}_k^{-1}, \quad (4.37)$$

grazie al quale calcoliamo la media corretta

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k [\mathbf{z}_k - \boldsymbol{\mu}_k] \quad (4.38)$$

e la relativa covarianza

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T. \quad (4.39)$$



# Capitolo 5

## Posizione

Data l'importanza della posizione corrente nelle applicazioni context-aware, in letteratura sono state ampiamente documentate varie tecniche di localizzazione; tuttavia, la maggior parte non fornisce risultati soddisfacenti in contesti indoor o non è direttamente implementabile sui dispositivi indossabili più comunemente diffusi. Ad esempio:

- le tecniche di fingerprinting su WLAN offrono una precisione inversamente proporzionale all'area esaminata;
- le tecniche di visione non risultano efficaci in ambienti poco illuminati;
- i sensori inerziali non sono autonomamente sufficienti a fornire una stima precisa senza il supporto di altre tecnologie, data la divergenza degli errori dovuta ai vari passi di integrazione dei valori di accelerometro e giroscopio.

Di seguito, esaminiamo due tecnologie di localizzazione scelte per le loro caratteristiche. In particolar modo:

- una tecnica basata sulle *differenze di campo magnetico*, dato il basso costo computazionale, requisito fondamentale per applicazioni real-time sviluppate su piattaforme mobile con limitata capacità computazionale;
- l'installazione di ripetitori GPS, data l'ampia documentazione in letteratura sulla tecnologia GNSS.

## 5.1 Magnetometer Measurement Difference

L'impiego di magnetometri come tecnica di localizzazione in contesti indoor offre notevoli vantaggi, nonostante l'utilizzo di componenti economici introduca degli errori non trascurabili sulla stima finale.

Sebbene non costituisca una tecnica ad hoc e sia ancora soggetta a diversi studi, si sta diffondendo l'uso di un approccio che non soffre delle limitazioni dell'approccio convenzionale: stiamo parlando di una tecnica basata sulle differenze di campo magnetico. Questo approccio sta dando risultati promettenti rispetto alle tecniche convenzionali di pattern matching sui valori di intensità di campo magnetico. La proposta di Kim e Kong (2015) si basa su tecniche MMD applicate ai meccanismi di predizione-correzione dei filtri Bayesiani, per ridurre l'incertezza introdotta dal *bias* [9].

### 5.1.1 Il magnetometro come strumento per la localizzazione

Le classiche tecniche basate sui magnetometri applicano azioni di pattern matching tra le rilevazioni di intensità del campo magnetico e una mappa di rilevazioni reali pre-registrata; questo porta dei vantaggi in termini di complessità computazionale, ma, data l'ampia diffusione di sensori economici su una vasta gamma di dispositivi mobili, anche le tecniche basate sui magnetometri soffrono degli errori introdotti dal *bias*.

La rilevazione di un magnetometro può essere modellata come

$$R_A(p) = (X(p) + X_b)^2 + (Y(p) + Y_b)^2 + (Z(p) + Z_b)^2 = A(p) + \gamma_A(p), \quad (5.1)$$

dove  $X(p)$ ,  $Y(p)$  e  $Z(p)$  rappresentano i valori reali di intensità di campo magnetico nel punto  $p$  lungo i rispettivi assi  $x$ ,  $y$  e  $z$ ; a questi viene sommato l'effetto del bias  $X_b$ ,  $Y_b$  e  $Z_b$ . Quindi  $R_A(p)$  può essere idealmente suddiviso in due parti:

$$S_A(p) = X^2(p) + Y^2(p) + Z^2(p) \quad (5.2a)$$

dove  $S_A(p)$  indica la mappa reale del campo magnetico, e

$$\gamma_A(p) = 2(X_b X(p) + Y_b Y(p) + Z_b Z(p)) + X_b^2 + Y_b^2 + Z_b^2 \quad (5.2b)$$

dove  $\gamma_A(p)$  rappresenta l'effetto del bias. L'errore  $\gamma_A(p)$  introdotto dal bias non segue né un andamento costante né lineare, per cui la tecnica di pattern

matching che calcola il grado di correlazione tra  $R_A$  e  $S_A$  può non dare i risultati previsti.

### 5.1.2 Approccio proposto

La tecnica che andremo ad analizzare si propone di mitigare l'incertezza introdotta dal bias attraverso l'applicazione di meccanismi di pattern matching sulle *differenze consecutive di campo magnetico*, in inglese Magnetometer Measurement Difference (MMD), applicate ai concetti di un filtro Bayesiano [9].

La mappa, secondo questo approccio, registra, per ogni punto, la differenza di intensità di campo magnetico lungo 8 direzioni: l'asse  $x$  positivo, l'asse  $x$  negativo, l'asse  $y$  positivo, l'asse  $y$  negativo, e le 4 direzioni diagonali tra gli assi. Quindi

$$\begin{aligned} S_M^{+x} &= X^2(x+1, y) + Y^2(x+1, y) + Z^2(x+1, y) \\ &\quad - X^2(x, y) - Y^2(x, y) - Z^2(x, y) \end{aligned} \quad (5.3a)$$

$$\begin{aligned} S_M^{-x} &= X^2(x-1, y) + Y^2(x-1, y) + Z^2(x-1, y) \\ &\quad - X^2(x, y) - Y^2(x, y) - Z^2(x, y) \end{aligned} \quad (5.3b)$$

$$\begin{aligned} S_M^{+y} &= X^2(x, y+1) + Y^2(x, y+1) + Z^2(x, y+1) \\ &\quad - X^2(x, y) - Y^2(x, y) - Z^2(x, y) \end{aligned} \quad (5.3c)$$

$$\begin{aligned} S_M^{-y} &= X^2(x, y-1) + Y^2(x, y-1) + Z^2(x, y-1) \\ &\quad - X^2(x, y) - Y^2(x, y) - Z^2(x, y) \end{aligned} \quad (5.3d)$$

$$\begin{aligned} S_M^{+x+y} &= X^2(x+1, y+1) + Y^2(x+1, y+1) + Z^2(x+1, y+1) \\ &\quad - X^2(x, y) - Y^2(x, y) - Z^2(x, y) \end{aligned} \quad (5.3e)$$

$$\begin{aligned} S_M^{-x+y} &= X^2(x-1, y+1) + Y^2(x-1, y+1) + Z^2(x-1, y+1) \\ &\quad - X^2(x, y) - Y^2(x, y) - Z^2(x, y) \end{aligned} \quad (5.3f)$$

$$\begin{aligned} S_M^{+x-y} &= X^2(x+1, y-1) + Y^2(x+1, y-1) + Z^2(x+1, y-1) \\ &\quad - X^2(x, y) - Y^2(x, y) - Z^2(x, y) \end{aligned} \quad (5.3g)$$

$$\begin{aligned} S_M^{-x-y} &= X^2(x-1, y-1) + Y^2(x-1, y-1) + Z^2(x-1, y-1) \\ &\quad - X^2(x, y) - Y^2(x, y) - Z^2(x, y) \end{aligned} \quad (5.3h)$$

Una MMD può essere modellata come:

$$\begin{aligned} R_M(p) &= (X(p+1) + X_b)^2 + (Y(p+1) + Y_b)^2 + (Z(p+1) + Z_b)^2 \\ &\quad - (X(p) + X_b)^2 - (Y(p) + Y_b)^2 - (Z(p) + Z_b)^2 \\ &= M(p) + \gamma_M(p), \end{aligned} \quad (5.4)$$

dove

$$S_M(p) = X^2(p+1) + Y^2(p+1) + Z^2(p+1) - X^2(p) - Y^2(p) - Z^2(p) \quad (5.5a)$$

$$\begin{aligned} \gamma_M(p) &= 2(X_b(X(p+1) - X(p)) + Y_b(Y(p+1) - Y(p)) \\ &\quad + Z_b(Z(p+1) - Z(p))). \end{aligned} \quad (5.5b)$$

$S_M(p)$  rappresenta un differenza di intensità di campo magnetico senza l'effetto del bias.

Sebbene l'effetto del bias non sia completamente eliminato, la tecnica basata su MMD si è dimostrata più robusta della tecnica convenzionale.

### 5.1.3 Algoritmo

La tecnica basata sulle differenze di intensità di campo magnetico si fonda su 4 presupposti:

- i disturbi delle rilevazioni siano approssimabili a rumore bianco con distribuzione GaussianA (AWGN);
- la mappa raccolga le MMD nelle 8 direzioni su punti distanti 50 cm tra loro, assumendo una velocità costante dell'utente di circa 0.6 m/s e una lunghezza di correlazione di 7 s, che equivale a circa a 8 campioni;
- la direzione di movimento sia fornita come input. Il movimento viene approssimato a uno spostamento lungo gli assi  $x$  e  $y$  e le diagonali, ottenendo un totale di 8 possibili direzioni di movimento;
- dopo un cambio di direzione, si assume un movimento in linea retta della stessa durata della lunghezza di correlazione.

### Valori iniziali

La matrice predetta delle correlazioni  $C_p$ , di dimensioni  $x_N \times y_N$  equivalenti al numero di campioni raccolti ogni 50 cm lungo ogni lato dell'area obiettivo, è settata inizialmente come matrice di soli 1

$$C_p = \mathbf{1}_{x_N \times y_N}. \quad (5.6)$$

### Predizione

Attraverso la seguente equazione

$$\begin{aligned} C_p(x, y) = & W_1^p C_c(x + 1, y) + W_2^p C_c(x - 1, y) + W_3^p C_c(x, y + 1) \\ & + W_4^p C_c(x, y - 1) + W_5^p C_c(x + 1, y + 1) \\ & + W_6^p C_c(x + 1, y - 1) + W_7^p C_c(x - 1, y + 1) \\ & + W_8^p C_c(x - 1, y - 1), \end{aligned} \quad (5.7)$$

è possibile calcolare la correlazione predetta di ogni punto: questo avviene sommando i vari valori di correlazione dei punti circostanti opportunamente pesati attraverso i coefficienti  $W_i^p$ , dando un maggior peso ai punti nella direzione di movimento, ad esempio 1/2, mentre gli altri saranno pesati con un valore inferiore, ad esempio 1/14.

La fase di predizione non va effettuata nel caso sia la prima iterazione dell'algoritmo, dato che non ci sono ancora dati disponibili.

