

An approach to robust INS/UWB positioning

Sistemi di Guida e Navigazione

Francesco Iori

Liu J, Pu J, Sun L, He Z. An Approach to Robust INS/UWB Integrated Positioning for Autonomous Indoor Mobile Robots. *Sensors (Basel)*. 2019;19(4):950. Published 2019 Feb 23. doi:10.3390/s19040950

Introduzione

Problemi

- Rumore tempo-variante e presenza di outliers nelle misure dei sistemi UWB, specialmente durante navigazione indoor o in ambienti affollati (e.g. effetti multi-path o non-line-of-sight)
- Sistemi di navigazione che stimano adattivamente la matrice di covarianza del rumore si basano generalmente sull'assunzione che cambi lentamente e/o che il numero di outliers sia sufficientemente ridotto da avere influenza limitata
- Complicazioni dovute a non-linearità del sistema (spesso derivanti dalla dinamica dell'orientazione)

Metodi utilizzati

- Modellazione dello stato dell'errore
- Modifica del peso dell'innovazione della matrice di covarianza con filtro fuzzy (aggiunta della parte fuzzy a un metodo IAE, Innovation Adaptive Estimation)
- Stima con sliding window della covarianza per utilizzo nella detezione di outliers tramite criterio di ortogonalità dell'innovazione

Formulazione del problema

Stato

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{p}_k, : \text{position} \\ \mathbf{v}_k, : \text{velocity} \\ \mathbf{q}_k, : \text{orientation quaternion} \\ \mathbf{a}_k, : \text{acceleration bias} \\ \boldsymbol{\omega}_k, : \text{angular velocity bias} \end{bmatrix}$$

Decomposizione dello stato come somma di
stato nominale e stato dell'errore

$$\mathbf{x} = \hat{\mathbf{x}} \oplus \delta\mathbf{x},$$
$$\delta\mathbf{x} = [\delta\mathbf{p}^T, \delta\mathbf{v}^T, \delta\boldsymbol{\theta}^T, \delta\mathbf{a}_b^T, \delta\boldsymbol{\omega}_b^T]^T$$

Evoluzione

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k)$$

Cfr. Trawny, N., & Roumeliotis, S.I. (2005). *Indirect Kalman Filter for 3 D Attitude Estimation* per riassunto su algebra dei quaternioni e utilizzo di filtro di Kalman indiretto sullo stato dell'errore

Modello cinematico

Assumendo che l'errore di orientazione sia sufficientemente piccolo si può **approssimare il quaternione per piccole rotazioni**: in questo modo l'equazione di update dello stato dell'errore diventa lineare in $\delta\theta$.

$$\delta\mathbf{q} = \begin{bmatrix} 1 \\ \delta\theta/2 \end{bmatrix}$$

C matrice di rotazione dal sistema inerziale al sistema globale, ricavabile dal quaternione di orientazione (stimato):

$$\mathbf{C} = (2q_0^2 - 1)\mathbf{I}_3 - 2q_0[\mathbf{q}_v]_{\times} + 2\mathbf{q}_v\mathbf{q}_v^T$$

Equazioni di stato:

$$\dot{\hat{\mathbf{x}}} = \begin{bmatrix} \dot{\hat{\mathbf{p}}} \\ \dot{\hat{\mathbf{v}}} \\ \dot{\hat{\mathbf{q}}} \\ \dot{\hat{\mathbf{a}}}_b \\ \dot{\hat{\boldsymbol{\omega}}}_b \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{v}} \\ \hat{\mathbf{C}}(\mathbf{a}_m - \hat{\mathbf{a}}_b) + \mathbf{g} \\ \frac{1}{2}\hat{\mathbf{q}} \otimes (\boldsymbol{\omega}_m - \hat{\boldsymbol{\omega}}_b) \\ 0 \\ 0 \end{bmatrix}$$

