

Oltre i limiti del
singolo sensore:
la rivoluzione della
SENSOR FUSION



Leonardo Chieco

Oltre i limiti del singolo sensore: la rivoluzione della Sensor Fusion

Negli ultimi anni la sensor fusion è diventata una delle tecnologie chiave per realizzare sistemi elettronici intelligenti, capaci di percepire, interpretare e reagire al mondo che li circonda. Dietro a concetti oggi familiari come la stabilizzazione di un drone, il tracciamento dei movimenti in uno smartphone o la precisione di un gimbal per videocamere, si nasconde infatti un complesso lavoro di integrazione tra sensori diversi. Nessun sensore, preso singolarmente, è in grado di fornire una misura perfetta: ognuno porta con sé limiti intrinseci, errori sistematici, rumore e derive nel tempo. Ed è proprio per superare queste limitazioni che nasce l'idea di fondere i dati provenienti da più sorgenti, sfruttando i punti di forza di ciascun dispositivo per compensarne le debolezze.

Per comprendere appieno le tecniche di sensor fusion, partiremo dalle basi, classificando i sensori disponibili sul mercato, così da capire il ruolo che ciascuno di essi può avere all'interno di un sistema complesso. Successivamente esamineremo le tecniche più diffuse di sensor fusion per poi concentrarci su due protagonisti assoluti nel campo della misura del movimento e dell'orientamento: accelerometri e giroscopi MEMS. Vedremo come è possibile fondere queste due fonti di informazioni attraverso il filtro di Kalman.

Buona lettura!

L'autore

Leonardo Chieco è un ingegnere elettronico con oltre 20 anni di esperienza nella progettazione e sviluppo di software per il controllo dell'automazione (PC/PLC), nella progettazione di schede elettroniche per applicazioni industriali, firmware, robotica e meccatronica.

LinkedIn: <https://www.linkedin.com/in/leonardo-chieco-53550b129/>

1. Introduzione

Negli ultimi decenni, l'elettronica e l'informatica hanno vissuto una crescita esponenziale, portando allo sviluppo di sistemi sempre più intelligenti e capaci di interagire con l'ambiente in modo complesso. Una delle chiavi di questa evoluzione è stata la capacità di raccogliere informazioni tramite sensori: accelerometri, giroscopi, magnetometri, sensori di pressione, di temperatura, di posizione, di velocità e molti altri. Tuttavia, un singolo sensore difficilmente è in grado di fornire un'informazione completa, accurata e priva di errori. È proprio da questa limitazione che nasce il concetto di sensor fusion, ovvero l'arte e la scienza di combinare i dati provenienti da più sensori per ottenere una stima più affidabile, robusta e utile della grandezza di interesse.

Storicamente, i primi sistemi elettronici basati su sensori facevano affidamento su una sola misura. Ad esempio, per determinare la velocità di un veicolo si utilizzava un tachimetro meccanico o un encoder; per stimare l'orientamento, un giroscopio; per misurare la distanza, un sensore a ultrasuoni. Questi approcci "monosensore" funzionavano, ma presentavano limiti evidenti:

- Errori intrinseci del sensore: rumore, offset, deriva.
- Sensibilità al contesto: un accelerometro può confondere un'accelerazione lineare con la gravità.
- Mancanza di ridondanza: se il sensore si guasta o fornisce dati distorti, il sistema non ha alternative.

Negli anni '60, con lo sviluppo dei sistemi di navigazione inerziale per applicazioni aerospaziali, si iniziò a comprendere l'importanza di combinare più fonti di informazione. In quegli anni nacque anche uno degli algoritmi più celebri e ancora oggi utilizzati: il filtro di Kalman, sviluppato da Rudolf Kalman nel 1960. Questo strumento matematico ha permesso di fondere dati provenienti da giroscopi e accelerometri per stimare la posizione e l'orientamento degli aerei e delle navicelle spaziali in maniera molto più stabile rispetto a quanto fosse possibile con i singoli sensori. Non a caso, la NASA lo adottò per le missioni Apollo.

Da allora, il concetto di sensor fusion si è diffuso in moltissimi campi, trovando nuove implementazioni con l'aumentare della potenza di calcolo dei microprocessori e con la miniaturizzazione dei sensori MEMS.

I principali benefici dell'uso della sensor fusion possono essere riassunti in alcuni punti fondamentali:

- **Maggiore accuratezza:** Combinando sensori che hanno punti di forza e debolezza diversi, si ottiene una misura più precisa. Ad esempio, un giroscopio è molto sensibile ai movimenti rapidi ma soffre di deriva, mentre un accelerometro è stabile sul lungo periodo ma sensibile al rumore: insieme forniscono una stima affidabile dell'orientamento.
- **Robustezza e affidabilità:** In presenza di disturbi o guasti di un sensore, gli altri possono compensare. Questo è fondamentale in applicazioni critiche come l'automotive o l'aerospazio.
- **Riduzione del rumore:** Algoritmi di fusione ben progettati (Kalman, Complementary Filter, Particle Filter, ecc.) permettono di filtrare il rumore statistico e ottenere segnali puliti e utilizzabili.
- **Maggiore ricchezza informativa:** Spesso la combinazione di più sensori non solo migliora l'accuratezza, ma rende possibile stimare grandezze che altrimenti non sarebbero misurabili.

Oggi la sensor fusion è praticamente ovunque, anche se spesso non ce ne rendiamo conto. Alcuni esempi emblematici:

- **Smartphone e wearable:** Nei telefoni moderni, la sensor fusion combina accelerometro, giroscopio e magnetometro per capire l'orientamento del dispositivo, rilevare i passi, stabilizzare la bussola digitale e migliorare la navigazione con GPS.
- **Droni e robotica mobile:** I droni utilizzano sensor fusion per mantenere la stabilità in volo: le misure di accelerometro e giroscopio vengono fuse per calcolare roll, pitch e yaw, mentre il GPS fornisce la posizione assoluta. Nei robot mobili, lidar e telecamere si combinano con l'odometria per navigare in ambienti complessi.
- **Automotive:** Le auto moderne impiegano la sensor fusion in sistemi ADAS (Advanced Driver Assistance Systems): radar, lidar, telecamere e sensori

ultrasonici lavorano insieme per rilevare ostacoli, pedoni e veicoli vicini. Anche l'ABS e l'ESP si basano su fusione di sensori inerziali e di velocità delle ruote.

- Applicazioni mediche: In ambito biomedicale, la sensor fusion consente di combinare dati di movimento e fisiologici per il monitoraggio dei pazienti, come avviene in dispositivi indossabili che seguono il battito cardiaco, la postura e l'attività fisica.
- Industria e automazione: Nelle linee di produzione, diversi sensori collaborano per monitorare vibrazioni, posizione e qualità dei processi, abilitando manutenzione predittiva e controllo di precisione.

2. Cosa sono i sensori

I sensori rappresentano gli occhi e le orecchie di qualunque sistema elettronico o informatico che interagisca con il mondo reale, trasformando fenomeni fisici in informazioni utilizzabili. Si possono considerare come una sorta di “organi di senso” delle macchine: così come gli occhi e le orecchie forniscono al cervello umano le informazioni sull’ambiente circostante, i sensori permettono a un sistema elettronico di acquisire dati sul mondo fisico e trasformarli in segnali interpretabili da un computer, un microcontrollore o un sistema di controllo.

Quando parliamo di **sensori**, quindi, ci riferiamo a dispositivi progettati per rilevare una grandezza fisica, chimica o biologica e convertirla in un segnale interpretabile da un sistema elettronico. Senza di essi, automobili, smartphone, robot industriali, droni, impianti medici e persino i sistemi domotici più semplici non potrebbero interagire con l’ambiente circostante. In altre parole, i sensori permettono alle macchine di percepire ciò che accade nel mondo reale e di reagire di conseguenza.

Il mercato dei sensori è oggi vastissimo: si spazia da dispositivi molto semplici, come un **sensore di temperatura basato su una resistenza termica**, a componenti estremamente sofisticati come un **LiDAR tridimensionale** usato per la guida autonoma. Per comprendere meglio questo universo, possiamo iniziare suddividendo i sensori secondo diversi criteri: la **natura della grandezza misurata**, la **tecnologia costruttiva** e il **tipo di segnale in uscita**.

I sensori si possono classificare in molti modi, ma una delle distinzioni più immediate è quella tra analogici e digitali.

Un **sensore analogico** fornisce un segnale continuo, cioè proporzionale in maniera lineare (o comunque nota) alla grandezza fisica misurata.

Esempi tipici:

- Un termistore NTC che varia la resistenza in base alla temperatura.
- Un potenziometro che fornisce una tensione proporzionale alla posizione della sua leva.
- Un microfono a condensatore che produce una tensione variabile in funzione delle onde sonore ricevute.

Un **sensore digitale** invece include già un sistema elettronico interno che digitalizza la misura e la comunica sotto forma di pacchetti di dati o segnali logici. Esempi tipici:

- Un sensore di temperatura digitale come il DS18B20, che comunica direttamente tramite protocollo 1-Wire.
- Un accelerometro digitale come l'MPU6050, che trasmette le misure via I²C o SPI.
- Un sensore di distanza a ultrasuoni come l'HC-SR04, che restituisce impulsi digitali misurabili in microsecondi.

I sensori digitali hanno il vantaggio di fornire dati più robusti al rumore e facilmente interpretabili via software, ma possono introdurre latenza o limitazioni di risoluzione dovute all'ADC integrato.

Nei paragrafi che seguono passeremo in rassegna le principali tipologie di sensori oggi reperibili sul mercato, strumenti imprescindibili per makers e ingegneri che desiderano progettare macchine sempre più complesse, intelligenti e capaci di interagire con l'ambiente circostante. L'elenco non pretende di essere esaustivo, ma ha l'obiettivo di offrire una panoramica chiara delle soluzioni disponibili, fornendo spunti utili per orientarsi nella scelta e per individuare con maggiore consapevolezza il sensore più adatto alle proprie necessità, anche attraverso la consultazione dei cataloghi dei produttori.

Sensori di posizione e spostamento

Una grande categoria di sensori riguarda la misura di **posizione lineare o angolare**.

- **Potentiometri**: un cursore mobile scorre su una resistenza variabile, traducendo lo spostamento meccanico in variazione di tensione. Ancora usati in joystick e pedali elettronici.
- **Encoder ottici**: rilevano la rotazione di un albero grazie a un disco forato e a un sistema LED/fotodiode. Possono essere **incrementali** (misurano variazioni) o **assoluti** (forniscono direttamente la posizione angolare).
- **Sensori induttivi**: sfruttano variazioni di campo elettromagnetico per rilevare spostamenti di oggetti metallici.

- **Sensori capacitivi di spostamento:** misurano variazioni di capacità dovute a piccoli movimenti.
- **LVDT (Linear Variable Differential Transformer):** trasduttori elettromagnetici di alta precisione per spostamenti lineari.

Questi sensori sono fondamentali in robotica, CNC, automazione industriale, sistemi automotive (es. posizione farfalla acceleratore).

Sensori di movimento e inerziali

I **sensori inerziali** sono oggi tra i più diffusi grazie alla miniaturizzazione MEMS.

- **Accelerometri:** misurano l'accelerazione lungo uno o più assi. Possono rilevare vibrazioni, urti, inclinazioni.
- **Giroscopi:** misurano la velocità angolare (gradi al secondo). Essenziali in stabilizzatori, droni, sistemi di navigazione.
- **IMU (Inertial Measurement Unit):** combinano accelerometri, giroscopi e spesso magnetometri, offrendo una stima completa di orientamento e movimento.
- **Magnetometri:** rilevano il campo magnetico terrestre, usati come bussola elettronica.

Questi sensori sono la base di smartphone, wearable, gimbal, videogiochi con motion control.

Sensori di forza, pressione e coppia

Un'altra famiglia riguarda la misura di **grandezze meccaniche**.