### Correzione - Calcolo della correlazione sui dati rilevati

In seguito a una nuova rilevazione, viene calcolata la correlazione tra la mappa delle differenze precedentemente precaricata e i dati MMD ottenuti sulle varie misurazioni. Quindi, per ogni punto della mappa, viene calcolata la correlazione lungo ognuna delle 8 direzioni (ad esempio lungo l'asse  $x$  positivo):

$$C_m^{+x}(k) = \frac{\sum_{p=0}^{N-1} (R_M(p) - \bar{m}^{+x}) + (S_M^{+x}(k+p) - \bar{m}^{+x})}{\sqrt{\sum_{p=0}^{N-1} (R_M(p) - \bar{m}^{+x})^2 \sum_{p=0}^{N-1} (S_M^{+x}(k+p) - \bar{m}^{+x})^2}} \quad (5.8a)$$

dove

$$\bar{m}^{+x} = \frac{\sum_{p=0}^{N-1} (R_M(p) + S_M^{+x}(p))}{2N}; \quad (5.8b)$$

successivamente viene calcolata la correlazione sulla rilevazione  $C_m$  pesando le correlazioni lungo le varie direzioni calcolate precedentemente attraverso i parametri  $W_i^m$ , in modo analogo al passo di predizione:

$$\begin{aligned} C_m(x, y) = & W_1^m C_m^{+x}(x, y) + W_2^p C_m^{-x}(x, y) + W_3^p C_m^{+y}(x, y) \\ & + W_4^p C_m^{-y}(x, y) + W_5^p C_m^{+x+y}(x, y) + W_6^p C_m^{+x-y}(x, y) \\ & + W_7^p C_m^{-x+y}(x, y) + W_8^p C_m^{-x-y}(x, y). \end{aligned} \quad (5.9)$$

### Correzione - Calcolo della correlazione a posteriori

Per la teoria dei filtri Bayesiani, la stima corretta della correlazione all'iterazione corrente è data da

$$C_c(x, y) = C_p(x, y) C_m(x, y). \quad (5.10)$$

#### 5.1.4 Osservazioni

L'ipotesi di movimento in linea retta dopo ogni variazione di direzione rappresenta una limitazione considerevole, soprattutto se applicata al contesto della realtà aumentata; inoltre, l'algoritmo non opera bene ai bordi della mappa, quindi sono necessarie considerazioni aggiuntive sul valore di correlazione dei punti più esterni.

## 5.2 GPS Indoor

Un Sistema Satellitare Globale di Navigazione, in inglese Global Navigation Satellite System (GNSS), permette la geolocalizzazione di un dispositivo ricevitore attraverso l'impiego di satelliti in orbita terrestre.

### 5.2.1 Introduzione ai sistemi GNSS

I sistemi di geolocalizzazione con copertura globale attualmente in funzione sono:

- *Global Positioning System (GPS)*: di proprietà del Dipartimento della Difesa degli Stati Uniti d'America e completamente operativo dal 1995, impiegava in origine una costellazione di 24 satelliti, di seguito ampliata a 32 di cui 31 effettivamente in funzione. Requisito fondamentale per il funzionamento di un sistema GNSS è disponibilità di una linea di vista tra il punto del terreno da geolocalizzare e almeno 4 satelliti; l'ampliamento della costellazione da 24 a 32 satelliti ha portato ad un incremento della precisione, dato che sono visibili circa 9 satelliti da ogni posizione che forniscono dati ridondanti utilizzati nel processo di localizzazione. Il sistema GPS permette la localizzazione con un raggio di precisione di 5 metri.
- *GLONASS*: gestito dalla difesa aerospaziale russa, è stato ripristinato e reso operativo per il posizionamento globale nel 2011. Il sistema vanta una costellazione di 27 satelliti di cui 24 attualmente in funzione. Il calcolo della posizione mostra un errore compreso tra i 4 e gli 8 metri.
- *Galileo*: commissionato dall'Unione Europea e attualmente in lavorazione, il sistema Galileo sarà pienamente operativo dal 2020. Sarà formato da 30 satelliti e permetterà una precisione di 1 metro per la pubblica utenza, e fino ad 1 centimetro per i privati. Fornisce funzionalità aggiuntive rispetto alla controparte americana, come l'invio del messaggio di integrità, ma sarà comunque pienamente compatibile con il sistema GPS; quindi, i due sistemi potranno essere utilizzati congiuntamente.

### 5.2.2 Il calcolo della posizione

Per il calcolo della posizione del ricevitore, si utilizzano i principi della triangolazione sulla superficie terrestre: considerando 3 punti  $A$ ,  $B$  e  $C$  e conoscendo la relativa distanza della posizione corrente da ciascuno di essi, basterà risolvere il sistema a tre equazioni (ovvero le equazioni delle circonferenze con centro  $A$ ,  $B$  e  $C$  e raggio equivalente alla relativa distanza da ciascun punto) e 2 incognite per ottenere la posizione corrente: due equazioni non sono sufficienti dato che le equazioni sono di secondo grado e, quindi, la risoluzione del sistema darebbe come risultato due soluzioni. Nel caso di un sistema GPS, il ricevitore calcola il tempo di arrivo dei segnali provenienti da diversi satelliti e ottiene la sua posizione attraverso queste

rilevazioni. Più precisamente applica una *triangolazione* sulle distanze dai vari satelliti utilizzati come punti di riferimento, la cui posizione è nota a priori: ogni satellite emette periodicamente un segnale, includendo un identificativo univoco per satellite, la propria posizione nell'orbita terrestre e l'orario di emissione del segnale. Nel caso gli orologi interni dei satelliti e del dispositivo ricevente siano sincronizzati, è possibile ottenere una stima del tempo di percorrenza del segnale  $\Delta T$  sottraendo l'orario di trasmissione all'orario di trasmissione del segnale; tenendo a mente che la velocità di propagazione delle onde radio equivale alla velocità della luce

$$c = 299792 \text{ km/s} \sim 300000 \text{ km/s}, \quad (5.11)$$

possiamo calcolare la distanza che separa il ricevitore da ogni satellite di interesse con

$$d = c \Delta T. \quad (5.12)$$

La distanza così calcolata identifica la superficie di una sfera di raggio  $d$  di cui la posizione del satellite in questione occupa l'origine. Per una localizzazione che includa anche l'altitudine, sono necessarie le informazioni relative a 4 satelliti, visto che:

- una triangolazione su 3 satelliti restituisce due soluzioni, sebbene spesso una delle due possa essere scartata, dato che non identifica un punto sulla superficie terrestre;
- gli orologi presenti sui dispositivi sono economici e, quindi, con un margine di errore più alto.

### 5.2.3 Utilizzo dei segnali GNSS per la localizzazione in contesti indoor

Il progresso tecnologico nel campo dei sistemi GNSS ha condotto allo sviluppo di ricevitori sempre più sensibili e la realizzazione di diversi sistemi GNSS fornisce una vasta disponibilità di satelliti con un elevato grado di precisione: tuttavia, operando nei contesti indoor, il segnale non raggiunge il ricevitore direttamente, ma subisce delle alterazioni dovute soprattutto agli ostacoli fisici che deve “oltrepassare”, in particolare [6]:

- *Attenuazione di spazio libero*: in inglese *path loss*, indica l'attenuazione della potenza di un segnale elettromagnetico causata dalla dissipazione durante la propagazione all'interno di un canale di comunicazione (spesso l'aria) senza incontrare alcun tipo di ostacolo. Ha un effetto non trascurabile solo sulle lunghe distanze (100-1000 m) ed è rappresentato da modelli deterministici;
- *Shadowing e Multipath*: il segnale si propaga attraverso 4 meccanismi:
  - Assorbimento: quando un segnale attraversa un ostacolo, la sua potenza ne risulta attenuata e il segnale leggermente curvato;
  - Riflessione: si verifica quando il segnale colpisce una superficie liscia molto ampia rispetto alla propria lunghezza d'onda;
  - Diffrazione: curvatura del segnale che avviene quando colpisce il bordo di un oggetto di grandi dimensioni comparate alla propria lunghezza d'onda;
  - Dispersione: avviene quando il segnale colpisce una superficie ruvida, le cui irregolarità sono dello stesso ordine di grandezza della lunghezza d'onda o inferiori.

Lo shadowing indica la variazione della potenza del segnale a causa di ostacoli sul percorso di propagazione. Dato che gli ostacoli possono variare per posizione, dimensioni e proprietà dielettriche, questo effetto è spesso modellato attraverso modelli statistici. Un altro effetto dovuto a ostacoli durante la propagazione del segnale, soprattutto in contesti indoor, è il multipath: quando il segnale incontra degli ostacoli, a seconda del tipo di ostacolo, può essere riflesso, diffratto o disperso, producendo delle copie con una potenza minore rispetto all'originale, sfasate in fase o frequenza e giungere a destinazione in ritardo rispetto alla componente principale. Le varie componenti vengono sommate dal ricevente, ottenendo un segnale distorto rispetto a quello trasmesso;

- *Mobilità*: se il ricevitore si trova in uno stato di moto, osserverà un cambiamento nella frequenza di trasmissione del segnale dovuta al movimento relativo tra il trasmettitore e il ricevitore, chiamato *effetto Doppler*.

Dato che, per la localizzazione indoor, non è possibile fare affidamento solamente su segnali inviati direttamente dai satelliti, si rende necessario valutare ipotesi alternative, come l'installazione di alcuni ripetitori, o anche di pseudoliti (pseudo-satelliti). In particolare, trattando della propagazione dei segnali GNSS attraverso gli edifici, sottolineiamo che le caratteristiche di propagazione dipendono dall'angolo di incidenza con cui il segnale attraversa la superficie dell'edificio e, anche, dall'elevazione del ripetitore che l'ha trasmessa [7].

Inoltre, sebbene mostrino un effetto relativamente ridotto sulla modifica della forma dell'onda, il materiale utilizzato per la costruzione degli edifici e la loro architettura influenza la potenza del segnale ricevuta. Sperimentalmente è stato dimostrato che, quando un segnale si scontra con materiali legnosi, essi mostrano un coefficiente di trasmissione molto alto mentre ne riflettono solo una piccola parte. Al contrario, il vetro colorato ha un alto coefficiente di riflessione. Se andiamo a considerare l'attenuazione causata in sé dall'attraversamento del segnale in uno specifico materiale, i materiali leggeri come il legno comportano una attenuazione limitata, mentre all'interno dei materiali più voluminosi, come il cemento, avviene una attenuazione consistente. Combinando il coefficiente di trasmissione con quello di attenuazione del materiale, si può determinare l'indebolimento complessivo del segnale causato da uno specifico materiale.

Un altro fattore da tenere in considerazione è la frequenza di trasmissione: comparando i risultati trasmettendo sia su banda L1 (1575.42 MHz) che su banda L5/E5 (1176 MHz), sebbene i coefficienti di trasmissione e di riflessione si possano ritenere indipendenti dalla frequenza portante, i materiali con un elevato coefficiente di attenuazione mostrano un'attenuazione aggiuntiva se attraversati da segnali ad alta frequenza, molto più accentuato che da quelli a frequenza minore [8].

In conclusione, la localizzazione all'interno degli edifici attraverso segnali GNSS è subordinata a:

- i materiali da costruzione utilizzati;
- la posizione di installazione dei ripetitori e la frequenza portante utilizzata dalla trasmissione, che devono essere valutati sul caso specifico.

L'uso del GNSS in contesti indoor può essere un componente estremamente utile per fornire aggiornamenti sulla posizione combinati ai valori di un IMU per soluzioni basate su *sensor fusion* [14].

# Capitolo 6

## Orientamento

Dopo la posizione, la seconda informazione di importanza rilevante nelle applicazioni context-aware è l'orientamento del dispositivo che, nel caso di dispositivi HMD, si traduce nel cosiddetto *head-tracking*.

### 6.1 Rappresentazione dell'orientamento

Esistono diversi approcci e convenzioni per rappresentare i movimenti rotazionali di un oggetto: di seguito prendiamo in analisi due tra gli approcci più utilizzati con i sensori inerziali [13].