$$\delta\dot{\mathbf{x}} = \begin{bmatrix} \delta\dot{\mathbf{p}} \\ \delta\dot{\mathbf{v}} \\ \delta\dot{\boldsymbol{\theta}} \\ \delta\dot{\mathbf{a}}_b \\ \delta\dot{\boldsymbol{\omega}}_b \end{bmatrix} = \begin{bmatrix} \delta\mathbf{v} \\ -\hat{\mathbf{C}}[\mathbf{a}_m - \hat{\mathbf{a}}_b]_{\times}\delta\boldsymbol{\theta} - \hat{\mathbf{C}}\delta\mathbf{a}_b - \hat{\mathbf{C}}\mathbf{a}_n \\ -[\boldsymbol{\omega}_m - \hat{\boldsymbol{\omega}}_b]_{\times}\delta\boldsymbol{\theta} - \boldsymbol{\omega}_w - \boldsymbol{\omega}_n \\ \mathbf{a}_w \\ \boldsymbol{\omega}_w \end{bmatrix}$$

[35] Zhen, W., Zeng, S., & Soberer, S. (2017). Robust localization and localizability estimation with a rotating laser scanner. 2017 IEEE International Conference on Robotics and Automation (ICRA), 6240-6245.

Misurazione UWB

Il sistema UWB misura il tempo di arrivo dei segnali provenienti dalle varie anchors, fornendo una stima della distanza del tag da ogni anchor.

Soluzione ai minimi quadrati per ottenere una misura di posizione.
Denominando:

- \mathbf{p}_n - posizione delle N anchors UWB
- $\mathbf{p}_{m,k}$ - posizione reale del tag UWB al tempo k
- $\mathbf{p}_{t,k}$ - posizione misurata del tag UWB al tempo k
- $d_{n,k}$ - distanza misurata dall'anchor n al tempo k
- $\bar{d}_{n,k}$ - distanza reale dall'anchor n al tempo k
- $\hat{\mathbf{p}}_{m,k}$ - soluzione che minimizza l'errore tra dist. reale e misurata

$$\hat{\mathbf{p}}_{m,k} = \arg \min_{\mathbf{p}_{m,k}} \sum_{n=1}^N (d_{n,k} - \bar{d}_{n,k})^2$$

$$\bar{d}_{n,k} = \|\mathbf{p}_n - \mathbf{p}_{t,k}\|_2 = \sqrt{(x_n - x_{t,k})^2 + (y_n - y_{t,k})^2 + (z_n - z_{t,k})^2}$$

$$d_{n,k} = \|\mathbf{p}_n - \mathbf{p}_{m,k}\|_2 = \sqrt{(x_n - x_{m,k})^2 + (y_n - y_{m,k})^2 + (z_n - z_{m,k})^2}$$

$$\left\{ \begin{array}{l} d_{1,k}^2 = (x_1 - x_{m,k})^2 + (y_1 - y_{m,k})^2 + (z_1 - z_{m,k})^2 \\ d_{2,k}^2 = (x_2 - x_{m,k})^2 + (y_2 - y_{m,k})^2 + (z_2 - z_{m,k})^2 \\ \vdots \\ d_{N,k}^2 = (x_N - x_{m,k})^2 + (y_N - y_{m,k})^2 + (z_N - z_{m,k})^2. \end{array} \right.$$

Misurazione UWB

Soluzione ai minimi quadrati
usando la **pseudo-inversa di G**

$$2\mathbf{G}\mathbf{p}_{m,k} = \mathbf{b}_k$$

$$\mathbf{G} = \begin{pmatrix} x_1 - x_2 & y_1 - y_2 & z_1 - z_2 \\ \vdots & \vdots & \vdots \\ x_1 - x_N & y_1 - y_N & z_1 - z_N \end{pmatrix}$$

$$\mathbf{b}_k = \begin{pmatrix} d_{2,k}^2 - d_{1,k}^2 + x_1^2 - x_2^2 + y_1^2 - y_2^2 + z_1^2 - z_2^2 \\ \vdots \\ d_{N,k}^2 - d_{1,k}^2 + x_1^2 - x_N^2 + y_1^2 - y_N^2 + z_1^2 - z_N^2 \end{pmatrix}$$

$$\mathbf{p}_{m,k} = \frac{1}{2}(\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{b}_k$$

$$\mathbf{v}_{m,k} = \frac{\Delta \mathbf{p}_k}{\Delta t_k} = \frac{\mathbf{p}_{m,k} - \mathbf{p}_{m,k-1}}{t_k - t_{k-1}}$$