- **Celle di carico (strain gauge):** misurano deformazioni elastiche in un materiale e quindi la forza applicata. Usate in bilance e macchine di prova.
- **Sensori piezoelettrici:** convertono pressioni o vibrazioni in segnali elettrici.
- **Sensori di pressione:** possono essere piezoresistivi, capacitivi o MEMS; servono per misurare pressione di fluidi o aria (es. sensore MAP nelle auto).
- **Sensori di coppia:** spesso basati su strain gauge montati su alberi rotanti.

Sensori di temperatura

Probabilmente i più diffusi in assoluto. Le principali tecnologie sono:

- **Termistori (NTC/PTC)**: resistenze che cambiano valore con la temperatura.
- **RTD (Resistance Temperature Detector)**: resistenze metalliche, molto precise (platino PT100).
- **Termocoppie**: generano una tensione proporzionale alla differenza di temperatura tra due giunzioni. Coprono range molto ampi (anche >1000 °C).
- **Sensori digitali**: integrati con ADC, compensazione e interfaccia (es. DS18B20, TMP102).

Sensori ottici e di luce

Questa categoria comprende sia dispositivi semplici, sia complessi.

- **Fotodiоди e fototransistor**: convertono luce in corrente.
- **LDR (Light Dependent Resistor)**: resistenza che varia con la luminosità.
- **Sensori di colore**: distinguono componenti RGB.
- **Sensori di distanza ottici**: a tempo di volo (ToF), triangolazione o modulazione di fase.
- **LiDAR**: sistemi avanzati per mappatura 3D basati su laser.
- **Telecamere CMOS/CCD**: sensori di immagine, usati in fotocamere, smartphone e visione artificiale.

Sensori magnetici

- **Magnetometri**: misurano intensità e direzione del campo magnetico.
- **Sensori ad effetto Hall**: rilevano la presenza di un campo magnetico e sono usati in tachimetri, encoder magnetici, sensori di prossimità.

Sensori di gas, chimici e ambientali

Negli ultimi anni, grazie all'IoT, sono cresciuti i sensori ambientali.

- **Sensori di gas (MQ-x)**: economici, misurano concentrazioni di gas come CO, metano, alcol.
- **Sensori elettrochimici**: molto precisi, usati in ambito medico (ossigeno, CO₂).
- **Sensori di umidità**: capacitivi o resistivi.
- **Sensori di qualità dell'aria**: combinano misure di VOC (composti organici volatili), polveri sottili, CO₂ equivalente.

Sensori acustici e vibrazionali

- **Microfoni**: a condensatore, MEMS, piezoelettrici.
- **Sensori ultrasonici**: usati per misurare distanze e rilevare ostacoli (HC-SR04).
- **Accelerometri piezoelettrici**: specializzati per vibrazioni ad alta frequenza.

Sensori biometrici e medicali

Con l'avvento dei wearable e della medicina digitale, molti sensori sono pensati per monitorare il corpo umano.

- **Sensori di frequenza cardiaca (PPG)**: misurano variazioni di flusso sanguigno con LED e fotodiodi.
- **Sensori SpO₂ (ossigenazione sangue)**.
- **Sensori ECG (elettrocardiogramma)**.
- **Sensori EMG (attività muscolare)**.

Sviluppi futuri

Il panorama dei sensori in commercio è talmente ampio che diventa impossibile coprirlo in maniera esaustiva. Tuttavia, le categorie descritte permettono di inquadrare i principali ambiti. L'evoluzione va nella direzione dei **sensori MEMS miniaturizzati**, dei **sensori intelligenti con processing integrato** e della **connessione IoT**, che consente di raccogliere dati in tempo reale da miliardi di dispositivi. In futuro, i sensori diventeranno sempre più invisibili e diffusi, costituendo la base del concetto di "Internet of Everything".

3. Tecniche di Sensor Fusion

Quando si parla di “sensor fusion”, spesso si pensa subito a qualche formula matematica complicata o a un algoritmo avanzato di controllo. In realtà, la storia e l’evoluzione di questo concetto hanno una radice molto più umana e intuitiva: la nostra stessa percezione. Anche il nostro cervello, infatti, fa sensor fusion in continuazione. Non ci affidiamo mai a un solo senso: se vediamo male in un ambiente buio, l’uditivo e il tatto diventano più importanti; se il rumore ci confonde, ci concentriamo sulla vista; se camminiamo al buio, usiamo la memoria spaziale insieme al tatto dei piedi sul pavimento. In questo senso, i sistemi artificiali che adottano la sensor fusion cercano di imitare la logica della natura: non affidarsi mai a una sola fonte, ma integrare più informazioni per ottenere una stima più robusta, più affidabile, più realistica.

In questo testo cercheremo di raccontare, con un approccio didattico, le principali tecniche di sensor fusion, non come fredde formule matematiche, ma come strumenti nati per risolvere problemi reali.

Le origini: mediare per ridurre l’errore

Immaginiamo di avere due sensori di temperatura in una stanza. Il primo è un termometro economico, che ha un errore tipico di ± 2 °C. Il secondo è più preciso, ma tende ad avere derive se rimane acceso a lungo. Cosa fare per ottenere una stima della temperatura più vicina alla realtà?

L’idea più semplice è **fare una media**: se uno segna 21 °C e l’altro 23 °C, probabilmente la temperatura reale è intorno ai 22 °C. Se poi sappiamo che uno dei due è generalmente più affidabile, potremmo fare una **media ponderata**, dando più importanza alla misura che consideriamo migliore.

$$x_{fus} = \frac{x_1 w_1 + x_2 w_2 + \cdots x_n w_n}{w_1 + w_2 + \cdots w_n} \quad w_i: \text{pesi}, \quad x_i: \text{misure}$$

Questa logica è la base della sensor fusion: ridurre l’incertezza combinando informazioni. All’inizio, la fusione dei sensori si è sviluppata così: con regole semplici, intuitive, quasi “di buon senso”.

Filtri complementari: sfruttare punti di forza diversi

Un passo avanti rispetto alla media è stato fatto quando si è capito che i sensori non solo producono errori casuali, ma hanno **punti di forza e debolezza strutturali**. Un accelerometro, ad esempio, è ottimo per capire l'orientamento rispetto alla gravità (cioè se siamo inclinati), ma è rumoroso e facilmente disturbato da accelerazioni improvvise. Un giroscopio, invece, misura bene la velocità angolare, quindi può dire come ci stiamo muovendo in tempo reale, ma se integriamo il segnale nel tempo per ottenere l'angolo, accumula deriva.

Ecco allora l'idea del **filtro complementare**: usare un passa-basso sull'accelerometro (per tenere la sua stabilità lenta) e un passa-alto sul giroscopio (per tenere la sua reattività veloce). La fusione dei due segnali produce un angolo stabile e al tempo stesso rapido nella risposta.

Questa tecnica è stata per anni lo standard nei droni fai-da-te, nei sistemi Arduino con MPU6050 e in molti dispositivi portatili. È semplice, richiede poca potenza di calcolo, e funziona sorprendentemente bene.

In formule semplificate, il filtro si scrive così:

$$\text{Angolo filtrato} = \alpha \cdot (\text{Angolo}_{\text{precedente}} + \text{Angolo}_{gyro}) + (1 - \alpha) \cdot \text{Angolo}_{acc}$$

Ovvero, in termini matematici,

$$\theta(t) = \alpha \cdot (\theta(t-1) + \dot{\theta}_{gyro} \Delta t) + (1 - \alpha) \cdot \theta_{acc}$$

Dove:

$\theta(t)$ = angolo stimato al tempo attuale

$\theta(t-1)$ = angolo stimato al passo precedente

$\dot{\theta}_{gyro}$ = velocità angolare fornita dal giroscopio

Δt = intervallo di campionamento

θ_{acc} = angolo stimato dall'accelerometro

α = costante (tipicamente 0.95–0.99) che bilancia il peso tra i due sensori

Il termine $\alpha \cdot (\theta(t-1) + \dot{\theta}_{gyro} \Delta t)$ tiene in conto il contributo del giroscopio, moltiplicando per il tempo Δt la misura della velocità angolare $\dot{\theta}_{gyro}$ per avere la posizione angolare.

Il termine $(1 - \alpha) \cdot \theta_{acc}$ corregge lentamente la stima con l'angolo ottenuto dalla lettura dell'accelerazione.

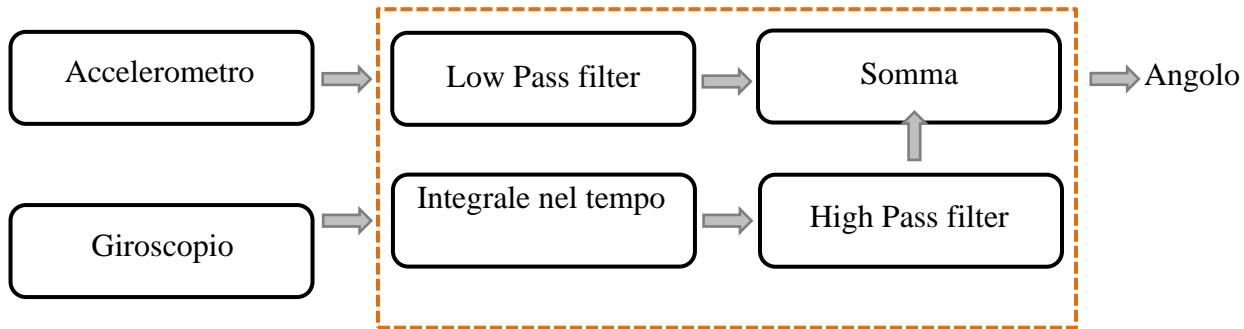


Figura 1: Struttura del filtro complementare

Cerchiamo di capire perché i coefficienti $(1 - \alpha)$ e α possono essere considerati rispettivamente un filtro passo basso ed un passa alto.

Sappiamo che la formulazione matematica per questi filtri è:

$$H_{LP}(s) = \frac{1}{1 + \tau s} \quad H_{HP}(s) = \frac{\tau s}{1 + \tau s}$$

Dove τ è la costante di tempo del filtro.

Si vede chiaramente che $H_{LP}(s) + H_{HP}(s) = 1$

Allora, se immaginiamo, $\alpha = H_{LP}(s)$ e $(1 - \alpha) = H_{HP}(s)$ si capisce, almeno intuitivamente, perché si parla di filtraggio.

Il filtro complementare è molto semplice e può essere implementato in poche righe di codice anche su microcontrollori con ridotte capacità computazionali.

Il filtro di Kalman: la rivoluzione probabilistica

Con l'aumento delle esigenze di precisione, però, è stato chiaro che media e filtri semplici non bastavano più. Negli anni '60, Rudolf Kalman sviluppò un algoritmo che avrebbe cambiato per sempre il mondo dell'ingegneria: il **filtro di Kalman**.

Il concetto è affascinante: non consideriamo le misure come valori assoluti, ma come **osservazioni rumorose di uno stato nascosto**. Lo stato (ad esempio la posizione di un satellite o l'orientamento di un drone) evolve secondo un modello dinamico, e i sensori forniscono dati imperfetti. Il filtro di Kalman usa la probabilità per "mettere insieme i pezzi": fa una predizione su dove ci aspettiamo di essere,

poi corregge questa predizione con i dati dei sensori, pesandoli in base alla loro affidabilità e rumorosità.

Il cuore del filtro è la **matrice di guadagno di Kalman**: un coefficiente calcolato automaticamente che decide quanto fidarsi della misura e quanto del modello. Se il sensore è molto rumoroso, il filtro si affida più alla predizione; se la misura è affidabile, gli dà più peso.

Questa idea, apparentemente astratta, ha avuto applicazioni enormi: la NASA lo usò nelle missioni Apollo per stimare la traiettoria dei razzi, e ancora oggi è usato nei GPS, nei droni, nei sistemi di navigazione automobilistica.

Con il tempo sono nate varianti:

- **EKF (Extended Kalman Filter)**, per sistemi non lineari, come i droni che devono stimare orientamento e posizione.
- **UKF (Unscented Kalman Filter)**, che usa punti campione per gestire meglio le non linearità.