#### 6.1.1 Angoli di Eulero

Gli Angoli di Eulero costituiscono la maniera più intuitiva per la rappresentazione dell'orientamento di un corpo rigido rispetto ad un sistema di riferimento *fisso*: gli angoli di Eulero possono essere espressi componendo le tre rotazioni fondamentali sugli assi di un sistema di riferimento:

$$\Phi = [\varphi, \theta, \psi]. \quad (6.1)$$

Nella convenzione di Tait-Bryan gli angoli, e di conseguenza gli assi di riferimento, prendono il nome dal gergo aeronautico [32, 4, 18]:

- *Roll*: l'angolo di rollio  $\varphi$  descrive la rotazione sull'asse  $x$ ;
- *Pitch*: l'angolo di beccheggio  $\theta$  descrive la rotazione sull'asse  $y$ ;

- *Yaw*: l'angolo di imbardata  $\psi$  descrive la rotazione sull'asse  $z$ .

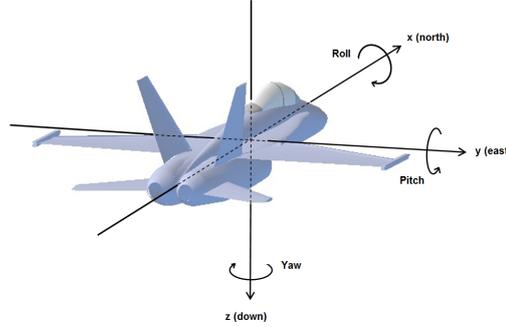


Figura 6.1: Sistema di riferimento: convenzione di Tait-Bryan.

Quando il corpo rigido si trova nella situazione iniziale, quindi allineato con il sistema di riferimento fisso, l'asse  $x$  punta a nord, l'asse  $y$  a est e l'asse  $z$  verso il basso. Gli angoli  $\varphi$  e  $\psi$  possono assumere valori compresi tra  $-\pi$  e  $\pi$ , mentre l'angolo  $\theta$  solamente tra  $-\frac{\pi}{2}$  e  $\frac{\pi}{2}$ . Tuttavia, per  $\theta = \pm\frac{\pi}{2}$ , gli angoli di pitch e roll coincidono, perdendo la possibilità di ruotare secondo un grado di libertà: questo fenomeno è noto come *gymbal lock* [18, 32, 12]. Sebbene questa singolarità causi problemi per varie tipologie di situazioni, come quelle aeronautiche, è ritenuta più che accettabile nella maggior parte delle applicazioni terrestri, dato che angoli di pitch di 90 gradi non rientrano nelle normali situazioni operative.

Le tre rotazioni elementari sono date dalle seguenti matrici di rotazione:

$$R_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi \\ 0 & -\sin \varphi & \cos \varphi \end{bmatrix} \quad (6.2a)$$

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (6.2b)$$

$$R_z(\varphi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.2c)$$

Come abbiamo già detto, ogni rotazione può essere ottenuta da una combinazione delle tre rotazioni elementari, quindi la matrice di rotazione in grado di rappresentare qualunque rotazione (secondo il formalismo scelto) è data da:

$$R_w^b(\Phi) = R_z(\psi)R_y(\theta)R_x(\varphi) = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ -\sin \varphi \sin \theta \cos \psi - \cos \varphi \sin \psi & \sin \varphi \sin \theta \sin \psi + \cos \theta \cos \psi & \sin \varphi \cos \theta \\ \cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \psi & \cos \varphi \sin \theta \sin \psi - \sin \varphi \sin \psi & \cos \varphi \cos \theta \end{bmatrix} \quad (6.3)$$

Questa matrice di rotazione vale per i sistemi di riferimento *right-handed*, dove prima viene eseguita la rotazione sull'asse  $z$ , successivamente sull'asse  $y$  e, infine, sull'asse  $x$ .

### 6.1.2 Quaternioni

Un quaternione è un elemento quadridimensionale definibile come

$$\mathbf{q} = [q_0, q_1, q_2, q_3]^T \quad (6.4a)$$

oppure

$$\mathbf{q} = q_0 + q_1i, q_2j, q_3k \quad (6.4b)$$

dove  $i$ ,  $j$  e  $k$  hanno caratteristiche simili a quelle di  $i$  del piano complesso bidimensionale [32, 18].

Un quaternione con norma unitaria è chiamato *quaternione unitario*, e può essere usato per rappresentare l'orientamento di un corpo rigido [18].

Per ottenere un quaternione dai valori degli angoli di Eurlero di utilizzano

le seguenti formule di conversione [32, 3]:

$$\begin{aligned}
 q_0 &= \cos \frac{\psi}{2} \cos \frac{\theta}{2} \cos \frac{\phi}{2} + \sin \frac{\psi}{2} \sin \frac{\theta}{2} \sin \frac{\phi}{2} \\
 q_1 &= \cos \frac{\psi}{2} \cos \frac{\theta}{2} \sin \frac{\phi}{2} - \sin \frac{\psi}{2} \sin \frac{\theta}{2} \cos \frac{\phi}{2} \\
 q_2 &= \cos \frac{\psi}{2} \sin \frac{\theta}{2} \sin \frac{\phi}{2} + \sin \frac{\psi}{2} \cos \frac{\theta}{2} \sin \frac{\phi}{2} \\
 q_3 &= \sin \frac{\psi}{2} \cos \frac{\theta}{2} \cos \frac{\phi}{2} - \cos \frac{\psi}{2} \sin \frac{\theta}{2} \sin \frac{\phi}{2}
 \end{aligned} \tag{6.5}$$

Per ottenere i valori degli angoli di Eulero corrispondenti ad uno specifico quaternion, invece [32, 3]:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan \frac{2(q_0 q_1 + q_2 q_3)}{1 - 2(q_1^2 + q_2^2)} \\ \arcsin 2(q_0 q_2 - q_3 q_1) \\ \arctan \frac{2(q_0 q_3 + q_1 q_2)}{1 - 2(q_2^2 + q_3^2)} \end{bmatrix} \tag{6.6}$$

Da notare che le funzioni  $\arctan$  e  $\arcsin$  dei linguaggi di programmazione normalmente restituiscono risultati compresi tra  $-\pi/2$  and  $\pi/2$ , e per tre rotazioni tra  $-\pi/2$  e  $\pi/2$  non è possibile ottenere tutti gli orientamenti: per ovviare a questo problema è necessario utilizzare le funzioni  $\text{atan2}$  e  $\text{asin2}$ . Sebbene attraverso i quaternioni possano essere rappresentate tutte le rotazioni, i parametri dei quaternioni unitari non hanno alcun significato fisico evidente; inoltre, il quaternion deve avere norma unitaria per simboleggiare una rotazione pura (vincolo non codificabile nei problemi di ottimizzazione) [4].

## 6.2 I sensori inerziali

Un *Inertial Measurement Unit (IMU)* è un dispositivo elettronico in grado di misurare la forza applicata ad un corpo, la sua velocità angolare e, in certi casi, anche l'intensità del campo magnetico che lo circonda, combinando i risultati di diversi sensori:

- **Accelerometro:** dispositivo che misura l'accelerazione di un corpo lungo le 3 dimensioni di riferimento (x,y e z);

- **Giroscopio:** permette di calcolare la velocità angolare rispetto ai tre assi di riferimento;
- **Magnetometro:** sebbene non misuri alcuna forma di “inerzia”, restituisce la forza del campo magnetico a cui è sottoposto il dispositivo.

### 6.2.1 Calcolo dell’orientamento in situazioni statiche

Denotando il valore dell’accelerazione fornito dall’accelerometro con  $acc^b$ , è possibile ottenere gli angoli di roll  $\varphi$  e pitch  $\theta$  grazie alla direzione della forza gravità, quindi [33, 20]:

$$\varphi^w = \text{atan}\left(\frac{acc_y^b}{acc_z^b}\right), \quad (6.7)$$

$$\theta^w = \text{atan}\left(\frac{-acc_x^b}{acc_y^b \sin \varphi + acc_z^b \cos \varphi}\right). \quad (6.8)$$

Questo perchè, in situazioni statiche, l’unica accelerazione è data dalla forza di gravità.

Per quanto riguarda il calcolo dell’angolo di yaw, è necessario utilizzare l’intensità del campo magnetico fornita dal magnetometro, denotata come  $mag^b$ . Utilizzando i valori di roll e pitch precedentemente calcolati, è possibile trasformare il vettore campo magnetico fornito dal magnetometro dal sistema di riferimento ruotato del dispositivo al world frame, ottenendo il vettore intensità di campo magnetico  $B^{hor}$  quando il dispositivo si trova su un piano orizzontale [19]:

$$\begin{aligned} B^{hor} &= [B_x^{hor}, B_y^{hor}, B_z^{hor}]^T \\ &= R_y(-\theta)R_x(-\varphi)[mag_x^b, mag_y^b, mag_z^b]^T. \end{aligned} \quad (6.9)$$

Quindi, l’angolo di yaw può essere calcolato come

$$\begin{aligned} \psi^w &= \text{atan}\frac{-B_y^{hor}}{B_x^{hor}} \\ &= \text{atan}\left(\frac{-\cos \varphi mag_y^b + \sin \varphi mag_z^b}{\cos \theta mag_x^b + \sin \varphi \sin \theta mag_y^b + \cos \varphi \sin \theta mag_z^b}\right). \end{aligned} \quad (6.10)$$

In sintesi, la combinazione di accelerometro e magnetometro, spesso chiamata *bussola digitale*, può determinare l'orientamento del dispositivo in situazioni statiche, misurando l'accelerazione di gravità e il vettore campo magnetico.

### 6.2.2 Velocità angolare

Le velocità angolari nel body frame  $\omega^b$  (fornite dal giroscopio) sono legate alle velocità angolari nel world frame  $\omega^w$  attraverso la seguente relazione [33]:

$$\begin{bmatrix} \omega_x^b \\ \omega_y^b \\ \omega_z^b \end{bmatrix} = R_x(\varphi)R_y(\theta) \begin{bmatrix} 0 \\ 0 \\ \omega_z^w \end{bmatrix} + R_x(\varphi) \begin{bmatrix} 0 \\ \omega_y^w \\ 0 \end{bmatrix} + \begin{bmatrix} \omega_x^w \\ 0 \\ 0 \end{bmatrix} = R_w^b(\Phi)\vec{\omega}^w. \quad (6.11)$$

Per calcolare le velocità angolari  $\omega^w$  basterà usare la trasposta della matrice  $R_w^b(\Phi)$  calcolata in precedenza:

$$\begin{aligned} \begin{bmatrix} \omega_x^w \\ \omega_y^w \\ \omega_z^w \end{bmatrix} &= R_b^w(\Phi)\vec{\omega}^b = (R_w^b(\Phi))^T \vec{\omega}^b \\ &= \begin{bmatrix} 1 & \sin \varphi \tan \theta & \cos \varphi \tan \theta \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi / \cos \theta & \cos \varphi / \cos \theta \end{bmatrix} \begin{bmatrix} \omega_x^b \\ \omega_y^b \\ \omega_z^b \end{bmatrix}. \end{aligned} \quad (6.12)$$

# Capitolo 7

## Sensor Fusion

Nelle applicazioni di navigazione, le tecniche di *dead reckoning* permettono di calcolare una stima della posizione corrente utilizzando solamente le informazioni relative alla posizione precedente e i dati di movimento come velocità, accelerazione e direzione del moto: questo comporta un inesorabile accumulo di errori. Per poter ridurre l'errore è possibile fare affidamento sui diversi punti di forza di diversi sensori, combinandoli attraverso tecniche di *sensor fusion*.