Da cui si ha l'**equazione di misura**:

$$\mathbf{z}_k = \begin{bmatrix} \delta \mathbf{p}_k \\ \delta \mathbf{v}_k \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{m,k} - \mathbf{p}_{INS,k} \\ \mathbf{v}_{m,k} - \mathbf{v}_{INS,k} \end{bmatrix} = \mathbf{H} \delta \mathbf{x}_k + \mathbf{v}_k$$

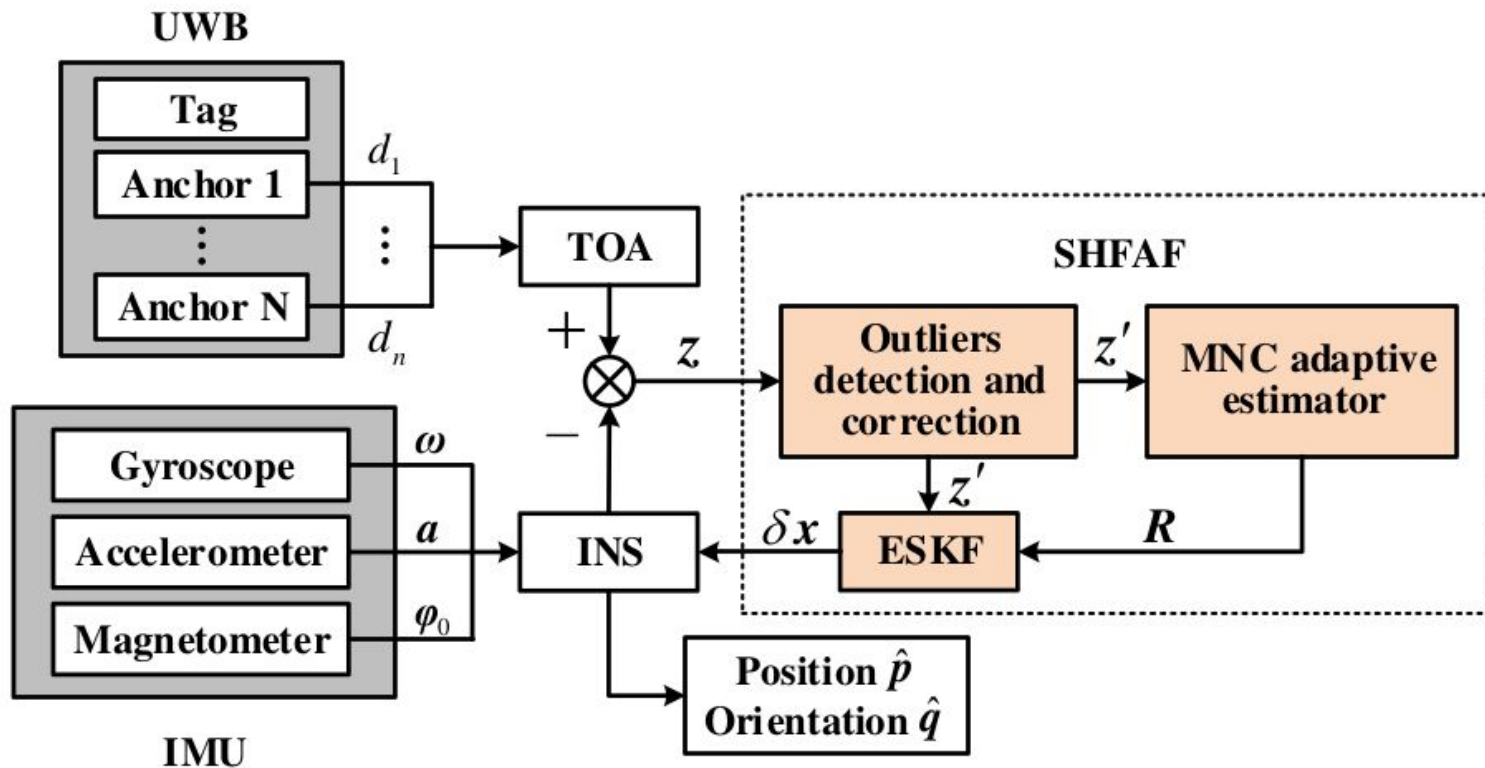
$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_3 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{I}_3 & 0 & 0 & 0 \end{bmatrix}$$