Ognuna di queste varianti porta con sé compromessi tra complessità matematica e precisione, ma tutte hanno lo stesso spirito: usare la probabilità per fondere dati diversi.

Filtri a particelle: quando la realtà non è gaussiana

I filtri a particelle, conosciuto anche come Sequential Monte Carlo (SMC) method, è una delle tecniche più potenti e flessibili di sensor fusion e di stima dello stato in sistemi complessi. A differenza del filtro di Kalman e delle sue varianti (EKF, UKF), che assumono che gli errori e le distribuzioni siano di tipo gaussiano e che i modelli siano lineari (o linearizzabili), il filtro a particelle è un metodo non parametrico. Significa che non si limita a rappresentare l'incertezza con una media e una varianza, ma può descrivere distribuzioni arbitrarie, anche multimodali, cioè con più "picchi" di probabilità.

Questa caratteristica lo rende ideale quando si ha a che fare con sistemi non lineari, rumorosi e complessi, dove i modelli tradizionali rischiano di essere troppo semplificativi.

L'idea alla base del filtro a particelle è molto intuitiva: invece di provare a stimare lo stato di un sistema con una formula chiusa (come fa il Kalman), si rappresenta la distribuzione di probabilità dello stato attraverso un insieme di campioni discreti, detti particelle. Ogni particella rappresenta una possibile ipotesi sullo stato del sistema. Insieme, tutte le particelle formano una "nuvola" che approssima la

distribuzione di probabilità. Ogni particella ha un peso, che indica quanto è plausibile rispetto alle osservazioni.

Il filtro a particelle procede iterativamente:

- **Predizione:** le particelle vengono propagate in avanti nel tempo seguendo il modello dinamico del sistema (es. equazioni del moto di un robot).
- **Aggiornamento:** si confrontano le particelle con la misura proveniente dai sensori e si aggiornano i pesi. Le particelle che “spiegano meglio” la misura hanno peso maggiore.
- **Resampling:** le particelle con peso molto basso vengono eliminate, mentre quelle con peso alto vengono “replicate”, in modo che la nuvola resti concentrata nelle regioni più probabili.

Con il passare delle iterazioni, le particelle si distribuiscono sempre più vicino allo stato reale del sistema.

Facciamo un esempio. Immaginiamo di avere un robot che si muove in un corridoio e vogliamo stimare la sua posizione lungo l'asse X. Il robot ha una stima imprecisa del movimento (deriva delle ruote, slittamento, ecc.), e ogni tanto riceve una misura di distanza da una porta (con rumore).

All'inizio non sappiamo dove sia il robot quindi generiamo tante particelle distribuite lungo tutto il corridoio.

Quando il robot si muove, tutte le particelle si spostano in base al modello di moto, ma in maniera “rumorosa”.

Quando il sensore rileva una porta, si confronta la posizione prevista di ogni particella con la posizione reale della porta: quelle che coincidono meglio ricevono un peso maggiore.

Successivamente, mediante il resampling, si eliminano le particelle incoerenti e vengono raddoppiate quelle plausibili.

Dopo varie iterazioni, anche se il sensore è rumoroso e il modello non è perfetto, la maggior parte delle particelle si concentra attorno alla posizione reale del robot. I filtri a particelle sono usati molto in robotica, specialmente nella localizzazione in ambienti sconosciuti (SLAM, simultaneous localization and mapping). Un robot può così costruire una mappa mentre capisce dove si trova, fondendo sensori come lidar, odometria, GPS e visione.

Deep Sensor Fusion

Negli ultimi anni, con l'aumento della potenza di calcolo e l'avvento del deep learning, sono emerse tecniche di sensor fusion basate su reti neurali e algoritmi di apprendimento automatico. Reti neurali profonde possono imparare automaticamente a pesare i contributi dei vari sensori per un certo compito, senza necessità di modellazione manuale. Ad esempio, in veicoli autonomi si usano reti neurali che integrano immagini da telecamere, nuvole di punti Lidar e dati radar. Se da un lato questa tecnologia permette di gestire dati eterogenei e non lineari, apprendere pattern complessi non modellabili con formule matematiche semplici, dall'altro richiede una notevole capacità computazionale e grandi quantità di dati per l'addestramento.

Sensor fusion gerarchica: dati, feature e decisioni

Non tutta la fusione avviene a livello di dati grezzi. Si possono distinguere tre livelli:

- **Data level:** fusione diretta dei dati (ad esempio media delle misure di temperatura).
- **Feature level:** si estraggono caratteristiche dai segnali e si fondono (ad esempio riconoscere la direzione di un movimento combinando accelerazioni e rotazioni).
- **Decision level:** ogni sensore fornisce una decisione (ad esempio "ostacolo presente"), e queste decisioni vengono fuse (ad esempio con voto a maggioranza).

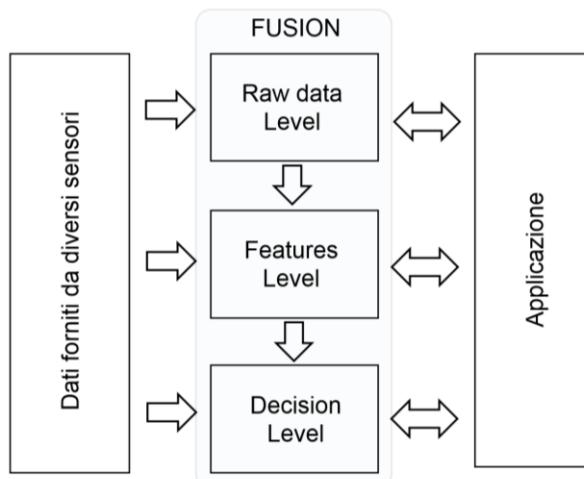


Figura 2: Gerarchia "sensor fusion"

Questi tre livelli non sono mutuamente esclusivi, ma piuttosto rappresentano una gerarchia logica: più si sale, più ci si allontana dal dato grezzo per arrivare a una rappresentazione sintetica e “interpretata” del mondo.

- **Data-level Fusion: unire i dati grezzi**

Nel livello più basso della gerarchia troviamo la fusione a livello di dati. Qui i sensori forniscono direttamente i loro output grezzi, ad esempio valori di tensione, conteggi digitalizzati, misure di accelerazione, intensità luminosa, segnali acustici, e questi vengono combinati senza un’elaborazione intermedia significativa.

Vantaggi:

- Massima informazione preservata: nessun dettaglio viene perso prima della fusione.
- Riduzione del rumore: più sensori che misurano la stessa grandezza possono essere mediati per attenuare il rumore casuale.
- Alta risoluzione: i dati combinati mantengono tutta la precisione disponibile.

Svantaggi

- Grande mole di dati: bisogna elaborare segnali spesso ridondanti e voluminosi.
- Dipendenza dall’allineamento temporale: è cruciale sincronizzare i sensori.
- Sensibilità a bias sistematici: se tutti i sensori hanno lo stesso errore, la fusione non lo elimina.

- **Feature-level Fusion: unire le caratteristiche**

Il secondo livello della gerarchia è quello della fusione delle feature, o caratteristiche. Qui non si lavora più sui dati grezzi, ma su grandezze derivate ed estratte da essi. Per esempio, da un microfono non si prende l’intero segnale audio, ma solo lo spettro delle frequenze dominanti; Le feature rappresentano quindi informazioni compresse ma altamente descrittive. La fusione avviene confrontando e combinando tali caratteristiche.

Vantaggi

- Riduzione della dimensionalità: meno dati da gestire, più facilità di calcolo.
- Maggiore robustezza: si riducono le componenti ridondanti e rumorose.
- Più flessibilità: diversi tipi di sensori possono fornire feature comparabili (es. rilevamento movimento da accelerometro e da visione).

Svantaggi

- Perdita di informazione: parte del segnale originale viene scartata.
- Scelta delle feature non banale: richiede conoscenza del dominio.

Un esempio pratico: in un sistema di riconoscimento gesti, un accelerometro fornisce la direzione media del movimento e una telecamera fornisce la traiettoria estratta da un video. La fusione a livello di feature può unire queste due informazioni in un vettore di movimento più accurato.

• **Decision-level Fusion: unire le decisioni**

Il livello più alto della gerarchia è la fusione a livello decisionale. Qui ogni sensore (o ogni modulo di elaborazione) produce una propria decisione autonoma, ad esempio “l’oggetto è presente”, “la persona sta camminando”, “il veicolo è fermo”, e tali decisioni vengono fuse per arrivare a una decisione finale.

Questa modalità è spesso usata in sistemi di sicurezza, diagnostica o classificazione, dove i sensori sono molto eterogenei.

Vantaggi

- Indipendenza dai dati grezzi: ogni sensore può essere trattato come una “scatola nera”.
- Facilità di integrazione: sensori anche molto diversi possono essere fusi senza problemi.
- Robustezza a guasti: se un sensore sbaglia, gli altri possono correggere la decisione.

Svantaggi

- Bassa granularità: non si può più “andare a vedere nei dati” per capire dettagli.
- Possibili conflitti: decisioni divergenti devono essere risolte con regole o votazioni.

Un esempio classico: in un sistema di guida autonoma, un radar decide che c’è un ostacolo, una telecamera lo conferma e un lidar lo individua con precisione. Le decisioni finali vengono fuse per dire “c’è un ostacolo a 15 m, in direzione 30°”.

La Sensor Fusion gerarchica quindi non è una sola tecnica, ma un concetto organizzativo. Al livello più basso (data-level) si massimizza l’informazione ma si ha un forte onere computazionale e problemi di sincronizzazione; al livello intermedio (feature-level) si cercano rappresentazioni compatte e più robuste; al livello alto (decision-level) si lavora con informazione già “digerita”, sacrificando dettaglio ma guadagnando in robustezza e modularità.

In molti sistemi reali i tre livelli convivono: ad esempio, in un’auto autonoma i sensori radar e lidar possono fondere i dati grezzi per avere mappe accurate (data-level), queste mappe possono poi generare feature come i bordi della strada (feature-level), e infine il sistema di controllo può fondere le decisioni “la strada è libera” o “c’è un pedone” con altre fonti (decision-level).

La Sensor Fusion gerarchica, quindi, rappresenta una scala di astrazione: si parte dai dati crudi e si sale verso interpretazioni sempre più sintetiche, fino ad arrivare alla decisione finale.

Sfide pratiche della sensor fusion

Quando si passa dalla teoria alla pratica, l’implementazione della sensor fusion su sistemi reali porta con sé una serie di sfide e criticità ricorrenti, tra cui:

- **Sincronizzazione:** i sensori non sono mai perfettamente allineati nel tempo.
- **Calibrazione:** ogni sensore ha un offset, una deriva, una sensibilità diversa.
- **Gestione dei guasti:** cosa succede se un sensore improvvisamente smette di funzionare o fornisce dati assurdi?
- **Scalabilità:** passare da due sensori a decine o centinaia richiede strategie nuove.

Queste sfide mostrano che la sensor fusion non è solo matematica: è anche ingegneria dei sistemi, gestione dei dati, progettazione robusta.

Un futuro di fusione ibrida

Il futuro della sensor fusion sembra andare verso un ibrido: da un lato, la solidità dei modelli matematici (come Kalman e particelle), dall'altro la flessibilità dell'intelligenza artificiale. L'idea è che i modelli classici possano garantire robustezza e interpretabilità, mentre il machine learning può gestire complessità e dati difficili da modellare.

In questo senso, la sensor fusion diventerà sempre più una **disciplina interdisciplinare**, a metà tra matematica, informatica, ingegneria e persino neuroscienze.