### 7.1 Introduzione alla Sensor Fusion per la localizzazione

Il termine *sensor fusion* identifica quell'insieme di tecniche atte alla combinazione dei flussi di informazioni emessi da un insieme di sensori, permettendo la riduzione dell'errore di stima che si otterrebbe utilizzando le diverse informazioni individualmente. Il processo di *sensor fusion* può essere facilmente implementato su un filtro di Kalman che, oltre a migliorare significativamente il processo di stima *online*, permette di ridurre gli errori introdotti dai rumori di rilevazione e dal bias.

Per quanto concerne il calcolo della posizione, un filtro di Kalman non è di alcuna utilità se l'unico sensore a disposizione è un accelerometro: per ottenere la posizione, l'accelerazione deve essere integrata due volte e, ad ogni passo di integrazione, si ottengono due costanti di valore non determi-

nabile: la velocità iniziale e la posizione iniziale. Le successive misurazioni dell'accelerometro non forniscono alcuna informazione aggiuntiva sullo stato iniziale. Inoltre, dato che le misurazioni sono rumorose, il processo di integrazione non fa che incrementare l'errore. Sono quindi necessari altri tipi di sensori che forniscano informazioni aggiuntive sulla posizione, come i sistemi GPS o le tecniche MMD viste in precedenza (vedi Capitolo 5).

La navigazione GPS/INS è un caso particolare di sensor fusion che permette di ridurre l'accumulo di errori combinando i vantaggi delle tecniche di posizionamento GPS e INS:

- l'uso delle informazioni dei sensori inerziali per il calcolo della posizione fornisce una risoluzione temporale dei dati più elevata rispetto a quello del singolo sistema GPS; inoltre, in caso di segnale GPS non disponibile, è possibile passare alla pura navigazione INS in attesa che la comunicazione con i satelliti GPS sia ripristinata;
- il segnale GPS permette di compensare il bias introdotto dall'uso di sensori inerziali.

L'orientamento, d'altro canto, non può essere ottenuto in modo preciso solo grazie alla bussola digitale o al giroscopio[33]:

- la combinazione tra accelerometro e magnetometro permette di calcolare valori affidabili in situazioni approssimativamente statiche, quindi su variazioni degli angoli relativamente lente. Inoltre, in situazioni dinamiche, il valore fornito dall'accelerometro racchiude, oltre al valore dell'accelerazione di gravità, anche una componente legata al moto rototraslazionale, quindi all'accelerazione lineare e all'accelerazione rotazionale, rendendo il calcolo dell'angolo estremamente impreciso;
- al contrario, il giroscopio fornisce rilevazioni molto precise nel breve periodo. Sebbene restituisca delle velocità angolari e, quindi, il calcolo dell'angolo richieda solamente un passo di integrazione, introduce comunque degli errori non trascurabili nel lungo periodo.

Per ottenere una stima più precisa dell'angolo si possono combinare i valori di velocità angolare forniti dal giroscopio con quelli di accelerometro e magnetometro: i primi permettono di ridurre l'errore derivante dal calcolo dell'orientamento attraverso la bussola digitale nelle fasi di transizione, mentre le rilevazioni della bussola impediscono la divergenza della stima causata dal bias del giroscopio.

## 7.2 Modello del sistema

Come anticipato, il GPS e la bussola digitale sono impiegati, rispettivamente, per ottenere i valori di posizione  $P_{GPS}$  e orientamento  $\Phi_{DC}$ ; quindi il *modello delle osservazioni* è dato da:

$$\begin{bmatrix} P_{GPS} \\ \Phi_{DC} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S^w \\ V^b \\ \Phi^w \end{bmatrix}. \quad (7.1)$$

mentre i valori restituiti dall'accelerometro  $acc^b = [acc_x^b, acc_y^b, acc_z^b]^T$  e dal giroscopio  $\omega = [\omega_x^b, \omega_y^b, \omega_z^b]^T$  sono utilizzati nelle equazioni di stato per correggere la risposta del sistema nelle situazioni dinamiche.

Per quanto riguarda il calcolo dell'orientamento, tenendo a mente l'Equazione 6.12, variazioni rapide degli angoli possono essere modellate come

$$\Phi_{k+1}^w = \Phi_k^w + R_b^w(\Phi) \vec{\omega}_k^b \Delta T \quad (7.2)$$

Le rilevazioni dell'accelerometro, invece, non possono essere utilizzate direttamente, dato che comprendono anche le componenti dell'accelerazione rototraslazionale del dispositivo e dell'accelerazione di gravità, ovvero

$$a\vec{c}^b = \vec{a}^b + \vec{g}^b + \vec{\omega}^b \times \vec{V}^b; \quad (7.3)$$

l'accelerazione lineare nel body frame è, quindi,

$$\vec{a}^b = a\vec{c}^b - \vec{g}^b - \vec{\omega}^b \times \vec{V}^b. \quad (7.4)$$

Di conseguenza, la velocità nel body frame, dalla cinematica, si ottiene sommando alla velocità attuale il prodotto tra l'accelerazione  $a\vec{c}^b$  e il tempo di campionamento  $\Delta T$

$$V_{k+1}^b = V_k^b + \Delta T (a\vec{c}^b - \vec{g}^b - \vec{\omega}^b \times \vec{V}^b)_k \quad (7.5)$$

Infine, è possibile ottenere la stima della posizione trasformando il vettore velocità dal body frame al world frame:

$$S_k^w + R_b^w(\Phi)(V_k^b \Delta T). \quad (7.6)$$

Dalle Equazioni 7.2,7.5 e 7.6 abbiamo così ottenuto il *modello dello stato* del sistema:

$$\begin{bmatrix} S^w \\ V^b \\ \Phi^w \end{bmatrix}_{k+1} = \begin{bmatrix} S_k^w + R_b^w(\Phi)(V_k^b \Delta T) \\ V_k^b + \Delta T(\text{acc}^b - \vec{g}^b - \vec{\omega}^b \times \vec{V}^b)_k \\ \Phi_k^w + R_b^w(\Phi)\omega_k^b \Delta T \end{bmatrix} \quad (7.7)$$

che in forma estesa risulta

$$\begin{bmatrix} S_x^w \\ S_y^w \\ S_z^w \\ V_x^b \\ V_y^b \\ V_z^b \\ \varphi^w \\ \theta^w \\ \psi^w \end{bmatrix}_{k+1} = \begin{bmatrix} S_x^w + \Delta T(\cos \theta \cos \psi V_x^b + (\sin \varphi \sin \theta \cos \psi - \cos \varphi \sin \psi) V_y^b \\ \quad + (\cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \psi) V_z^b) \\ S_y^w + \Delta T(\cos \theta \sin \psi V_x^b + (\sin \varphi \sin \theta \sin \psi + \cos \varphi \cos \psi) V_y^b \\ \quad + (\cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi) V_z^b) \\ S_z^w + \Delta T(-\sin \theta V_x^b + \sin \varphi \cos \theta V_y^b + \cos \varphi \cos \theta V_z^b) \\ V_x^b + \Delta T(\text{acc}_x^b - \omega_y^b V_z^b + \omega_z^b V_y^b + g \sin \theta) \\ V_y^b + \Delta T(\text{acc}_y^b - \omega_z^b V_x^b + \omega_x^b V_z^b - g \sin \varphi \cos \theta) \\ V_z^b + \Delta T(\text{acc}_z^b - \omega_x^b V_y^b + \omega_y^b V_x^b - g \cos \varphi \cos \theta) \\ \varphi^w + (\omega_x^b + (\sin \varphi \tan \theta) \omega_y^b + (\cos \varphi \tan \theta) \omega_z^b) \Delta T \\ \theta^w + (\cos \varphi \omega_y^b - \sin \varphi \omega_z^b) \Delta T \\ \psi^w + ((\sin \varphi / \cos \theta) \omega_y^b + (\cos \varphi / \cos \theta) \omega_z^b) \Delta T \end{bmatrix}_k \quad (7.8)$$

### 7.3 Osservazioni

Dato il modello del sistema non lineare, il capitolo successivo si concentra principalmente sull'analisi delle prestazioni dei due metodi di stima basati sul filtro di Kalman, ovvero l'EKF e l'UKF, per evidenziare quale dei due sia il più adatto al caso di studio preso in esame.

# Capitolo 8

## Configurazione dell'esperimento

### 8.1 Introduzione

Lo scopo dell'esperimento consiste nel determinare quale tra gli approcci basati su EKF e UKF sia il più adatto alla localizzazione su una implementazione GPS/INS, dal punto di vista della qualità delle soluzioni e della complessità computazionale. Come già visto nel Capitolo 4, l'EKF applica una linearizzazione analitica del sistema impiegando le matrici Jacobiane, mentre l'UKF, attraverso l'unscented transform, applica una linearizzazione statistica per "eliminare" la non-linearità.

La sperimentazione è composta da una fase di registrazione dei dati dell'IMU e del GPS su percorsi predefiniti e una successiva analisi offline attraverso il software MATLAB, per comparare le prestazioni dell'EKF e dell'UKF.

#### 8.1.1 Decomposizione di Cholesky

L'UKF, durante la formazione dei punti sigma, richiede il calcolo della radice quadrata della matrice di covarianza dell'errore di stato; uno dei metodi più diffusi e raccomandati per le applicazioni di GPS/INS, e utilizzato anche in questa tesi, è la *decomposizione di Cholesky* [22]: nel filtro di Kalman queste matrici sono sempre matrici definite positive, quindi la matrice  $P$  può essere decomposta come

$$P = LL^T \tag{8.1}$$

dove  $L$  può essere interpretata come la radice quadrata della matrice  $P$  ed è una matrice triangolare inferiore con valori reali positivi sulla diagonale, mentre  $L^T$  denota la sua trasposta. In MATLAB è possibile utilizzare direttamente la funzione `chol(P, 'lower')`.

## 8.2 Acquisizione dei dati

Il dispositivo scelto per la sperimentazione è un LG Nexus 4, con sistema operativo Android 5.1.1 Lollipop, sensori inerziali e ricevitore GPS.

### 8.2.1 Android API

Per poter accedere ai dati dei sensori inerziali del dispositivo è necessario utilizzare l'API Android, ad esempio:

```
private SensorManager manager;
private Sensor acc;
private Sensor gyro;
private Sensor mag;

/*...*/

manager = (SensorManager)
    getSystemService(Context.SENSOR_SERVICE);
acc = manager.getDefaultSensor(Sensor.TYPE_LINEAR_ACCELERATION);
gyro = manager.getDefaultSensor(Sensor.TYPE_GYROSCOPE);
mag = manager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
```

Alcuni di questi sensori potrebbero non essere hardware, ma implementati in parte via software per modificare il valore restituito dai componenti hardware:

- `TYPE_LINEAR_ACCELERATION`: restituisce l'accelerazione lineare lungo i tre assi di riferimento, escludendo l'accelerazione di gravità;
- `TYPE_GYROSCOPE`: restituisce la velocità angolare su ognuno degli assi di riferimento, includendo una compensazione del gyro-bias;

valori positivi identificano rotazioni in senso orario. In alcuni dispositivi, il giroscopio può essere implementato completamente via software grazie ai dati restituiti dagli altri sensori;

- TYPE\_MAGNETIC\_FIELD: restituisce l'intensità del campo magnetico lungo i tre assi di riferimento, tenendo in considerazione le distorsioni legate agli effetti magnetici causati dai vari componenti del dispositivo.