Matrice di
misura

$$\mathbf{v} \sim \mathcal{N}\{0, \mathbf{R}\}$$

R matrice di
covarianza del rumore

Overview



Predizione

Stato nominale

Equazione non-lineare,
integrazione con Eulero

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1})$$

$$\dot{\hat{\mathbf{x}}} = \begin{bmatrix} \dot{\hat{\mathbf{p}}} \\ \dot{\hat{\mathbf{v}}} \\ \dot{\hat{\mathbf{q}}} \\ \dot{\hat{\mathbf{a}}}_b \\ \dot{\hat{\boldsymbol{\omega}}}_b \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{v}} \\ \hat{\mathbf{C}}(\mathbf{a}_m - \hat{\mathbf{a}}_b) + \mathbf{g} \\ \frac{1}{2}\hat{\mathbf{q}} \otimes (\boldsymbol{\omega}_m - \hat{\boldsymbol{\omega}}_b) \\ 0 \\ 0 \end{bmatrix}$$

Stato dell'errore

$$\delta \mathbf{x}_{k|k-1} = \mathbf{F}_{\delta \mathbf{x}, k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}) \cdot \delta \mathbf{x}_{k-1}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{\delta \mathbf{x}, k-1} \mathbf{P}_{k-1} \mathbf{F}_{\delta \mathbf{x}, k-1}^T + \boldsymbol{\Gamma}_n \mathbf{Q}_n \boldsymbol{\Gamma}_n^T$$

La prima equazione risulta sempre zero, $\delta \mathbf{x}$ è re-inizializzato a zero alla fine di ogni step.

La matrice \mathbf{F} è ricavata per discretizzazione di:

$$\delta \dot{\mathbf{x}} = \begin{bmatrix} \delta \dot{\mathbf{p}} \\ \delta \dot{\mathbf{v}} \\ \delta \dot{\boldsymbol{\theta}} \\ \delta \dot{\mathbf{a}}_b \\ \delta \dot{\boldsymbol{\omega}}_b \end{bmatrix} = \begin{bmatrix} \delta \mathbf{v} \\ -\hat{\mathbf{C}}[\mathbf{a}_m - \hat{\mathbf{a}}_b]_{\times} \delta \boldsymbol{\theta} - \hat{\mathbf{C}} \delta \mathbf{a}_b - \hat{\mathbf{C}} \mathbf{a}_n \\ -[\boldsymbol{\omega}_m - \hat{\boldsymbol{\omega}}_b]_{\times} \delta \boldsymbol{\theta} - \boldsymbol{\omega}_w - \boldsymbol{\omega}_n \\ \mathbf{a}_w \\ \boldsymbol{\omega}_w \end{bmatrix}$$

Predizione

$$\mathbf{F}_{\delta \mathbf{x}, k-1} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3 \Delta t & 0 & 0 & 0 \\ 0 & \mathbf{I}_3 & -\hat{\mathbf{C}}_{k-1} [\mathbf{a}_{m,k-1} - \hat{\mathbf{a}}_{b,k-1}]_{\times} \Delta t & -\hat{\mathbf{C}}_{k-1} \Delta t & 0 \\ 0 & 0 & \hat{\mathbf{C}}_{k-1} \{(\boldsymbol{\omega}_{m,k-1} - \hat{\boldsymbol{\omega}}_{b,k-1}) \Delta t\} & 0 & -\mathbf{I}_3 \Delta t \\ 0 & 0 & 0 & \mathbf{I}_3 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I}_3 \end{bmatrix}$$