Esempio pratico di Sensor Fusion in automotive

Immagina un'auto moderna dotata di diversi sensori:

- **Radar**: misura la distanza e la velocità relativa degli oggetti circostanti. È molto robusto anche in condizioni di pioggia o nebbia.
- **Telecamera**: riconosce corsie, segnali stradali, pedoni e veicoli. Ha grande dettaglio, ma soffre in condizioni di scarsa visibilità.
- **Lidar**: crea una mappa 3D dell'ambiente con altissima precisione spaziale.
- **IMU (accelerometri + giroscopi)**: stima i movimenti e l'assetto dell'auto in tempo reale.
- **GPS**: fornisce la posizione assoluta sul globo.

Singolarmente, ogni sensore ha **limiti e incertezze**.

Con la sensor fusion, invece, i dati vengono combinati in tempo reale tramite algoritmi (es. Filtro di Kalman esteso, Particle Filter, o reti neurali), così da ottenere una percezione dell'ambiente molto più affidabile e robusta. Prendiamo come esempio la frenata automatica di emergenza (AEB):

1. Il radar rileva un ostacolo a 30 metri.
2. La telecamera conferma che è un pedone, non un cartello stradale.
3. L'IMU verifica che l'auto non sta sterzando bruscamente (quindi la traiettoria è dritta verso il pedone).
4. Il sistema decide di attivare la frenata automatica.

Risultato: l'auto evita l'incidente perché l'informazione combinata dei sensori è più sicura di quella proveniente da uno solo.

4. Sensor fusion con filtro di Kalman

Quando si parla di sensor fusion, spesso il primo esempio concreto che viene proposto è l'integrazione tra accelerometro e giroscopio. Non si tratta di una scelta casuale: questi due sensori rappresentano un punto di partenza ideale per comprendere come la combinazione di misure apparentemente semplici possa condurre a stime robuste e utilizzabili in applicazioni reali. Più che il risultato finale, ossia il calcolo di un angolo di inclinazione stabile e affidabile, ciò che conta davvero in questo capitolo è il percorso logico e metodologico che ci conduce alla soluzione.

L'idea di fondo è semplice: raramente esiste un sensore perfetto. Ogni dispositivo di misura è soggetto a limitazioni, rumore, derive e incertezze. L'accelerometro, ad esempio, ci fornisce un'informazione diretta sulla direzione della gravità e quindi sull'inclinazione rispetto alla verticale. Tuttavia, la misura è estremamente sensibile alle vibrazioni e ai movimenti rapidi: basta agitare un dispositivo o montarlo su un veicolo in movimento per rendersi conto che i valori oscillano vistosamente. Il giroscopio, dal canto suo, misura la velocità angolare con grande reattività e precisione nel breve termine, ma soffre di un problema opposto: l'integrazione del segnale nel tempo porta inevitabilmente a una deriva crescente, tanto più marcata quanto più lungo è l'intervallo di osservazione.

Questi due esempi già ci pongono di fronte a un dilemma tipico dell'ingegneria: nessuna delle due fonti informative è sufficiente da sola, eppure entrambe contengono un frammento della verità. La soluzione naturale è quindi combinarli: affidarsi al giroscopio per cogliere la dinamica a breve termine e all'accelerometro per correggere la deriva nel lungo periodo. Questa intuizione è la base stessa della sensor fusion, che possiamo vedere qui in una forma semplice ma altamente rappresentativa.

Il passo successivo è tradurre questa intuizione in un modello matematico e algoritmico. Ed è qui che entra in gioco il filtro di Kalman, uno degli strumenti più eleganti e potenti sviluppati nel campo del controllo e dell'elaborazione dei segnali. Il filtro di Kalman rappresenta un'applicazione rigorosa della teoria della probabilità e dell'ottimizzazione: partendo da un modello di stato che descrive come il sistema evolve nel tempo, e da una descrizione statistica degli errori di misura, il filtro fornisce la miglior stima possibile (nel senso di minimo errore quadratico medio) della grandezza d'interesse.

Vale la pena sottolineare un aspetto importante: l'approccio che seguiremo non è pensato per fornire una “ricetta” da copiare alla lettera, ma per insegnare un modo di pensare. Non basta sapere che un filtro di Kalman funziona: bisogna comprendere quali ipotesi lo sostengono, quali parametri ne determinano il comportamento, e come queste scelte si riflettono sul risultato. Solo così si diventa in grado di adattare la tecnica a nuovi problemi, che inevitabilmente si presenteranno con caratteristiche diverse.

In questo capitolo, dunque, ci muoveremo su più livelli: partiremo dalla descrizione dei sensori (accelerometro e giroscopio), per capire a fondo cosa misurano e quali sono i loro limiti intrinseci. Poi introdurremo il problema della stima dell'orientamento, mostrando perché nessun sensore, preso singolarmente, è sufficiente. A quel punto, discuteremo la strategia di fusione basata sul filtro di Kalman. Lo scopo non è soltanto mostrare i calcoli, ma soprattutto rendere chiaro il ragionamento dietro ogni passaggio.

Il giroscopio

Il giroscopio è uno dei sensori più affascinanti e utili in ambito elettronico moderno. Nasce come strumento meccanico nell'Ottocento per mantenere la stabilità delle navi, grazie a un rotore che conserva la sua direzione nello spazio. Oggi però, grazie alla microelettronica, i giroscopi sono realizzati in tecnologia MEMS (Micro-Electro-Mechanical Systems) e funzionano senza parti in movimento visibili.

Un **giroscopio MEMS** è un sensore che misura la velocità angolare, cioè la velocità con cui un corpo ruota attorno a uno o più assi. È ampiamente usato in smartphone, droni, robotica, videogame controller e sistemi di navigazione, perché consente di stimare l'orientamento di un oggetto nello spazio.

Il principio di base sfrutta la **forza di Coriolis**, un fenomeno fisico che agisce su un corpo in movimento quando questo si trova in un sistema di riferimento rotante. Ricordiamo che la forza di Coriolis è una forza apparente che devia la traiettoria di un corpo in movimento su un sistema in rotazione. Questa forza è pari a:

$$\vec{F}_C = 2 m \vec{\omega} \times \vec{v}$$

- F_C = forza di Coriolis
- M = massa

- ω = velocità angolare
- v = velocità tangenziale

Questa forza è **perpendicolare sia al movimento della massa sia all'asse di rotazione.**

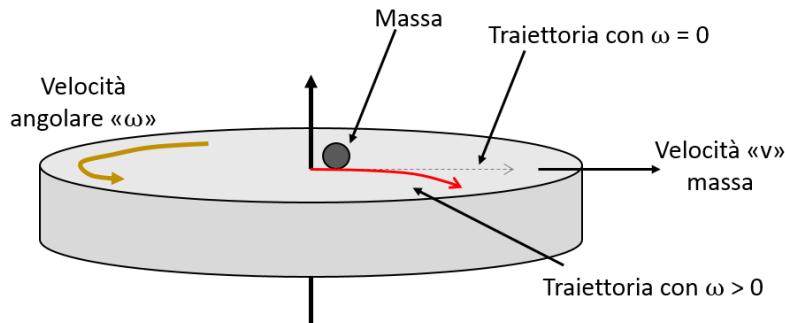


Figura 3: Principio della forza di Coriolis

Un **giroscopio MEMS** basato sull'effetto Coriolis funziona grazie all'interazione tra vibrazioni forzate e rotazioni applicate. Al suo interno troviamo una **piccola massa mobile** (*proof mass*), sospesa da microscopiche molle elastiche, che viene mantenuta in **oscillazione controllata** lungo un asse specifico chiamato *asse di eccitazione*. Quando il giroscopio non ruota, la massa vibra semplicemente lungo il suo asse di eccitazione, senza che compaiano spostamenti sull'altro asse. Ma nel momento in cui il dispositivo subisce una **rotazione attorno a un asse perpendicolare al piano del sensore**, entra in gioco la **forza di Coriolis**. Questa forza, proporzionale alla velocità angolare, provoca una deviazione della massa oscillante lungo l'asse ortogonale, generando un piccolo spostamento che può essere rilevato tramite condensatori ad armature mobili, che misurano variazioni minime di capacità elettrica dovute allo spostamento. L'elettronica integrata traduce poi questa variazione in un segnale proporzionale alla velocità angolare, consentendo di sapere **quanto velocemente e in quale direzione sta ruotando il dispositivo**.

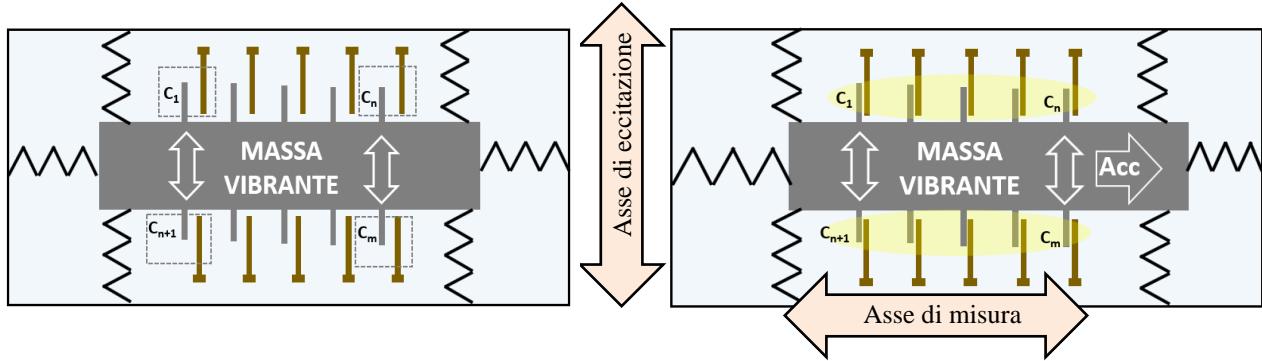


Figura 4: Struttura interna di un giroscopio MEMS (a sinistra in condizioni di riposo, a destra sottoposto a rotazione)

Nel mondo dei microcontrollori, i giroscopi sono fondamentali per applicazioni come stabilizzazione di droni e robot, riconoscimento dei gesti in smartphone e controller di videogiochi.

Il pregio del giroscopio è la capacità di rilevare con grande precisione i movimenti rapidi, ma presenta almeno due difetti importanti che occorre tenere in debita considerazione. Vediamoli in dettaglio.

L'Angle Random Walk rappresenta l'effetto del **rumore casuale ad alta frequenza** presente nell'uscita del giroscopio.

Un giroscopio ideale dovrebbe restituire una velocità angolare costante se il dispositivo è fermo. Nella pratica, invece, l'elettronica e i processi di misura introducono un **rumore bianco** (completamente casuale) che si sovrappone al segnale. Poiché l'angolo stimato si ottiene **integrando la velocità angolare nel tempo**, anche un piccolo rumore casuale si accumula, producendo una **deviazione angolare che cresce con la radice del tempo**. Questo è ciò che chiamiamo *random walk angolare*: il sensore “cammina” lontano dal valore reale senza una direzione precisa, come un ubriaco.

Se il giroscopio ha un rumore bianco con densità spettrale N [$^{\circ}/\sqrt{h}$], l'errore angolare cresce con la legge:

$$\sigma_\theta(t) \approx N \cdot \sqrt{t}$$

$\sigma_\theta(t)$ è la deviazione standard dell'angolo,
 t è il tempo di integrazione.

Il termine \sqrt{t} indica che più a lungo usiamo il giroscopio senza correzioni esterne, più l'errore sull'angolo stimato si allarga a causa dell'ARW.

La Bias Instability rappresenta la **variazione lenta e casuale del bias del giroscopio nel tempo**.

Il **bias** è l'offset che il sensore mostra anche quando è fermo. Per esempio, un giroscopio perfetto dovrebbe leggere 0 °/s quando non c'è rotazione. Ma nella realtà c'è sempre un piccolo offset (es. +0.05 °/s). Se fosse fisso, potremmo compensarlo con una calibrazione iniziale. Il problema è che **questo bias non è stabile**. A causa di diversi effetti (termici, vibrazioni meccaniche, ...) il bias oscilla lentamente nel tempo, in modo imprevedibile.

È importante osservare che il bias **non** si manifesta come rumore bianco, ma come una **drift lenta e quasi costante** che cambia su scale temporali di secondi o minuti.