L'utilizzo di questi sensori "logici" implica dei ritardi e, nel caso del giroscopio, l'applicazione di forme di compensazione del bias comporta dei salti tra i vari valori rilevati. Il nostro progetto consiste nell'applicare tecniche di sensor fusion, le quali potrebbero restituire valori distorti se utilizzate su rilevazioni già parzialmente elaborate: detto ciò, riteniamo più affidabili i valori puri. Per poter utilizzare direttamente i valori dei sensori si possono utilizzare:

```
private SensorManager manager;
private Sensor unc_acc;
private Sensor unc_gyro;
private Sensor unc_mag;

/*....*/

manager = (SensorManager)
    getSystemService(Context.SENSOR_SERVICE);
unc_acc = manager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
unc_gyro =
    manager.getDefaultSensor(Sensor.TYPE_GYROSCOPE_UNCALIBRATED);
unc_mag =
    manager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD_UNCALIBRATED);
```

dove

- TYPE\_ACCELEROMETER: identifica un sensore che restituisce l'accelerazione lungo ciascun asse, tenendo conto anche della gravità;
- TYPE\_GYROSCOPE\_UNCALIBRATED: identifica un giroscopio che non effettua alcun tipo di *gyro-drift compensation*, anche se questi bias sono restituiti separatamente per poter essere utilizzati in calibrazioni custom (in SensorEvent.values);

- `TYPE_MAGNETIC_FIELD_UNCALIBRATED`: identifica un magnetometro che non effettua alcun tipo di calibrazione legata alle distorsioni causate dall'uso di magneti sul dispositivo, sebbene non escluda le calibrazioni di fabbrica.

Nel caso del nostro dispositivo, il giroscopio è stato implementato via software (l'opzione `GYROSCOPE_UNCALIBRATED` non risulta quindi disponibile), rendendo queste considerazioni irrilevanti.

Per quanto concerne la posizione GPS, innanzitutto è necessario scegliere da quale provider ricevere gli aggiornamenti della posizione, quindi dal `NETWORK_PROVIDER` o dal `GPS_PROVIDER`; successivamente, si richiede uno dei due permessi seguenti nel file manifest dell'applicazione:

- `ACCESS_COARSE_LOCATION`: permette l'accesso ai soli dati del `NETWORK_PROVIDER`;
- `ACCESS_FINE_LOCATION`: permette l'accesso ai dati di entrambi i provider, `NETWORK_PROVIDER` e `GPS_PROVIDER`.

```
private LocationManager locationManager;

/*...*/

locationManager = (LocationManager)
    this.getSystemService(Context.LOCATION_SERVICE);

LocationListener locListener = new LocationListener() {
    public void onLocationChanged(Location location) {}

    public void onStatusChanged(String provider, int status,
        Bundle extras) {}

    public void onProviderEnabled(String provider) {}

    public void onProviderDisabled(String provider) {}
};

locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER,
    0, 0, locationListener);
```

Lo sviluppo di applicazioni per Android 5.0 (API level 21) o superiori, richiede di dichiarare nel file manifest che l'app utilizzerà le funzionalità hardware `android.hardware.location.network` o `android.hardware.location.gps`, a seconda del provider utilizzato.

### 8.2.2 Allineamento dei sensori

Le equazioni esposte fino ad ora sono valide per sensori allineati secondo il sistema di riferimento del dispositivo, mentre, su diversi dispositivi, è facile che i sensori siano stati montati con diversi orientamenti:

- l'accelerazione di gravità permette di determinare facilmente la disposizione degli assi dell'accelerometro, visto che puntando l'ipotetico asse di riferimento del nostro dispositivo verso terra possiamo leggere su quale asse sarà indicata un'accelerazione di  $1g$  e con quale verso;
- per verificare la disposizione del giroscopio è sufficiente ruotare in senso antiorario il dispositivo lungo i tre ipotetici assi di riferimento, tenendo a mente che i valori dovrebbero corrispondere a velocità angolari con segno positivo;
- per quanto riguarda il magnetometro, è sufficiente tenere presente che la componente orizzontale del campo magnetico punta sempre al polo nord magnetico.

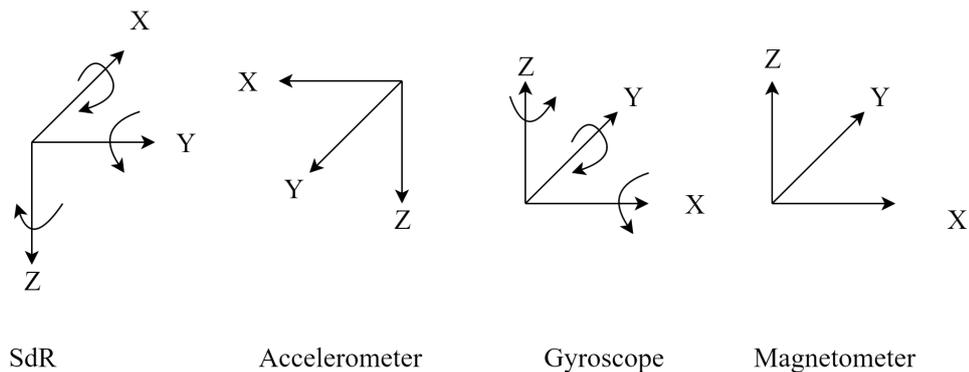


Figura 8.1: Sistemi di riferimento del dispositivo e dei sensori.

La Figura 8.1 riassume i risultati dell'allineamento. Denominando i valori restituiti dall'accelerometro, dal giroscopio e dal magnetometro rispettivamente con  $\alpha = [\alpha_x, \alpha_y, \alpha_z]^T$ ,  $\beta = [\beta_x, \beta_y, \beta_z]^T$  e  $\gamma = [\gamma_x, \gamma_y, \gamma_z]^T$ , i valori di  $acc$ ,  $\omega$  e  $mag$  da utilizzare sono:

$$acc = [-\alpha_y, -\alpha_x, \alpha_z]^T \quad (8.2a)$$

$$\omega = [\beta_y, \beta_x, -\beta_z]^T \quad (8.2b)$$

$$mag = [\gamma_y, \gamma_x, -\gamma_z]^T \quad (8.2c)$$

### 8.2.3 Conversione tra coordinate geografiche e cartesiane

Il sistema GPS fornisce la posizione corrente in coordinate geografiche; tuttavia, per poter utilizzare queste informazioni nelle nostre equazioni, sono necessarie le coordinate in forma cartesiana; assumendo che il raggio della terra sia mediamente  $R \sim 6371000m$ , e che le distanze percorse siano relativamente brevi rispetto ad un punto di riferimento denotato come  $refPoint = [refLat, refLon, refAlt]$ , possiamo assumere che:

$$x = \frac{\pi R}{180}(latitude - refLat) \quad (8.3a)$$

$$y = \frac{\pi R}{180} \cos(refLat \frac{\pi}{180})(longitude - refLon) \quad (8.3b)$$

$$z = -(altitude - refAlt) \quad (8.3c)$$

che risulta in una approssimazione molto accurata sulle brevi distanze.

### 8.2.4 Dataset

Il confronto tra l'EKF e l'UKF avviene su quattro dataset distinti:

- *Situazione statica*: questo dataset rappresenta una situazione in cui il dispositivo giace su un piano orizzontale, senza perturbazioni, ed è composto da 849 input;
- *Moto rettilineo*: il dispositivo, posizionato orizzontalmente al terreno, viene condotto da un punto ad un altro senza alcun tipo di rotazione. Il dataset è composto da 623 input;

- *Rotazioni elementari*: il dispositivo viene ruotato rispetto ai suoi assi di riferimento; più in dettaglio, viene effettuata una rotazione negativa seguita da una positiva per ogni asse. Durante le rotazioni vengono raccolte 277 rilevazioni.
- *Percorso a "zigzag"*: il dispositivo, sempre in posizione orizzontale, viene condotto lungo un percorso con diversi cambi di direzione, ottenendo un dataset di 734 input.

## 8.3 Analisi dei dati

### 8.3.1 Piattaforma

La piattaforma hardware utilizzata per l'analisi dei dati e la comparazione dei due approcci è un Notebook ASUS N551JX che dispone di:

- CPU: Processore Intel Core i7-4720HQ di quarta generazione, 4 core fisici e 8 virtuali, frequenza base di 2,60 GHz e memoria cache da 6 MB;
- RAM: 1 banco DDR3L da 1600 MHz SDRAM;
- Memoria: 1 HDD da 1 TB e velocità 5400 RPM.

L'implementazione degli algoritmi è avvenuta completamente su Matlab, quindi i risultati ottenuti sono da ritenersi relativi a questa piattaforma di test (vedi Sezione 9.2).

### 8.3.2 Parametri iniziali

Il comportamento finale del filtro di Kalman, come anticipato nella Sezione 4.2, è influenzato da un'appropriata scelta dei parametri  $x_0$ ,  $P_0$ ,  $R$  e  $Q$ . Assumendo di non conoscere con precisione lo stato iniziale, prendiamo  $x_0$  come

$$x_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T, \quad (8.4a)$$

mentre  $P_0$  viene impostato con valori relativamente alti

$$P_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.4b)$$

per permettere inizialmente alle rilevazioni di influenzare il filtro in modo adeguato.

Per quanto riguarda la matrice di covarianza  $Q$ , spesso si imposta come matrice diagonale

$$Q = q \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.4c)$$

dove  $q$  viene inizializzato con valori dell'ordine di  $10^{-9}$  sulla diagonale e, sperimentalmente, abbiamo osservato un buon comportamento per  $q = 10^{-8}$ . Infine la matrice  $R$ , che indica il grado di incertezza nelle rilevazioni, può essere impostata calcolando i valori delle diverse deviazioni standard delle variabile di rilevazione in situazioni statiche: sperimentalmente otteniamo

$$R = \begin{bmatrix} 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.002 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.002 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.04 \end{bmatrix}. \quad (8.4d)$$

# Capitolo 9

## Risultati sperimentali

Il confronto tra l'EKF e l'UKF avviene sui quattro dataset definiti nella Sottosezione 8.2.4. Di seguito presentiamo i risultati dell'applicazione delle tecniche di filtraggio EKF e UKF, in particolar modo i risultati ottenuti sulla stima dell'orientamento; successivamente compariamo i risultati ottenuti dai due approcci in termini di deviazione standard e complessità computazionale.