Nota:

$$\mathbf{Q}_n = \begin{bmatrix} (\sigma_{a_n} \Delta t)^2 \mathbf{I}_3 & 0 & 0 & 0 \\ 0 & (\sigma_{\omega_n} \Delta t)^2 \mathbf{I}_3 & 0 & 0 \\ 0 & 0 & (\sigma_{a_\omega} \Delta t)^2 \mathbf{I}_3 & 0 \\ 0 & 0 & 0 & (\sigma_{\omega_\omega} \Delta t)^2 \mathbf{I}_3 \end{bmatrix}$$

$$\mathbf{\Gamma}_n = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \mathbf{I}_3 & 0 & 0 & 0 \\ 0 & \mathbf{I}_3 & 0 & 0 \\ 0 & 0 & \mathbf{I}_3 & 0 \\ 0 & 0 & 0 & \mathbf{I}_3 \end{bmatrix}$$

[35] Zhen, W., Zeng, S., & Soberer, S. (2017). *Robust localization and localizability estimation with a rotating laser scanner*. 2017 IEEE International Conference on Robotics and Automation (ICRA), 6240-6245.

$\hat{\mathbf{C}}\{\boldsymbol{\gamma}\}$: matrice di rotazione ottenuta considerando $\boldsymbol{\gamma}$ come vettore di rotazione in notazione asse-angolo

$\hat{\mathbf{C}}$: matrice di rotazione dal sistema inerziale al sistema globale, ottenuta dal quaternionione di orientazione (stimato)

$[\cdot]_{\times}$: matrice skew-simmetrica

Stima della covarianza

Filtro di Kalman tradizionale

$$\varepsilon_k = \mathbf{z}_k - \mathbf{H}\delta\mathbf{x}_{k|k-1}$$

$$\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}_k$$



Caso ideale, filtro convergente

$$\varepsilon_k \varepsilon_k^T = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}_k$$

$$\hat{\mathbf{R}}_k = \varepsilon_k \varepsilon_k^T - \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T$$

- ε_k - Innovazione
- \mathbf{S}_k - Covarianza (teorica) dell'innovazione
- $\hat{\mathbf{R}}_k$ - Stima teorica della covarianza

Equazione di update della covarianza

$$\mathbf{R}_k = (1 - s_k^\alpha d_k) \mathbf{R}_{k-1} + s_k^\alpha d_k \overbrace{(\varepsilon_k \varepsilon_k^T - \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T)}^{\hat{\mathbf{R}}_k}$$

Stima della covarianza

Equazione di update della covarianza

$$\mathbf{R}_k = (1 - s_k^\alpha d_k) \mathbf{R}_{k-1} + s_k^\alpha d_k \overbrace{(\varepsilon_k \varepsilon_k^T - \mathbf{H} \mathbf{P}_{k|k-1} \mathbf{H}^T)}^{\hat{\mathbf{R}}_k}$$

Modified Innovation Contribution Weight (MICW)

Nell'ICW tradizionale ($\lambda=1$) il peso dell'innovazione diventa molto piccolo quando si stabilizza, **riducendo l'informazione data dall'innovazione**. Questo porta a ritardare la stima della MNC quando si ha rumore tempo-variante.

$$d_k = (\lambda - b) / (\lambda - b^{k+1})$$

$$\lambda \geq 1$$

L'introduzione di $\lambda > 1$ dà un **peso più alto dell'innovazione** una volta stabilizzato

b : forgetting factor (compreso solitamente tra [0.95,0.99])

Stima della covarianza

Equazione di update della covarianza

$$\mathbf{R}_k = (1 - s_k^\alpha d_k) \mathbf{R}_{k-1} + s_k^\alpha d_k \overbrace{(\varepsilon_k \varepsilon_k^T - \mathbf{H} \mathbf{P}_{k|k-1} \mathbf{H}^T)}^{\hat{\mathbf{R}}_k}$$

Fuzzy system

1. Inizialmente il MICW viene lasciato convergere per farlo tendere a stabilizzarsi ($\mathbf{s}_k=1$), poi il **sistema fuzzy** diventa attivo e aiuta ad avere una **migliore adattività**.
2. Il parametro α **compie uno scaling di \mathbf{s}_k** . Un valore più alto corrisponde a una convergenza più veloce al valore reale di \mathbf{R} , ma può portare a stime instabili.