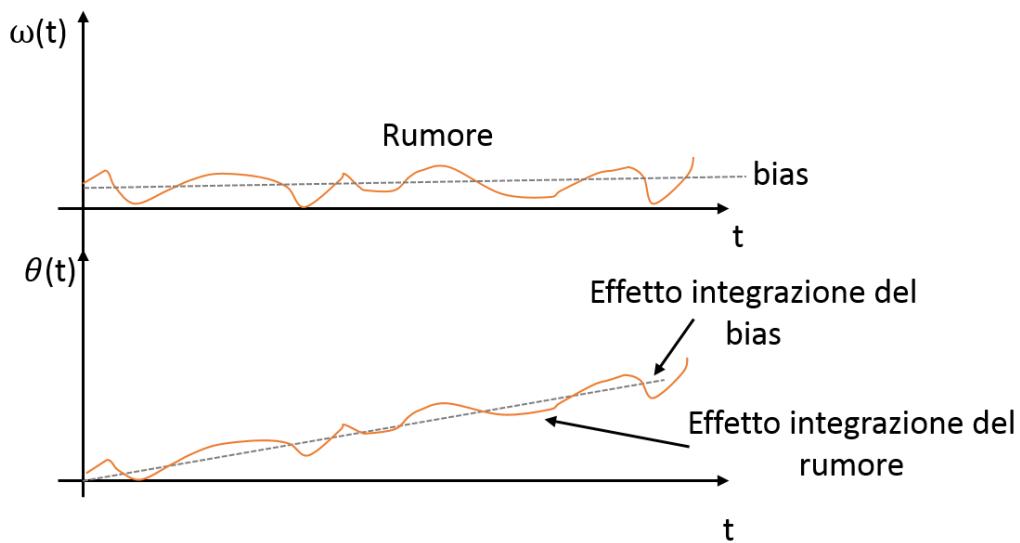


Figura 5: Effetto del rumore e del Bias nella misura dell'angolo

Una volta compreso che la Bias Instability rappresenta una deriva lenta e imprevedibile del segnale del giroscopio, la domanda naturale è: come si può mitigare questo fenomeno?

Non esiste una soluzione unica e definitiva, ma una strategia efficace consiste nell'affiancare al giroscopio altre fonti di informazione che possano “correggere” le sue derive. In altre parole, invece di affidarsi ciecamente a un solo sensore, si sfrutta la sensor fusion, cioè la combinazione di più misure per arrivare a una stima più robusta.

L'Accelerometro

L'accelerometro, come suggerisce il nome, misura l'accelerazione lineare lungo uno o più assi. Anche qui, nella versione MEMS, la misura si ottiene rilevando

minuscole variazioni di capacità elettrica dovute allo spostamento di masse interne microscopiche quando il sensore subisce una forza.

Un accelerometro a 3 assi fornisce tre valori (a_x, a_y, a_z), che rappresentano le accelerazioni lungo ciascun asse del sistema di riferimento del sensore. La particolarità, come già visto, è che anche quando il sensore è fermo, esso “vede” sempre l’accelerazione di gravità (circa 9.81 m/s^2) proiettata sui tre assi. Questo fatto permette di usarlo come un “inclinometro”, cioè per ricavare l’angolo di rollio e beccheggio rispetto alla verticale.

Gli accelerometri sono molto diffusi e vengono utilizzati nei telefoni cellulari (per cambiare l’orientamento dello schermo), nei sistemi airbag delle automobili, nei contapassi e dispositivi fitness, nei sistemi di navigazione per droni e robot.

Gli accelerometri **MEMS** funzionano sfruttando **microstrutture meccaniche integrate in un chip di silicio**, che rispondono alle forze di accelerazione con uno spostamento fisico rilevabile elettricamente.

Un accelerometro MEMS è composto da:

1. **Massa sospesa (proof mass):** Una piccolissima massa di silicio (pochi microgrammi) che rappresenta l’elemento sensibile.
2. **Molle di sospensione:** Sostengono la massa e permettono uno spostamento controllato in una o più direzioni. Hanno una costante elastica k che governa la relazione tra forza e spostamento: $F=k\cdot x$.
3. **Elettrodi fissi e mobili:** Funzionano come condensatori variabili: la massa è collegata ad armature che, muovendosi rispetto a elettrodi fissi, modificano la capacità elettrica.
4. **Circuiteria elettronica integrata:** Amplifica e converte in digitale la variazione di capacità, trasformandola in un valore di accelerazione.

La figura seguente mostra, a livello intuitivo, la struttura interna di un accelerometro MEMS. Distinguiamo la massa sospesa, le molle calibrate e i condensatori “a pettine” ($C_1 \dots C_m$).

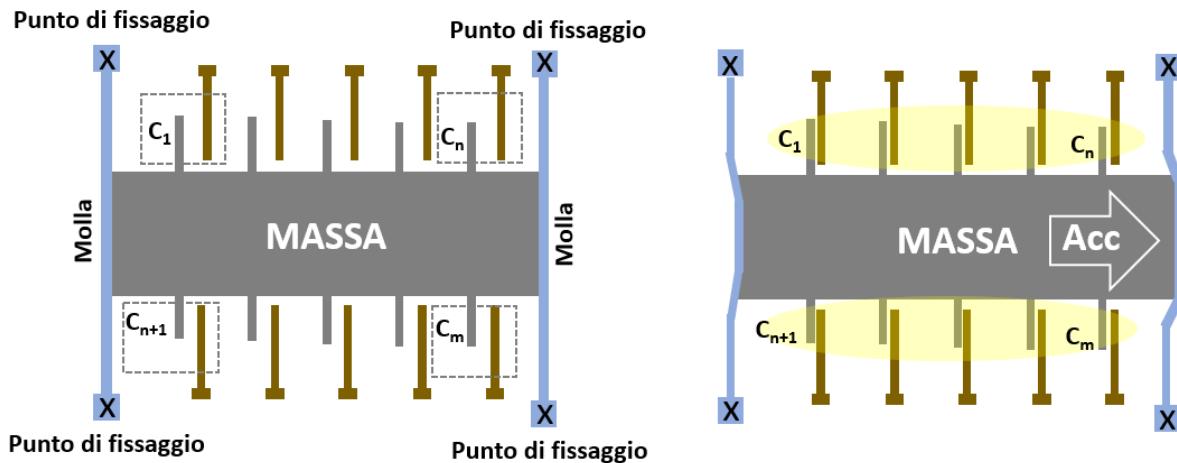


Figura 6: Struttura interna di un accelerometro MEMS

L'accelerometro MEMS si comporta come un **oscillatore massa-molla** con smorzamento viscoso.

Equazione del moto: $m\ddot{x}(t) + b\dot{x}(t) + k x(t) = -ma(t)$

dove:

m = massa sospesa,

b = coefficiente di smorzamento della massa nel chip,

k = costante elastica delle molle,

$x(t)$ = spostamento della massa, $\ddot{x}(t)$ accelerazione della massa, $\dot{x}(t)$ velocità della massa.

$a(t)$ = accelerazione applicata al chip.

Se la frequenza dell'accelerazione è molto inferiore alla frequenza di risonanza del sistema, la relazione si semplifica in:

$$x(t) = \frac{-ma(t)}{k}$$

Cioè, lo spostamento della massa è **direttamente proporzionale all'accelerazione**.

Quando la massa si muove modifica la configurazione dei condensatori a pettine modificando la distanza tra le armature. Ricordiamo che la capacità di un condensatore piano si calcola attraverso la relazione:

$$C = \epsilon \frac{A}{d}$$

ϵ è la costante dielettrica,
A è l'area delle armature,
d è la distanza tra di esse.

Quando la massa sospesa si sposta per effetto dell'accelerazione, cambia d (o l'area di sovrapposizione), quindi cambia C.

Un apposito circuito, integrato nel sensore, legge la **differenza di capacità**, proporzionale allo spostamento, quindi all'accelerazione e genera un segnale d'uscita (analogico o digitale).

Il sensore MPU-6050

L'MPU6050 è probabilmente uno dei sensori più diffusi e apprezzati nel mondo dell'elettronica embedded e del fai-da-te tecnologico. Si tratta di un IMU (Inertial Measurement Unit) a 6 gradi di libertà (6-DOF) che integra in un unico chip due strumenti fondamentali: un accelerometro a tre assi e un giroscopio a tre assi. In altre parole, con un componente di dimensioni ridottissime e dal costo contenuto è possibile ottenere informazioni sia sulle accelerazioni lineari sia sulle velocità angolari, cioè sui movimenti e le rotazioni nello spazio.

Questa combinazione rende l'MPU6050 estremamente versatile: è il cuore di molti progetti di robotica amatoriale, di sistemi di stabilizzazione (ad esempio per droni o gimbal), di dispositivi indossabili, di giochi elettronici basati su gesture.

L'MPU6050, prodotto originariamente da InvenSense (poi acquisita da TDK), si distingue per alcune caratteristiche tecniche che hanno contribuito alla sua enorme diffusione:

- Accelerometro a 3 assi: misura le accelerazioni lungo gli assi X, Y e Z. È configurabile su diversi fondi scala ($\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$), permettendo di adattare la sensibilità in base all'applicazione.
- Giroscopio a 3 assi: misura le velocità angolari (quanto velocemente il dispositivo ruota) con fondi scala configurabili (± 250 , ± 500 , ± 1000 , ± 2000 °/s).
- Risoluzione: I dati di accelerometro e giroscopio sono numeri interi con segno a 16 bit.

- Interfaccia digitale I²C (o SPI in alcune versioni): permette una comunicazione semplice e veloce con microcontrollori come Arduino o ESP32.
- Frequenza di campionamento elevata: può arrivare fino a diverse centinaia di Hz, rendendolo adatto a sistemi di controllo in tempo reale.
- Registri interni e FIFO buffer: riducono il carico di lavoro del microcontrollore, permettendo di accumulare campioni e leggerli in blocco.
- Digital Motion Processor (DMP): un co-processore integrato che può elaborare direttamente i dati grezzi dei sensori per fornire stime già filtrate di orientamento, riducendo così la necessità di calcoli complessi lato microcontrollore.

Per ulteriori informazioni in merito a questo MEMS vi consiglio di consultare il datasheet del produttore.

Rappresentazione dell'orientamento con le matrici di Eulero

Dopo aver descritto l'MPU6050 e compreso le sue principali caratteristiche, possiamo renderci conto di quanto sia straordinario avere a disposizione, in un unico chip, un accelerometro e un giroscopio entrambi triassiali. Questo dispositivo rappresenta una sorta di "finestra sensoriale" che permette a un sistema elettronico di percepire il proprio orientamento nello spazio e i propri movimenti dinamici. Ma avere i dati grezzi del sensore non basta: essi devono essere interpretati, elaborati e trasformati in grandezze più significative, come gli **angoli di pitch e roll (beccheggio e rollio)**, o, in una visione più completa, gli **angoli di Eulero** che descrivono lo stato di rotazione di un corpo rigido nello spazio tridimensionale.

Se ci fermassimo al livello dei segnali misurati dal sensore, avremmo infatti soltanto valori numerici corrispondenti a accelerazioni lungo gli assi x,y,z e velocità angolari lungo gli stessi assi. Ma come possiamo passare da queste quantità a un'informazione più intuitiva, ad esempio l'inclinazione di un robot, la rotazione di un drone o la posizione di una piattaforma stabilizzata? La risposta risiede in un quadro matematico ben definito, che prende origine dai lavori di **Leonhard Euler**, uno dei più grandi matematici del XVIII secolo.

Euler, nel 1775, dimostrò un teorema fondamentale: **qualsiasi spostamento di un corpo rigido con un punto fisso nello spazio può essere descritto come**

una rotazione attorno a un asse passante per quel punto. Questo teorema, apparentemente astratto, ha conseguenze molto pratiche: significa che, invece di dover immaginare trasformazioni complicate nello spazio, possiamo sempre ridurre il problema a un insieme di rotazioni attorno ad assi opportunamente scelti. Da qui nasce la definizione degli **angoli di Eulero**, una tripletta di angoli che, applicati in sequenza, descrive qualsiasi rotazione possibile di un corpo.