### 9.1 Risultati preliminari

La prima simulazione prevede l'applicazione delle tecniche di filtraggio in una situazione statica dove l'orientamento, quindi, si stabilizza su valori costanti. Le Figure 9.1 e 9.2 evidenziano chiaramente come, impiegando le tecniche di filtraggio e a seguito di un'opportuna regolazione dei parametri  $Q$  e  $R$ , sia possibile ottenere una stima dello stato migliore rispetto all'uso dei singoli dati forniti dalla bussola digitale. Da notare in particolar modo la stima dell'angolo di yaw: l'angolo  $\psi$  è ottenuto, oltre che dai valori dell'accelerometro, dalle rilevazioni del magnetometro che introducono una certa variabilità dei dati rispetto agli altri angoli; l'uso dei filtri riduce notevolmente questo errore.

La seconda simulazione prevede l'analisi delle tecniche di filtering mentre il dispositivo è in movimento, dove le rilevazioni della bussola digitale sono ancora più sensibili. Come già anticipato, quando il dispositivo si trova in uno stato di moto gli accelerometri forniscono dati che comprendono anche

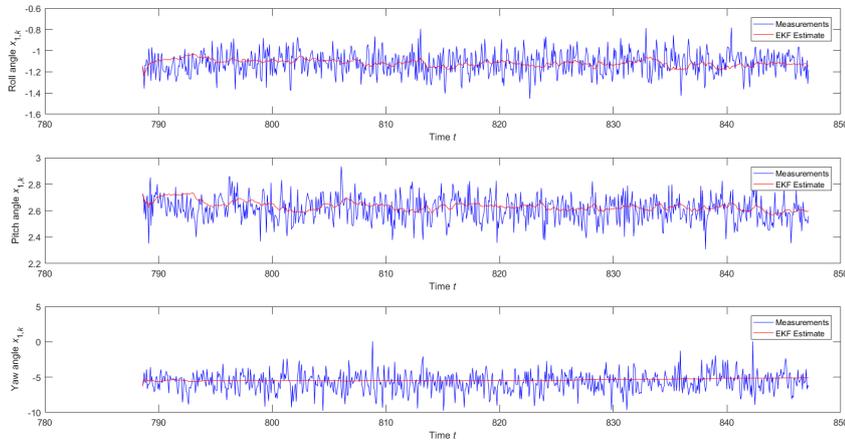


Figura 9.1: Grafico della stima dell'orientamento ottenuto grazie all'EKF in una situazione statica.

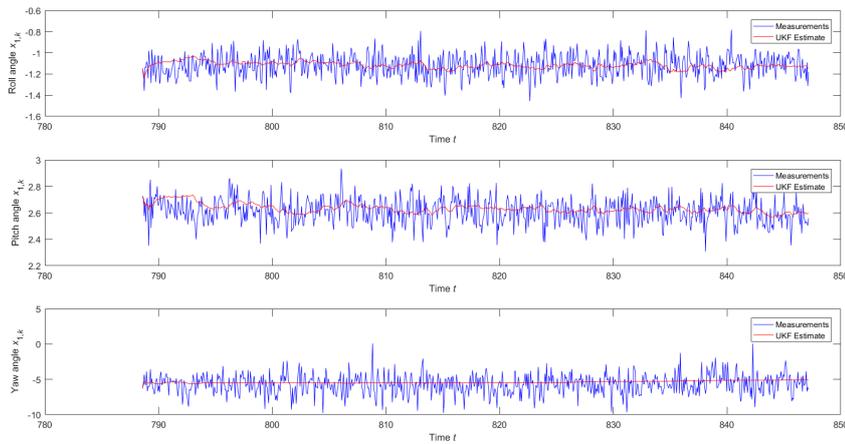


Figura 9.2: Grafico della stima dell'orientamento ottenuto grazie all'UKF in una situazione statica.

una componente relativa al movimento, oltre che all'accelerazione di gravità; questo fa sì che la bussola digitale sia da ritenere inaffidabile. Dalle Figure

9.3 e 9.4 si evince che, grazie alle tecniche di sensor fusion, con il contributo del giroscopio, questa inaccuratezza sia quasi completamente eliminata.

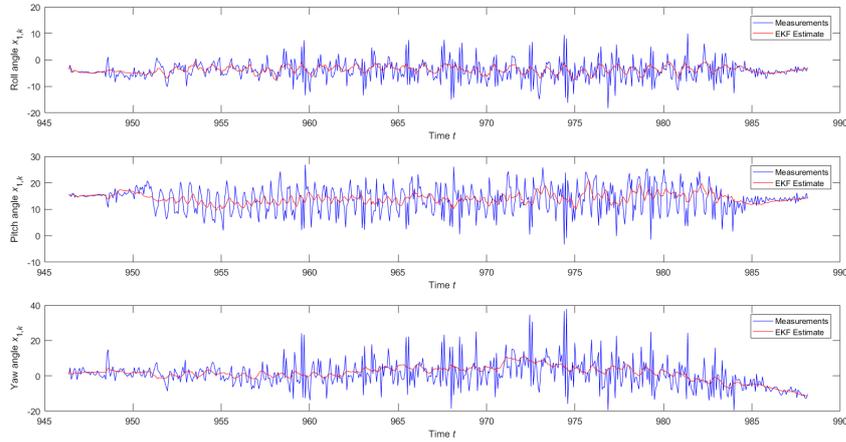


Figura 9.3: Grafico della stima dell'orientamento ottenuto grazie all'EKF mentre il dispositivo segue un movimento rettilineo.

Per una visione più accurata dei grafici relativi ai diversi dataset, invitiamo a consultare l'Appendice A.

## 9.2 Comparazione tra EKF e UKF

Dopo aver verificato la validità dell'approccio a sensor fusion, andiamo a stabilire quale dei due filtri sia il più adatto per essere impiegato per ridurre l'errore di stima dello stato in applicazioni context-aware: l'obiettivo consiste nel massimizzare la precisione della stima minimizzando la complessità computazionale richiesta.

### 9.2.1 Deviazione standard

Come indicatore della precisione prendiamo in considerazione la deviazione standard mostrata dalla serie di dati in cui i valori degli angoli dovrebbero rimanere costanti: quindi la situazione statica e il moto rettilineo.

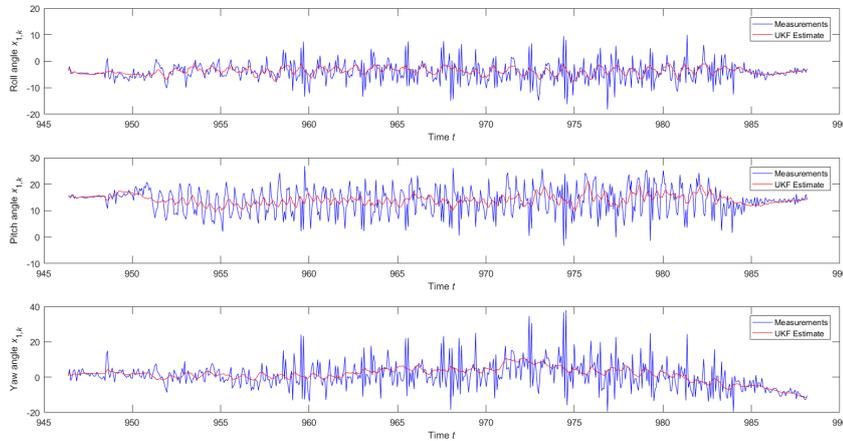


Figura 9.4: Grafico della stima dell'orientamento ottenuto grazie all'UKF mentre il dispositivo segue un movimento rettilineo.

La Tabella 9.1 riassume i risultati delle deviazioni standard dei tre angoli in una situazione statica, dove risulta ancora più evidente il miglioramento nella stima introdotto dal processo di filtering:

- $\sigma_\varphi$  la dispersione ottenuta per l'angolo di roll equivale a circa il 29% dell'originale;
- $\sigma_\theta$ : le stime dell'angolo di pitch stimate dai filtri mostrano una dispersione di circa il 35% dei dati puri forniti dalla bussola digitale;
- $\sigma_\psi$ : la dispersione delle stime degli angoli di yaw è ridotta al 11%.

Tuttavia, sebbene l'EKF mostri una deviazione standard leggermente inferiore all'UKF, i due valori possono essere considerati uguali.

La Tabella 9.2 mostra invece i risultati nel caso di un movimento rettilineo: anche in questo caso non si distingue un approccio migliore dell'altro in termini di precisione:

- $\sigma_\varphi$  la dispersione ottenuta per l'angolo di roll equivale a circa il 37% dell'originale;

<i>Sorgente dati</i>	$\sigma_\varphi(rad)$	$\sigma_\theta(rad)$	$\sigma_\psi(rad)$
Dati non filtrati	0.0018020147	0.0016201730	0.0234501675
EKF	0.0005235200	0.0005725785	0.0025157856
UKF	0.0005234693	0.0005725767	0.0026351684

Tabella 9.1: Le deviazioni standard degli angoli di roll, pitch e yaw ottenute sui dati non filtrati e filtrati con l'EKF e l'UKF in una situazione statica.

- $\sigma_\theta$ : le stime dell'angolo di pitch stimate dai filtri mostrano una dispersione di circa il 36% dei dati puri forniti dalla bussola digitale;
- $\sigma_\psi$ : la dispersione delle stime degli angoli di yaw è ridotta al 53%.

<i>Sorgente dati</i>	$\sigma_\varphi(rad)$	$\sigma_\theta(rad)$	$\sigma_\psi(rad)$
Dati non filtrati	0.0636888825	0.0871997100	0.1347495117
EKF	0.0233196492	0.0313930496	0.0718874522
UKF	0.0233194734	0.0313930201	0.0719703589

Tabella 9.2: Le deviazioni standard degli angoli di roll, pitch e yaw ottenute sui dati non filtrati e filtrati con l'EKF e l'UKF durante un movimento rettilineo.

In termini di riduzione dell'errore di stima, sia l'EKF che l'UKF forniscono risultati più che soddisfacenti, ma pressochè identici.

### 9.2.2 Complessità computazionale

Un buon indicatore della complessità computazionale dei due filtri è il tempo totale di stima, quindi il tempo necessario al filtro per calcolare la stima utilizzando tutti i dati di ciascun dataset, senza considerare la fase di inizializzazione.

La Tabella 9.3 mostra il tempo necessario impiegato per portare a termine la simulazione su ogni dataset. La tabella 9.4, invece, illustra il tempo necessario a ciascun algoritmo per compiere una singola iterazione.

<i>Dataset</i>	<i>EKF Time (s)</i>	<i>UKF Time (s)</i>
Situazione statica (849 iterazioni)	0.068037	0.333995
Moto rettilineo (623 iterazioni)	0.063376	0.260835
Rotazioni elementari (277 iterazioni)	0.045170	0.143752
Percorso a “zigzag” (734 iterazioni)	0.066479	0.299805

Tabella 9.3: Tempo di computazione richiesto dall'EKF e dall'UKF, da fermo.

	<i>EKF Time (<math>\mu s</math>)</i>	<i>UKF Time (<math>\mu s</math>)</i>
Tempo medio di esecuzione	92.362235	414.503334

Tabella 9.4: Tempo di computazione richiesto dall'EKF e dall'UKF per completare una singola iterazione del processo di stima.