$$\hat{\mathbf{S}}_k = E(\varepsilon_k \varepsilon_k^T)$$

$$r_k = \left| \frac{\text{Tr}(\hat{\mathbf{S}}_k)}{\text{Tr}(\mathbf{S}_k)} - 1 \right|$$

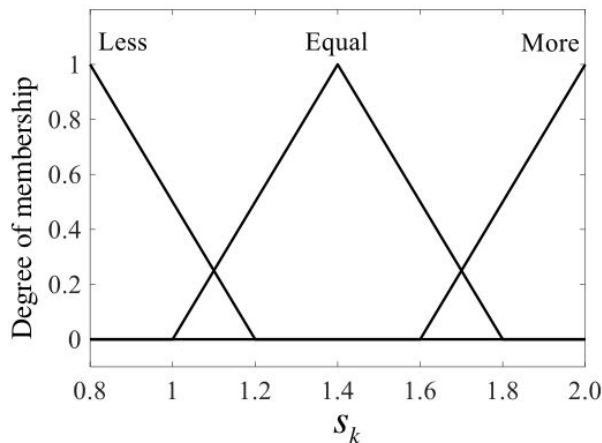
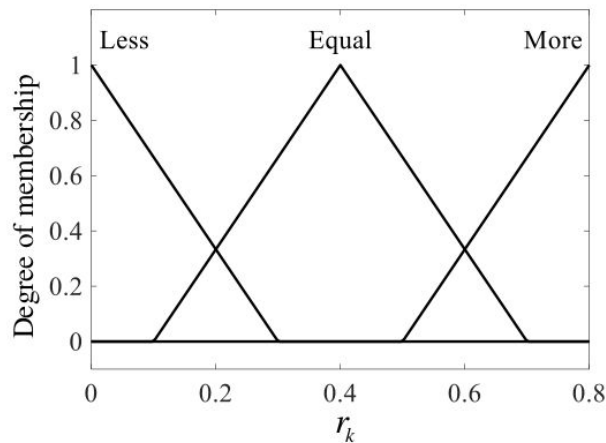
$$s_k = \begin{cases} 1 & , k \leq k_s \\ \text{fuzzy}(r_k) & , k > k_s \end{cases}$$

Stima della covarianza

Equazione di update della covarianza

$$\mathbf{R}_k = (1 - s_k^\alpha d_k) \mathbf{R}_{k-1} + s_k^\alpha d_k \underbrace{(\varepsilon_k \varepsilon_k^T - \mathbf{H} \mathbf{P}_{k|k-1} \mathbf{H}^T)}_{\hat{\mathbf{R}}_k}$$

Fuzzy rules e membership functions



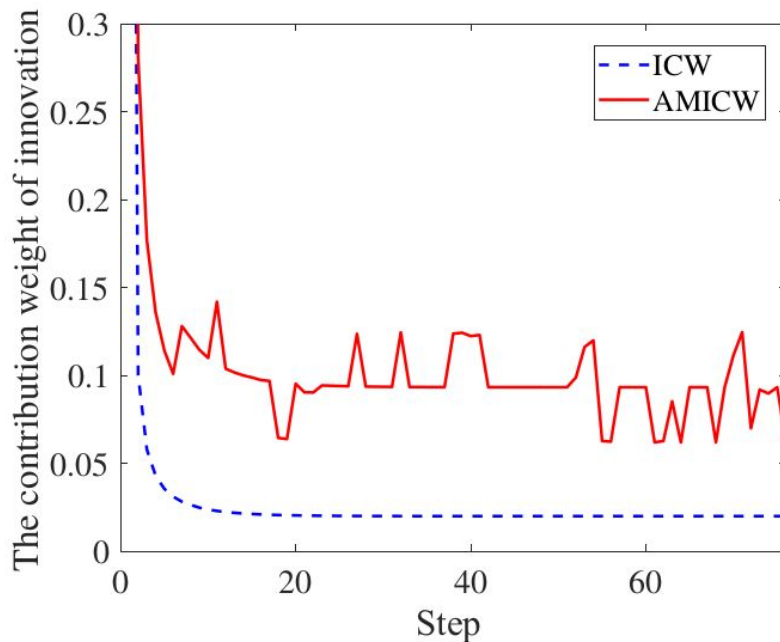
If $r_k \in \mathbf{Equal}$, then $s_k \in \mathbf{Equal}$