Il legame tra i dati dell'MPU6050 e gli angoli di Eulero passa attraverso le cosiddette **matrici di rotazione**. Queste matrici non sono altro che rappresentazioni compatte e rigorose delle trasformazioni geometriche che permettono di passare da un sistema di riferimento a un altro. Immaginiamo, ad esempio, un piccolo drone equipaggiato con un MPU6050: il sensore misura accelerazioni e rotazioni rispetto al proprio sistema di assi locali (che chiameremo **sistema corpo**). Tuttavia, ciò che spesso ci interessa non è tanto la misura rispetto a quegli assi, quanto la posizione e l'orientamento del drone rispetto a un sistema di riferimento esterno, come il sistema di coordinate del mondo reale (che chiameremo **sistema inerziale** o **sistema terra**).

Per tradurre queste informazioni da un sistema all'altro abbiamo bisogno delle **matrici di rotazione**, che stabiliscono il collegamento matematico tra il sistema corpo e il sistema terra. Attraverso queste matrici possiamo trasformare le accelerazioni misurate dal sensore in informazioni relative all'orientamento. Ad esempio, quando l'accelerometro è fermo e non ci sono altre accelerazioni oltre alla gravità, la direzione del vettore di gravità rispetto al sistema corpo ci permette di dedurre l'assetto del dispositivo. Ma per compiere questa deduzione dobbiamo proiettare correttamente le componenti del vettore di gravità attraverso le rotazioni descritte dagli angoli di Eulero.

Dunque, l'introduzione alle matrici di rotazione non è un esercizio teorico fine a se stesso, ma un passaggio obbligato per chiunque voglia lavorare con sensori inerziali come l'MPU6050. Esse costituiscono il ponte tra il mondo delle **misure grezze** e quello delle **informazioni utili**. Attraverso le matrici di rotazione possiamo formalizzare il concetto di orientamento, ricavare le formule per il calcolo di pitch e roll a partire dalle componenti dell'accelerazione, e successivamente integrare queste formule in sistemi di fusione sensoriale più complessi.

Per rendere l'idea con un esempio concreto, immaginiamo di voler stabilizzare una piattaforma gimbal con due assi (pitch e roll) utilizzando un MPU6050. Il sensore ci fornisce in ogni istante i valori di accelerazione lungo gli assi x,y,z. Se

la piattaforma è perfettamente orizzontale, l'unica accelerazione misurata sarà quella della gravità, interamente allineata all'asse z. Ma se la piattaforma si inclina, parte di questa accelerazione apparirà lungo gli assi x e y. Attraverso le formule derivate dalle matrici di rotazione, possiamo calcolare di quanto la piattaforma si è inclinata e comandare i servomotori per compensare l'errore.

Dunque, partendo dai dati grezzi che riceviamo, ovvero le **componenti del vettore gravità** proiettate lungo i tre assi del sensore (g_x , g_y , g_z), dobbiamo calcolare gli **angoli di roll (ϕ)** e di **pitch (θ)**.

- **Roll (ϕ)** rappresenta l'inclinazione del sensore rispetto all'asse longitudinale, cioè una rotazione attorno all'asse x. Per immaginarlo, pensa a un aereo: il roll è il movimento che l'aereo compie quando si inclina a destra o a sinistra, come per virare. Allo stesso modo, se teniamo in mano il sensore e lo ruotiamo lateralmente, l'angolo di roll descrive questo movimento.
- **Pitch (θ)** invece descrive l'inclinazione attorno all'asse trasversale, cioè una rotazione attorno all'asse y. Nell'esempio dell'aereo, il pitch è l'angolo che si ottiene quando l'aereo alza il muso verso l'alto o lo abbassa verso il basso. Per il nostro sensore, questo equivale a inclinarlo in avanti o indietro.

Insieme, ϕ (roll) e θ (pitch) descrivono l'orientamento del dispositivo rispetto al piano orizzontale: il primo indica l'inclinazione laterale (verso destra o sinistra), mentre il secondo rappresenta la pendenza in avanti o all'indietro.

Esiste anche un terzo angolo, detto yaw, che corrisponde alla rotazione attorno all'asse verticale (asse z). Tuttavia, in questo contesto non lo consideriamo, perché per stimarlo in maniera affidabile sarebbe necessario un magnetometro, ovvero un sensore capace di misurare il campo magnetico terrestre e quindi fornire un riferimento assoluto per la direzione.

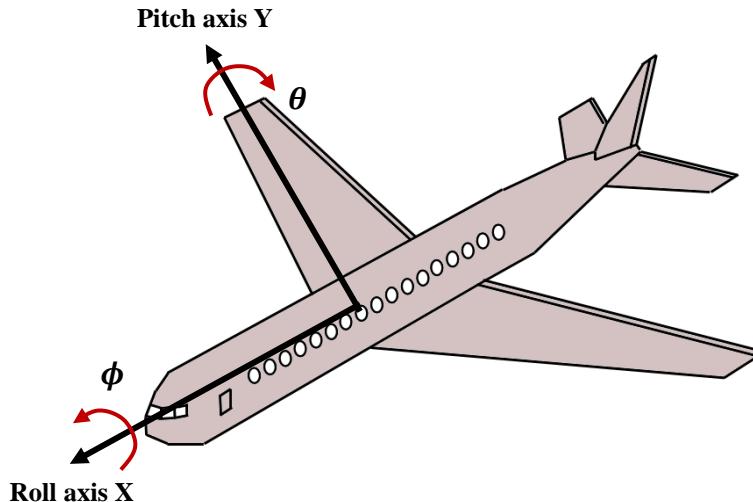


Figura 7: Concetti di Pitch e Roll in un velivolo

Supponiamo di mantenere in condizioni statiche l'accelerometro oppure *in moto uniforme*, cioè non soggetto ad accelerazioni lineari significative oltre la gravità. Sotto questa ipotesi si dimostra che le proiezioni di "g" lungo gli assi del dispositivo valgono:

$$\begin{aligned} a_x &= g \sin \theta \\ a_y &= -g \cos \theta \sin \phi \\ a_z &= -g \cos \theta \cos \phi \end{aligned}$$

Dove ϕ (Roll) rappresenta l'inclinazione del sensore rispetto all'asse x e θ (Pitch) è l'inclinazione rispetto all'asse y.

Facciamo una verifica di coerenza: se $\phi=0$ e $\theta=0$ allora $a_x=0$, $a_y=0$, $a_z=-g$ cioè il sensore misura $-g$ sull'asse z quando è orizzontale.

Da queste equazioni dobbiamo ricavare gli angoli θ e ϕ .

Iniziamo a ricavare il roll ϕ .

$$\frac{a_y}{a_z} = \frac{-g \cos \theta \sin \phi}{-g \cos \theta \cos \phi} = \tan \phi$$

Quindi

$$\phi = \arctan \frac{a_y}{a_z}$$

Ricaviamo il pitch θ .

Calcoliamo:

$$\begin{aligned} a_y^2 + a_z^2 &= (-g \cos \theta \sin \phi)^2 + (-g \cos \theta \cos \phi)^2 = g^2 \cos^2 \theta (\sin^2 \phi + \cos^2 \phi) \\ &= g^2 \cos^2 \theta \end{aligned}$$

(ricordiamo che $\sin^2 \phi + \cos^2 \phi = 1$, quindi ϕ “scompare” dai calcoli)

Dalla relazione ottenuta, ricaviamo:

$$\sqrt{a_y^2 + a_z^2} = g \cos \theta$$

Ora, prendiamo la relazione:

$$a_x = g \sin \theta$$

e calcoliamo:

$$\frac{a_x}{\sqrt{a_y^2 + a_z^2}} = \frac{g \sin \theta}{g \cos \theta} = \tan \theta$$

Quindi:

$$\theta = \arctan\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right)$$

Bene, ora sappiamo come ricavare θ e ϕ a partire dalle proiezioni di “g” sugli assi dell’accelerometro.

Il giroscopio fornisce direttamente i valori di velocità angolare rilevati sul sistema di riferimento condiviso con l’accelerometro e non necessita di ulteriori calcoli. Abbiamo, dunque, tutti gli strumenti per iniziare a configurare il nostro filtro di Kalman per il sensor fusion.



Per chi non avesse ancora familiarità con il filtro di Kalman, rimando al mio tutorial specifico, dove il metodo viene illustrato in dettaglio.

Il filtro di Kalman per il sensor fusion

In rete si trovano numerosi esempi di applicazione del filtro di Kalman per la fusione dei dati di accelerometro e giroscopio, con lo scopo di compensare rumore e bias. Non ripeteremo quindi un'analisi di questo tipo. Vogliamo invece porci una domanda meno comune ma molto interessante: ha senso fondere più sensori identici per aumentare l'accuratezza di una misura? La risposta è sì. Utilizzare più sensori “uguali”, ad esempio diversi giroscopi MEMS dello stesso modello, e fonderne i dati con un Kalman può portare a vantaggi concreti. Se i loro errori non sono perfettamente correlati, il rumore complessivo si riduce all'incirca come $\sqrt{\frac{1}{N}}$.

Quando abbiamo a disposizione un accelerometro e più giroscopi dello stesso tipo, l'idea di base è quella di fondere tutte le misure in un unico stimatore che riesca a “tirare fuori” il meglio da ciascun sensore, sfruttando i punti di forza di ognuno e compensandone i difetti. L'accelerometro, infatti, fornisce un'informazione assoluta sull'inclinazione (ad esempio il pitch), ma questa misura è rumorosa e disturbata dalle accelerazioni dinamiche che non dipendono dalla gravità. I giroscopi, al contrario, danno una misura molto stabile della velocità angolare, ma soffrono di drift dovuto ai bias che si accumulano nel tempo. Aggiungendo più giroscopi identici, e trattando i loro errori come almeno in parte indipendenti, è possibile ridurre il rumore complessivo e stimare in maniera più affidabile anche i bias individuali.

Per modellare il sistema con il filtro di Kalman dobbiamo definire uno stato esteso, che non contiene solo l'angolo di interesse (il pitch), ma anche la velocità angolare e i bias di ciascun giroscopio. In questo modo, il filtro può contemporaneamente stimare sia l'evoluzione dell'angolo che l'errore sistematico di ogni sensore. È un approccio molto didattico, ma utile perché mostra chiaramente come il Kalman riesca a scomporre l'informazione “nascosta” dentro misure rumorose.

Il funzionamento del filtro si articola in due momenti fondamentali: predizione e correzione.

Fase di predizione: In questa fase il filtro “immagina” come dovrebbe evolvere lo stato del sistema nel tempo, basandosi sul modello matematico e sulle misure dei giroscopi. In pratica, l’angolo al passo successivo viene previsto integrando la velocità angolare stimata, mentre i bias dei giroscopi vengono ipotizzati costanti (con piccole variazioni casuali ammesse dal modello). Questa previsione, inevitabilmente, porta con sé incertezza, che viene rappresentata e aggiornata nella matrice di covarianza dell’errore.

Fase di correzione: A questo punto entra in gioco l’accelerometro, che fornisce una misura indipendente dell’angolo. Questa informazione viene usata come riferimento assoluto per correggere la previsione dei giroscopi. Il filtro calcola un opportuno “guadagno di Kalman”, che determina quanto fiducia dare alla misura rispetto al modello. Se l’accelerometro è molto rumoroso, il guadagno sarà basso e il filtro si affiderà di più alla previsione dei giroscopi. Se invece la previsione dei giroscopi è incerta (ad esempio perché i bias non sono ancora ben stimati), allora il guadagno darà più peso all’accelerometro. In questo modo, le due fonti vengono fuse dinamicamente in maniera ottimale.