In questo caso la differenza risulta più marcata: l'UKF ha una complessità computazionale di quasi 4.5 volte superiore all'EKF.

### 9.3 Outline

Le tecniche di sensor fusion permettono una riduzione notevole dell'errore di stima dello stato attuale di un sistema. In termini di prestazioni e tenendo a mente i nostri requisiti, l'EKF si è dimostrato la soluzione più adatta.

# Capitolo 10

## Conclusioni

Le applicazioni context-aware necessitano di una stima precisa della posizione e dell'orientamento del dispositivo nei contesti indoor; per raggiungere tale obiettivo, viene proposto un approccio basato su navigazione GPS/INS a cui vengono applicate tecniche di sensor fusion per ridurre l'errore di stima. In questa tesi vengono analizzate e comparate due tecniche utilizzate come base per l'approccio a sensor fusion, ovvero l'Extended Kalman Filter e l'Unscented Kalman Filter delle quali l'EKF si è dimostrato indubbiamente più adatto per le applicazioni di localizzazione, data la complessità computazionale inferiore rispetto all'UKF.

È importante sottolineare che la teoria alla base di questa tesi non è da considerarsi applicabile solamente ai sistemi GPS, ma è direttamente estendibile ad altre tecniche di localizzazione: una tecnica che sta dando risultati promettenti è l'impiego dell'*Ultra-Wide Band* per la localizzazione, dove l'UWB consiste in un protocollo di comunicazione wireless che permette un uso efficiente della banda grazie a un'alta frequenza e una bassa potenza di trasmissione.

Il passo successivo consiste nella creazione di un framework che permetta di ottenere le informazioni sulla posizione e l'orientamento del dispositivo, una condizione fondamentale ed essenziale per la diffusione dei dispositivi e delle applicazioni di realtà aumentata.



# Appendice A

## Grafici delle simulazioni

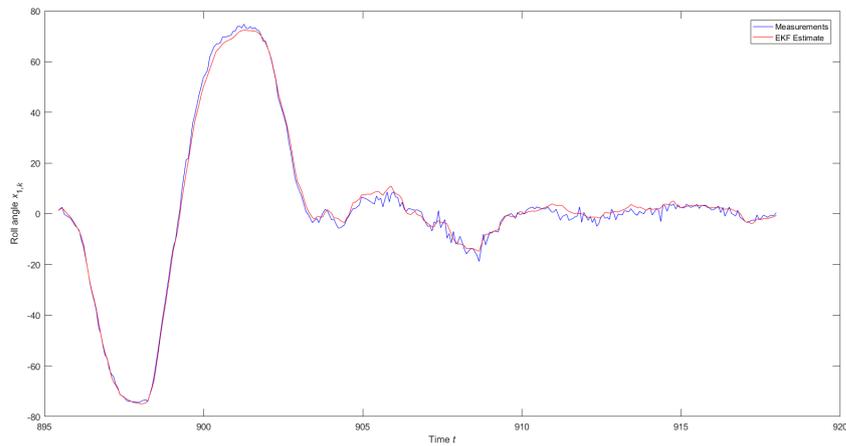


Figura A.1: Filtro EKF, situazione statica, stima dell'angolo di roll.

APPENDICE A. GRAFICI DELLE SIMULAZIONI

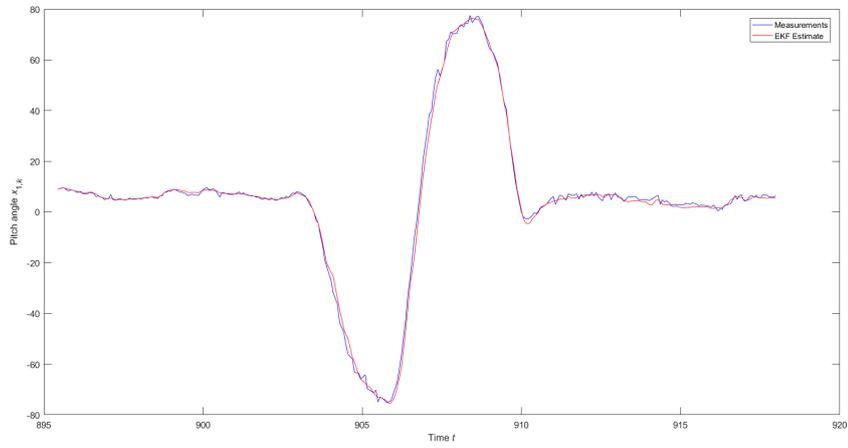


Figura A.2: Filtro EKF, situazione statica, stima dell'angolo di pitch.

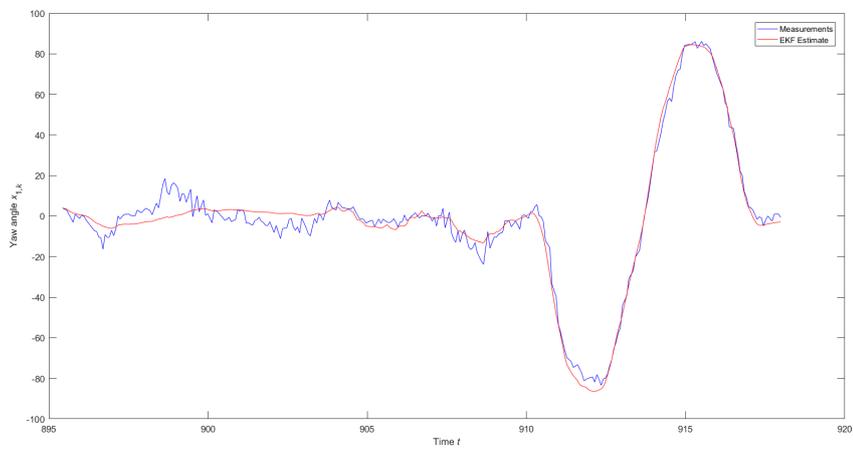


Figura A.3: Filtro EKF, situazione statica, stima dell'angolo di yaw.

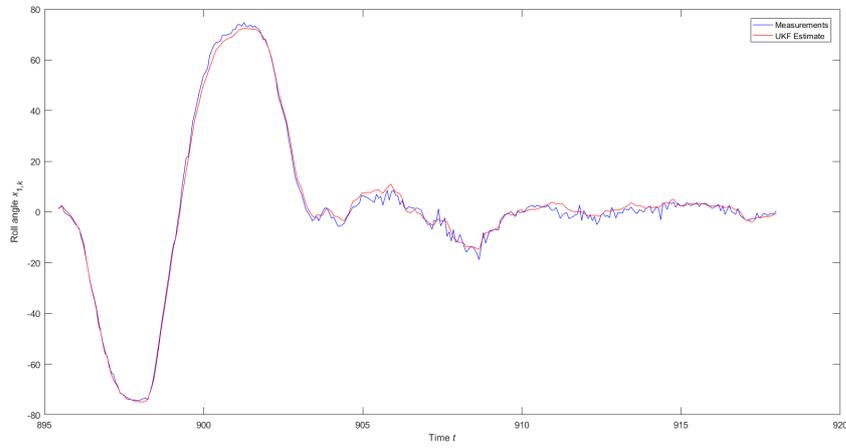


Figura A.4: Filtro UKF, situazione statica, stima dell'angolo di roll.

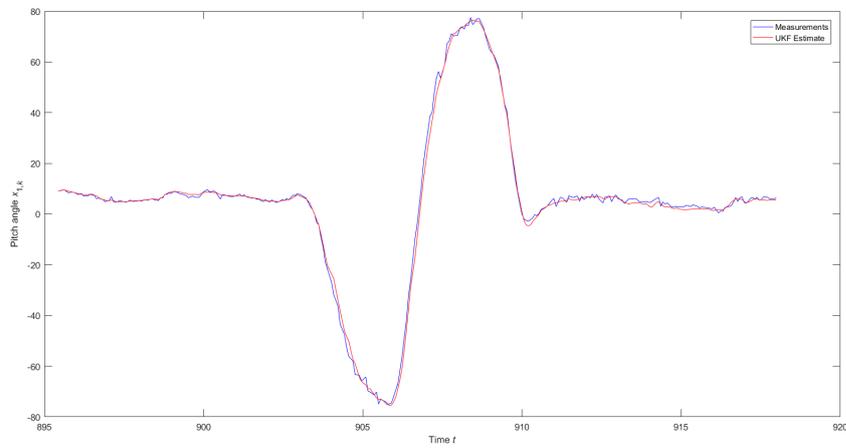


Figura A.5: Filtro UKF, situazione statica, stima dell'angolo di pitch.

APPENDICE A. GRAFICI DELLE SIMULAZIONI

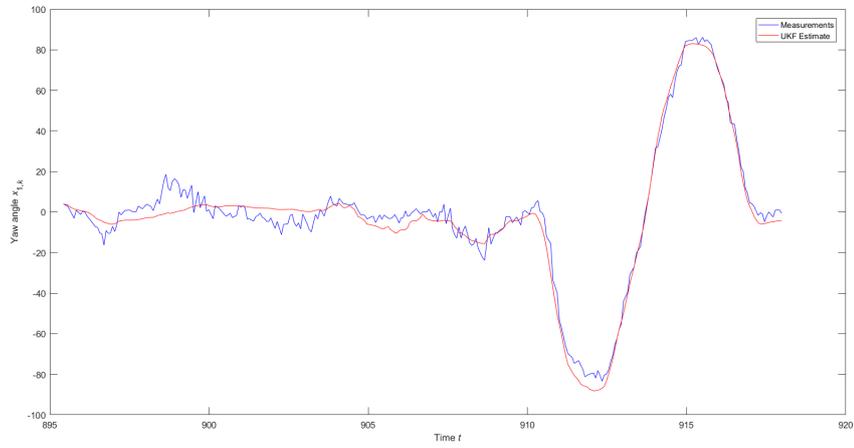


Figura A.6: Filtro UKF, situazione statica, stima dell'angolo di yaw.

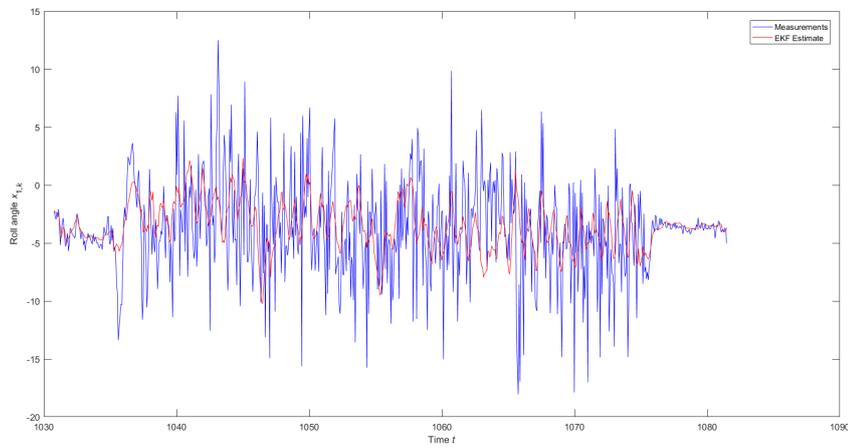


Figura A.7: Filtro EKF, percorso a zigzag, stima dell'angolo di roll.