If $r_k \in \mathbf{More}$, then $s_k \in \mathbf{More}$

If $r_k \in \mathbf{Less}$, then $s_k \in \mathbf{Less}$

Stima della covarianza

AMICW (adaptive modified innovation contribution weight)



ICW

$\lambda=1$ e filtro fuzzy assente
($s_k = 1$)

Outliers detection

Le misure **UWB** possono essere **disturbate da effetti multi-path** (ricezione multipla in seguito a rimbalzi) e **NLOS** (non-line-of-sight, ostacoli lungo il percorso di propagazione).

Per migliorare la stima e ridurre la possibile divergenza del filtro viene utilizzato un metodo di **detezione e correzione** basato sul criterio di ortogonalità. Nel caso di assenza di outliers:

$$E(\mathbf{z}_k \mathbf{z}_k^T) = E(\boldsymbol{\varepsilon}_k \boldsymbol{\varepsilon}_k^T) + E(\mathbf{z}_{k|k-1} \mathbf{z}_{k|k-1}^T)$$

Cfr. [31] Wang, L., & Li, S. (2018). Enhanced Multi-sensor Data Fusion Methodology based on Multiple Model Estimation for Integrated Navigation System. *International Journal of Control, Automation and Systems*, 16, 295-305.

Per verificare la presenza di un outlier viene utilizzato il **rapporto tra la stima e l'aspettativa teorica** di $\mathbf{z}_k \mathbf{z}_k^T$.

Chiamando $G_{i,k}$ e $D_{i,k}$ gli elementi sulla diagonale di $\mathbf{E}(\mathbf{z}_k \mathbf{z}_k^T)$ e $\mathbf{E}(\mathbf{z}_{k|k-1} \mathbf{z}_{k|k-1}^T)$

$$M_{i,k} = G_{i,k} / D_{i,k}$$

$$fr_{i,k} = \begin{cases} 1 & M_{i,k} \leq \xi_i \\ 1 / \sqrt{M_{i,k}} & M_{i,k} > \xi_i \end{cases}$$

con ξ_i una soglia prestabilita di sensibilità.

Outliers detection

L'eventuale outlier rilevato, è **corretto con**:

$$\mathbf{z}'_k = \mathbf{fr}_k \times \mathbf{z}_k.$$

$$\mathbf{fr}_k = \text{diag}[fr_{1,k}, fr_{2,k}, \dots, fr_{\text{rank}(\mathbf{R}),k}]$$

dove maggiore è la deviazione tra misura e predizione ($M_{i,k}$) minore è $fr_{i,k}$

La matrice $\mathbf{E}(\mathbf{z}_{k|k-1}\mathbf{z}_{k|k-1}^T)$ è calcolata dall'equazione di update di \mathbf{z} come:

$$\mathbf{D}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}_k + \mathbf{H}\delta\mathbf{x}_{k|k-1}\delta\mathbf{x}_{k|k-1}^T\mathbf{H}^T$$

La matrice $\mathbf{E}(\varepsilon_k\varepsilon_k^T)$ è stimata tramite **sliding window** considerando gli ultimi N valori dell'innovazione ε :

$$\hat{\mathbf{S}}_k = E(\varepsilon_k\varepsilon_k^T) = \sum_{j=k-l+1}^k \sigma_j \varepsilon_j \varepsilon_j^T$$

$$\sigma_j = a^{k-j}(1-a)/(1-a^l)$$

l = lunghezza della finestra (numero di samples)

a = fading coefficient $\in [0.95-0.99]$

Update della misura (Kalman)