La logica predizione–correzione, iterata a ogni passo temporale, consente di ottenere una stima dell’angolo fluida, stabile e robusta.

L’aspetto più interessante è che, grazie alla presenza di più giroscopi, il filtro può stimare e separare il contributo del bias di ciascun sensore. Questo significa che non solo il rumore casuale viene mediato e attenuato, ma anche le **derive sistematiche vengono progressivamente compensate**. Alla fine, l’angolo stimato risulta molto più stabile e accurato di quello che si otterebbe con un singolo sensore.

Dunque, i dati di progetto sono:

- Vogliamo stimare l’**angolo θ** e l’**angolar rate ω** su un asse (pitch).
- Disponiamo di **N giroscopi identici** montati co-allineati (stesso asse), ciascuno con **bias** proprio b_i e **rumore bianco** v_i .

- Un **accelerometro** fornisce un'osservazione “assoluta” di θ (da ax, ay, az), valida quando prevale la gravità.
- Campionamento costante: Δt .

Iniziamo a definire il vettore di stato:

$$x_k = \begin{bmatrix} \theta_k \\ \omega_k \\ b_{1,k} \\ b_{2,k} \\ \dots \\ b_{N,k} \end{bmatrix}$$

Dove θ_k [rad], $\theta_k \left[\frac{\text{rad}}{\text{s}} \right]$, $b_{i,k} \left[\frac{\text{rad}}{\text{s}} \right]$

Notiamo che lo stato del sistema di misure all'istante k è composto dai valori di angolo e velocità angolare e dai bias di ogni giroscopio.

Vediamo ora di modellare la dinamica del sistema facendo attenzione a non confondere la velocità angolare ω con in rumore w .

Supponiamo di essere in condizioni stazionarie (constant-rate) e in presenza di bias random-walk (variazione lenta e non prevedibile).

$$\begin{aligned} \theta_{k+1} &= \theta_k + \Delta t \omega_k + w_{\theta,k} \\ \omega_{k+1} &= \omega_k + w_{\omega,k} \\ b_{i,k+1} &= b_{i,k} + w_{b_{i,k}} \quad i = 1, \dots, N \end{aligned}$$

Osserviamo che:

1. La dinamica dell'angolo reale è semplicissima: $\theta_{k+1} = \theta_k + \Delta t \omega_k + w_{\theta,k}$. In questa relazione ω_k è la velocità angolare **reale**, non quella misurata dal giroscopio, mentre $w_{\theta,k}$ è un rumore di processo. Il concetto di bias entrerà nella definizione di $z_{i,k}^g$. I giroscopi non misurano la ω_k vera, ma una versione affetta da bias e rumore. Quindi il bias non influisce sulla dinamica di θ , ma modifica le misure del giroscopio.
2. Quando scriviamo $b_{i,k+1} = b_{i,k} + w_{b_{i,k}}$ **non** stiamo dicendo che il bias sia **costante nel tempo**. Stiamo assumendo che:
 - **in media** il bias rimanga uguale,

- ma che possa variare lentamente e in modo imprevedibile,
- questa variazione è modellata dal rumore $w_{b_i,k}$.

Il bias di un giroscopio MEMS **non è un valore fisso** ma varia con **temperatura, tempo di utilizzo**, o può avere **drift casuale**.

Per il filtro di Kalman, il modo più semplice di rappresentare questa incertezza è il **modello “random walk”**: $b_{i,k+1} = b_{i,k} + \text{rumore piccolo}$

Quindi, il bias è "quasi costante", ma il filtro si lascia un margine per adattarsi a lente variazioni.

3. Quando scriviamo $\omega_{k+1} = \omega_k + w_{\omega,k}$ stiamo dicendo che in assenza di accelerazioni angolari esterne, la velocità angolare rimane costante tra due istanti consecutivi. Tuttavia, siccome il mondo reale non è mai perfetto, introduciamo un termine $w_{\omega,k}$ (rumore di processo) che rappresenta variazioni imprevedibili della velocità angolare (accelerazioni angolari reali che non stiamo modellando esplicitamente).

Fatte queste importanti precisazioni, torniamo alle tre equazioni di stato e scriviamole in forma matriciale.

$$\mathbf{x}_{k+1} = F \mathbf{x}_k + \mathbf{w}_k$$

Con

$$F = \begin{bmatrix} 1 & \Delta t & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}, \quad \mathbf{w}_k \sim N(0, Q)$$

$$Q = \text{diag}(q_\theta, q_\omega, q_{b_1}, q_{b_1} \dots q_{b_N})$$

Facciamo alcune considerazioni su Q che, come sappiamo, modella quanto incerto è il modello dinamico $\mathbf{x}_{k+1} = F \mathbf{x}_k$.

- q_θ : in genere è piccolo e assorbe modellazioni imperfette.
- q_ω : indica quanta variabilità di ω ci aspettiamo tra due istanti (noi la stiamo ritenendo costante, ma in realtà non è così).

- q_{b_i} : “quanto” ci aspettiamo che il bias del giroscopio “i” vari nel tempo (collega la Bias Instability al filtro).

Gli elementi di Q dicono “quanto permetto allo stato di cambiare indipendentemente dal modello deterministico”, ovvero quantifica l’errore che può avvenire tra un passo e il successivo: accelerazioni angolari impreviste, variazioni lente del bias, modellazioni approssimate, ecc.

È spesso il parametro più delicato da scegliere ma si può stimare mediante un semplice metodo sperimentale. Vediamo come:

- Registriamo i dati del giroscopio per un lungo periodo mentre il sistema è fermo o in condizioni operative note.
- Stimiamo i vari ω_k e calcoliamo $\Delta\omega_k = \omega_{k-1} - \omega_k$. La varianza di $\Delta\omega$ fornisce q_ω .
- Per l’angolo θ valgono le stesse considerazioni. La varianza degli errori di predizione $\theta_{k+1} - (\theta_k + \omega_k \Delta t)$ fornisce q_θ

Passiamo ora alle misure:

Giroscopi (misure di velocità angolare ω_i con bias $b_{i,k}$)

$$z_{i,k}^g = \omega_k + b_{i,k} + v_{i,k}^g \quad i = 1, \dots, N$$

Accelerometro (angolo di inclinazione)

$$z_k^a = \theta_k + b_{i,k} + v_k^a$$

Mettiamo tutto in un unico vettore z_k :

$$z_k = \begin{bmatrix} z_{1,k}^g \\ z_{2,k}^g \\ \vdots \\ z_{N,k}^g \\ \vdots \\ z_k^a \end{bmatrix}$$

Ricordando le definizioni di z_k^a e $z_{i,k}^g$ viste sopra, possiamo scrivere:

$$z_k = H x_k + v_k$$

Con matrice H di dimensioni $(N+1) \times (2+N)$ e definita come:

$$H = \begin{bmatrix} 0 & 1 & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 1 & 0 & 0 & \dots & 1 & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix}, \quad v_k \sim N(0, R)$$

Il rumore di misura lo modelliamo con:

$$R = \text{diag}(\sigma_{g,1}^2, \dots, \sigma_{g,N}^2, \sigma_a^2)$$

Vediamo come stimare R , ovvero la matrice del rumore di misura. R contiene le varianze (non le deviazioni standard) delle misure, *nello stesso passo di campionamento Δt* . È possibile stimare le varianze direttamente dalle misure ottenendo valori affidabili. Il metodo è il seguente:

1. Si mantiene l'IMU fermo su un piano stabile.
2. Si campionano i giroscopi e l'accelerometro per un tempo T (es. 60s) a frequenza $1/\Delta t$.
3. Prendiamo le serie dei valori ottenuti.
 - per i giroscopi si calcola direttamente la varianza di ogni misura di velocità angolare.
 - per l'accelerometro, calcoliamo gli angoli $\theta_{\text{acc}}[n]$ a partire dalle accelerazioni sugli assi x,y,z e poi la $\text{Var}(\theta_{\text{acc}})$
4. Popoliamo R con le varianze ottenute.

Ora abbiamo tutti gli strumenti per avviare le iterazioni del filtro di Kalman.

Nelle immagini seguenti trovate i risultati delle simulazioni effettuate per evidenziare l'effetto della ridondanza dei giroscopi. Come si vede, già con cinque giroscopi i segnali stimati di angolo e velocità angolare risultano notevolmente puliti.