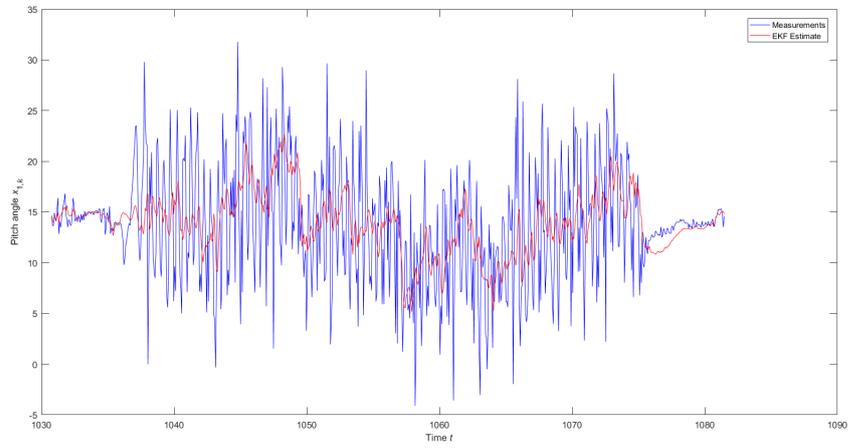


Figura A.8: Filtro EKF, percorso a zigzag, stima dell'angolo di pitch.

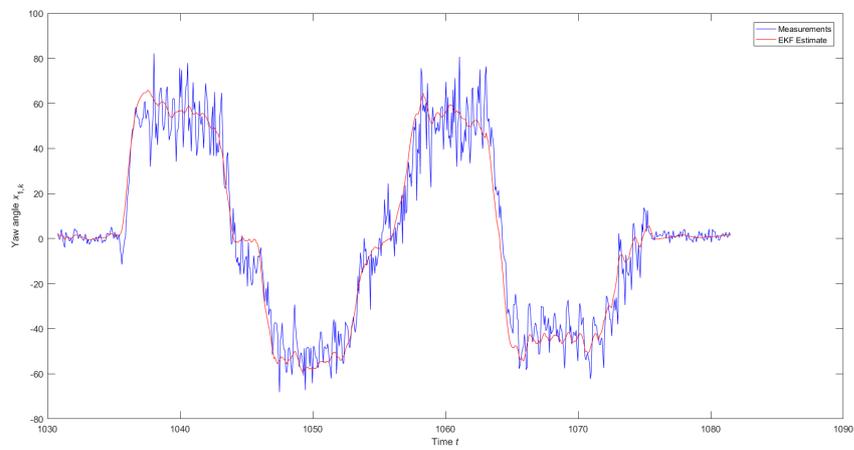


Figura A.9: Filtro EKF, percorso a zigzag, stima dell'angolo di yaw.

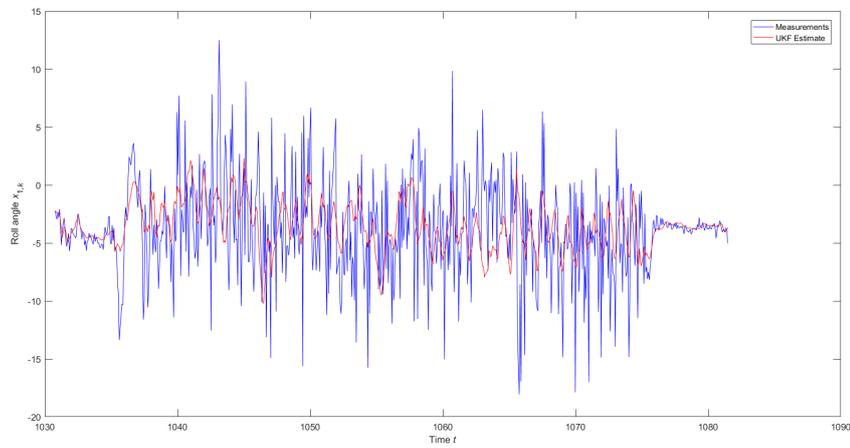


Figura A.10: Filtro UKF, percorso a zigzag, stima dell'angolo di roll.

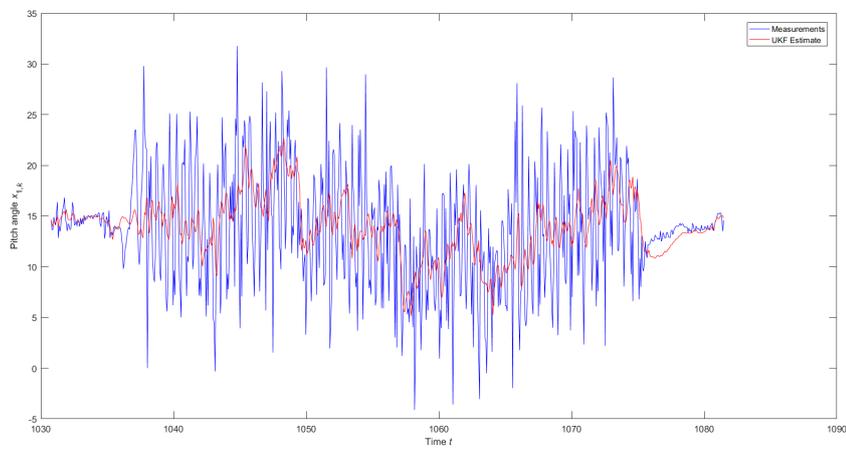


Figura A.11: Filtro UKF, percorso a zigzag, stima dell'angolo di pitch.

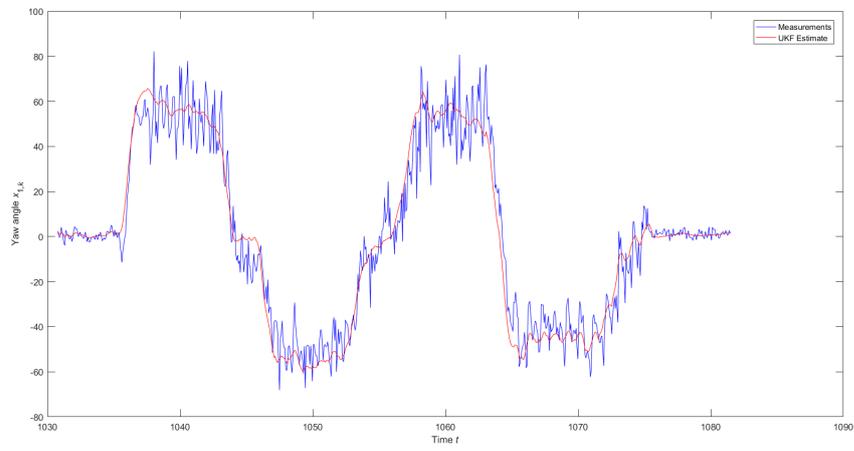


Figura A.12: Filtro UKF, percorso a zigzag, stima dell'angolo di yaw.



# Bibliografia

- [1] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and virtual environments*, 6(4):355–385, 1997.
- [2] W. Barfield. *Fundamentals of wearable computers and augmented reality, Second Edition*. CRC Press, 2016.
- [3] J.-L. Blanco. A tutorial on se (3) transformation parameterizations and on-manifold optimization. *University of Malaga, Tech. Rep*, 2010.
- [4] J. Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 58(15-16):1–35, 2006.
- [5] V. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. *IEEE pervasive computing*, 2(3):24–33, 2003.
- [6] A. Goldsmith. *Wireless communications*. Cambridge university press, 2005.
- [7] G. Hein, M. Paonni, V. Kropp, and A. Teuber. Gnss indoor - fighting the fading, part 1. 2008.
- [8] G. Hein, A. Teuber, H.-J. Thierfelder, and A. Wolfe. Gnss indoor - fighting the fading, part 2. 2008.
- [9] B. Kim and S.-H. Kong. Indoor positioning based on bayesian filter using magnetometer measurement difference. In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pages 1–5. IEEE, 2015.

- 
- [10] G. Kortuem, Z. Segall, and M. Bauer. Context-aware, adaptive wearable computers as remote interfaces to intelligent environments. In *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*, pages 58–65. IEEE, 1998.
- [11] J. Lategahn, M. Müller, and C. Röhrig. Robust pedestrian localization in indoor environments with an imu aided tdoa system. In *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on*, pages 465–472. IEEE, 2014.
- [12] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 421–430. ACM, 2012.
- [13] T. Magnusson. State estimation of uav using extended kalman filter. 2013.
- [14] R. Mautz. Indoor positioning technologies, 2012.
- [15] P. Milgram and H. Colquhoun. A taxonomy of real and virtual world display integration. *Mixed reality: Merging real and virtual worlds*, 1:1–26, 1999.
- [16] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Photonics for industrial applications*, pages 282–292. International Society for Optics and Photonics, 1995.
- [17] N. Naik, R. Gemson, M. Ananthasayanam, et al. Introduction to the kalman filter and tuning its statistics for near optimal estimates and cramer rao bound. *arXiv preprint arXiv:1503.04313*, 2015.
- [18] A. Noth, S. Fux, and S. Bouabdallah. Inertial measurement unit.
- [19] T. Ozyagcilar. Implementing a tilt-compensated ecompass using accelerometer and magnetometer sensors. *Freescale semiconductor, AN*, 4248, 2012.
- [20] M. Pedley. Tilt sensing using a three-axis accelerometer. *Freescale semiconductor application note*, pages 1–22, 2013.

## BIBLIOGRAFIA

---

- [21] B. J. Rhodes. *Just-in-time information retrieval*. PhD thesis, Massachusetts Institute of Technology, 2000.
- [22] M. Rhudy, Y. Gu, J. Gross, and M. R. Napolitano. Evaluation of matrix square root operations for ukf within a uav gps/ins sensor fusion application. *International Journal of Navigation and Observation*, 2011, 2012.
- [23] S. Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.
- [24] T. Starner. The challenges of wearable computing: Part 1. *Ieee Micro*, 21(4):54–67, 2001.
- [25] T. Starner. The challenges of wearable computing: Part 2. *Ieee Micro*, 21(4):54–67, 2001.
- [26] T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, R. W. Picard, and A. Pentland. Augmented reality through wearable computing. *Presence: Teleoperators and Virtual Environments*, 6(4):386–398, 1997.
- [27] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- [28] M. van Biezen. Special topics - the kalman filter.
- [29] E. A. Wan and R. Van Der Merwe. The unscented kalman filter for non-linear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. Ieee, 2000.
- [30] G. F. Welch. Kalman filter. In *Computer vision*, pages 435–437. Springer, 2014.
- [31] X. Yun and E. R. Bachmann. Design, implementation, and experimental results of a quaternion-based kalman filter for human body motion tracking. *IEEE Transactions on Robotics*, 22(6):1216–1227, Dec 2006.
- [32] R. Zhang, F. Hoffinger, and L. Reindl. Inertial sensor based indoor localization and monitoring system for emergency responders. *IEEE Sensors Journal*, 13(2):838–848, 2013.

- [33] H. Zhao and Z. Wang. Motion measurement using inertial sensors, ultrasonic sensors, and magnetometers with extended kalman filter for data fusion. *IEEE Sensors Journal*, 12(5):943–953, 2012.