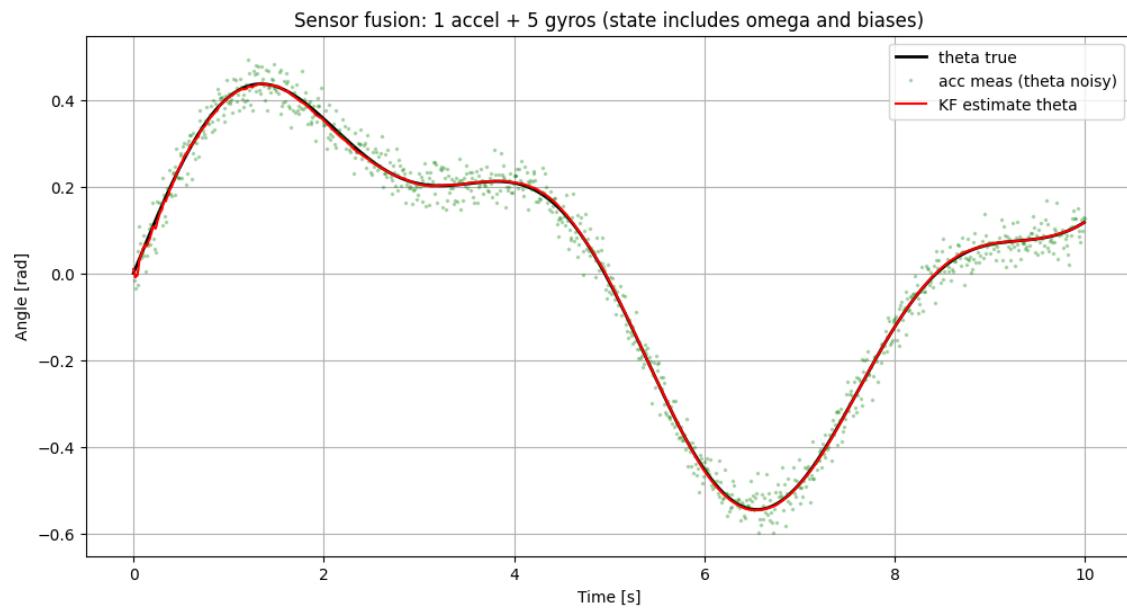


Figura 8: Stima dell'angolo con un accelerometro e 5 giroscopi

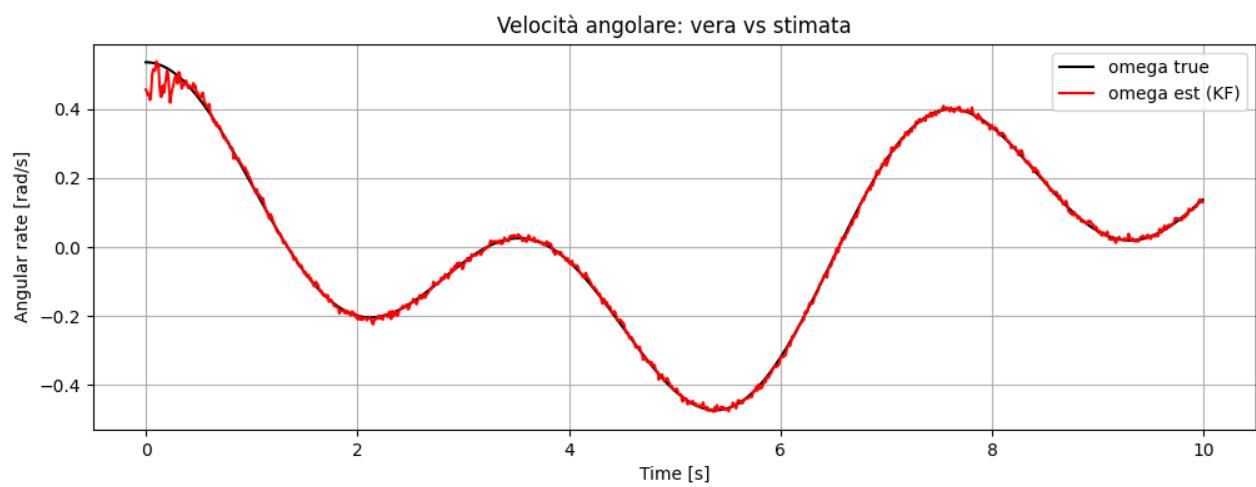


Figura 9: Velocità angolare stimata con 5 giroscopi

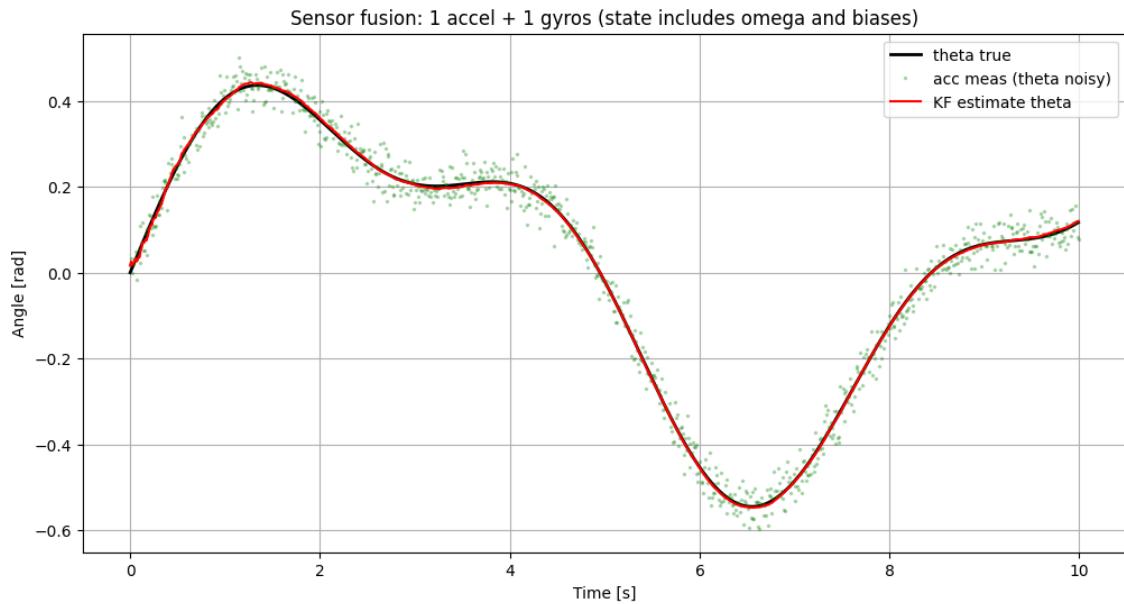


Figura 10: Stima dell'angolo con un accelerometro e un giroscopio

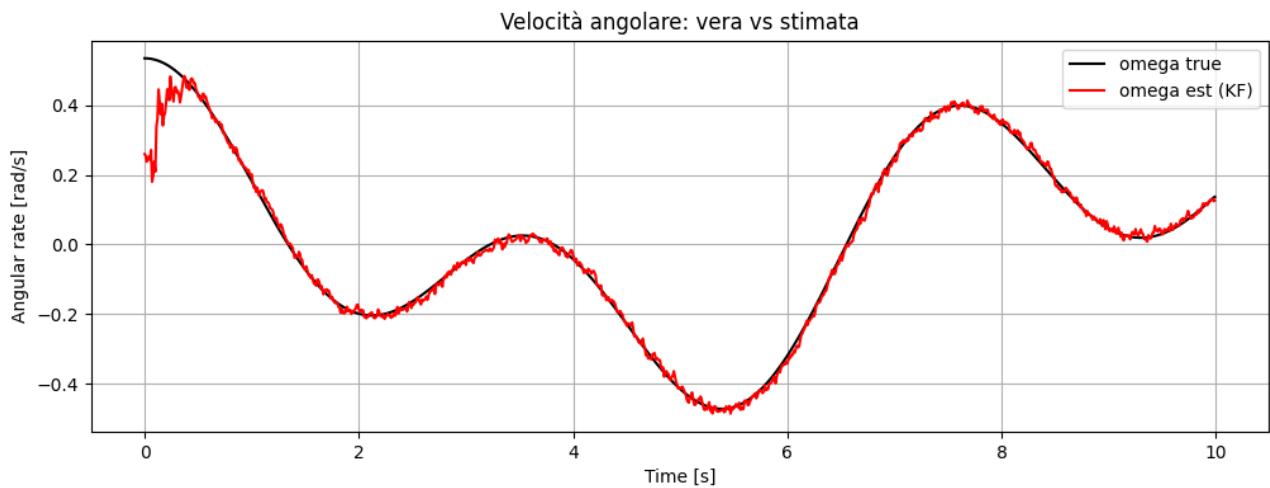


Figura 11: Velocità angolare stimata con un solo giroscopio

In appendice trovate il sorgente Python per provare a simulare il processo e valutare gli effetti di Q ed R sulle stime.

5. Appendice

```
import numpy as np
import matplotlib.pyplot as plt

# -----
# Configurazione (parametrizzabile)
# -----
np.random.seed(0)
dt = 0.01                      # passo campionamento [s]
T = 10.0                         # durata [s]
t = np.arange(0, T, dt)
steps = len(t)

num_gyros = 5                    # <-- numero di giroscopi (modificabile)

# Sensibilità / rumore (scelte esemplificative)
gyro_noise_std = 0.02            # [rad/s] std del rumore del singolo giroscopio
gyro_bias_true = np.linspace(-0.05, 0.05, num_gyros)  # bias veri in rad/s per ogni gyro
acc_angle_noise_std = 0.03       # [rad] std del rumore dell'accelerometro (angolo)

# -----
# Segnale "reale" (simulazione)
# -----
# Angolo vero (rad): combinazione di sinusoidi per avere dinamica
true_theta = 0.4 * np.sin(0.7 * t) + 0.15 * np.sin(1.7 * t)
# Velocità angolare vera (rad/s)
true_omega = np.gradient(true_theta, dt)

# -----
# Simulazione misure
# -----
# Giroscopi: omega + bias_i + rumore
gyro_meas = np.zeros((num_gyros, steps))
for i in range(num_gyros):
    gyro_meas[i, :] = true_omega + gyro_bias_true[i] + np.random.normal(0, gyro_noise_std, size=steps)

# Accelerometro: fornisce stima di theta (quando non ci sono accelerazioni significative)
acc_meas = true_theta + np.random.normal(0, acc_angle_noise_std, size=steps)

# -----
# Costruzione matrici del filtro di Kalman (lineare)
# Stato x = [theta, omega, b1, b2, ..., bn]^T
# -----
n_state = 2 + num_gyros
n_meas = 1 + num_gyros  # 1 misura accel (theta), N misure gyro (omega + b_i)

# Matrice di transizione F
F = np.eye(n_state)
F[0, 1] = dt      # theta_k = theta_{k-1} + dt * omega_{k-1}
# omega e bias si appoggiano su modello random-walk (omega assume identity)

# Non usiamo B: i giroscopi sono parte della misura, non input esterno al modello

# Matrice di osservazione H:
# prima riga: accelerometro misura theta -> [1, 0, 0, ..., 0]
# successive righe: gyro i misura omega + b_i -> [0, 1, e_i]
H = np.zeros((n_meas, n_state))
H[0, 0] = 1.0
for i in range(num_gyros):
```

```

H[1 + i, 1] = 1.0      # omega
H[1 + i, 2 + i] = 1.0  # bias i

# Rumore di processo Q (scelta ragionevole iniziale)
# Mettiamo rumore su omega (accelerazioni angolari non modellate) e su bias (random walk)
q_theta = 1e-8
q_omega = 1e-4
q_bias = 1e-8
Q = np.zeros((n_state, n_state))
Q[0, 0] = q_theta
Q[1, 1] = q_omega
for i in range(num_gyros):
    Q[2 + i, 2 + i] = q_bias

# Rumore di misura R: diag(var(acc), var(gyro1), ..., var(gyroN))
R = np.zeros((n_meas, n_meas))
R[0, 0] = acc_angle_noise_std**2
for i in range(num_gyros):
    R[1 + i, 1 + i] = gyro_noise_std**2

# Stampiamo le matrici
np.set_printoptions(precision=4, suppress=True)
print("F (state transition):\n", F)
print("\nH (observation):\n", H)
print("\nQ (process covariance):\n", Q)
print("\nR (measurement covariance):\n", R)

# -----
# Inizializzazione Kalman
# -----
x = np.zeros((n_state, 1))      # stato iniziale: zeros (puoi inizializzare theta da acc_meas[0])
x[0, 0] = acc_meas[0]           # inizializziamo theta dall'accelerometro
P = np.eye(n_state) * 0.1        # covarianza iniziale (sovrastimata)

# Storage risultati
x_hist = np.zeros((n_state, steps))
innovations = np.zeros((n_meas, steps))

# -----
# Ciclo Kalman (predizione + aggiornamento con vettore misure)
# -----
for k in range(steps):
    # Predizione
    x = F @ x
    P = F @ P @ F.T + Q

    # Costruzione vettore misure z (accel, gyro1..gyroN)
    z = np.zeros((n_meas, 1))
    z[0, 0] = acc_meas[k]
    for i in range(num_gyros):
        z[1 + i, 0] = gyro_meas[i, k]

    # Innovazione
    y = z - (H @ x)
    S = H @ P @ H.T + R
    K = P @ H.T @ np.linalg.inv(S)

    # Aggiornamento
    x = x + K @ y
    P = (np.eye(n_state) - K @ H) @ P

    # Salva
    x_hist[:, k] = x.flatten()

```

```

innovations[:, k] = y.flatten()

# -----
# Risultati e grafici
# -----
theta_est = x_hist[0, :]
omega_est = x_hist[1, :]
bias_est = x_hist[2:, :] # shape (num_gyros, steps)

# Calcolo RMSE dell'angolo
rmse_theta = np.sqrt(np.mean((theta_est - true_theta)**2))
print(f"\nRMSE theta (KF fused {num_gyros} gyros): {rmse_theta:.6f} rad = {rmse_theta*180/np.pi:.4f} deg")

# Plot angolo vero, misura accel, stima KF
plt.figure(figsize=(12, 6))
plt.plot(t, true_theta, 'k-', label='theta true', linewidth=2)
plt.plot(t, acc_meas, 'g.', alpha=0.25, label='acc meas (theta noisy)', markersize=3)
plt.plot(t, theta_est, 'r-', label='KF estimate theta', linewidth=1.5)
plt.xlabel('Time [s]')
plt.ylabel('Angle [rad]')
plt.title(f'Sensor fusion: 1 accel + {num_gyros} gyros (state includes omega and biases)')
plt.legend()
plt.grid(True)

# Plot stime dei bias (confronto col bias vero)
plt.figure(figsize=(12, 6))
for i in range(num_gyros):
    plt.plot(t, bias_est[i, :], label=f'est bias {i+1}')
    plt.hlines(gyro_bias_true[i], 0, T, colors='k', linestyles='dashed', alpha=0.4)
plt.xlabel('Time [s]')
plt.ylabel('Bias estimate [rad/s]')
plt.title('Stime dei bias dei singoli giroscopi (linee tratteggiate = bias veri)')
plt.legend()
plt.grid(True)

# Plot omega vero vs stimato
plt.figure(figsize=(12, 4))
plt.plot(t, true_omega, 'k-', label='omega true')
plt.plot(t, omega_est, 'r-', label='omega est (KF)')
plt.xlabel('Time [s]')
plt.ylabel('Angular rate [rad/s]')
plt.title('Velocità angolare: vera vs stimata')
plt.legend()
plt.grid(True)

plt.show()

```