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k|k-1} \mathbf{H}^T + \mathbf{R}_k)^{-1}$$

$$\delta \mathbf{x}_k = \mathbf{K}_k \varepsilon_k$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_{k|k-1}$$



$$\begin{bmatrix} \hat{\mathbf{p}}_k \\ \hat{\mathbf{v}}_k \\ \hat{\mathbf{q}}_k \\ \hat{\mathbf{a}}_{b,k} \\ \hat{\boldsymbol{\omega}}_{b,k} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{p}}_{k-1} + \delta \mathbf{p}_k \\ \hat{\mathbf{v}}_{k-1} + \delta \mathbf{v}_k \\ \hat{\mathbf{q}}_{k-1} \otimes \mathbf{q}\{\delta \boldsymbol{\theta}_k\} \\ \hat{\mathbf{a}}_{b,k-1} + \delta \mathbf{a}_{b,k} \\ \hat{\boldsymbol{\omega}}_{b,k-1} + \delta \boldsymbol{\omega}_{b,k} \end{bmatrix}$$

Alla fine di ogni step $\delta \mathbf{x}$ viene resettato a zero

Overview

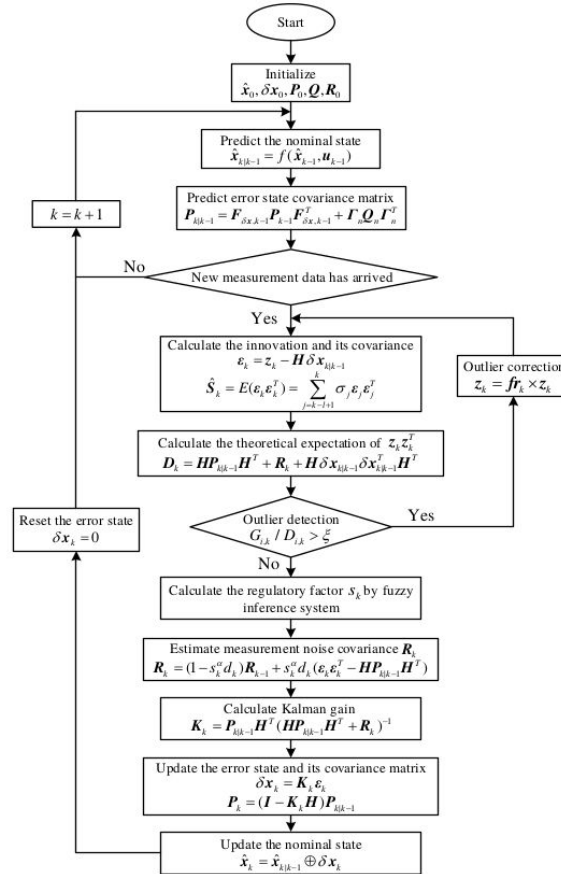


Figure 5. The flowchart of Sage-Husa fuzzy adaptive error state Kalman filter.

Simulazione

Per analizzare il comportamento del filtro sono stati utilizzati i seguenti parametri (riportati anche nel paper originale):

IMU

$$\begin{aligned} f_{IMU} &= 100Hz && \text{- frequenza di update} \\ \sigma_{IMU}^2 &= 0.06^2 && \text{- rumore bianco di misura} \end{aligned}$$

UWB

posizione anchors = $((-5, -1), (1, -1), (1, 5), (-5, 5))[m]$

$$\begin{aligned} \sigma_{UWB,r} &= 0.1^2 && \text{- rumore bianco misura} \\ u_{UWB,int} &\in [-0.2, 0.2]m && \text{- interferenza di misura (dist. uniforme)} \\ u_{out} &\in [-2, 2]m && \text{- aggiunta outliers (dist. uniforme)} \end{aligned}$$

La traiettoria di riferimento è stata generata simulando un veicolo di tipo uniciclo, sul quale è “montato” il sistema di navigazione al quale arrivano solo le misure dei sensori.

Il filtro è stato **inizializzato con una posizione errata** aggiungendo 0.5 m alla posizione reale, sia sull'asse x che sull'asse y.

Il valore iniziale della matrice di covarianza del rumore di misura è stata inizializzato a 4 volte il valore reale.

Risultati

I **cerchi verdi** rappresentano gli **outliers** inseriti nella misura.

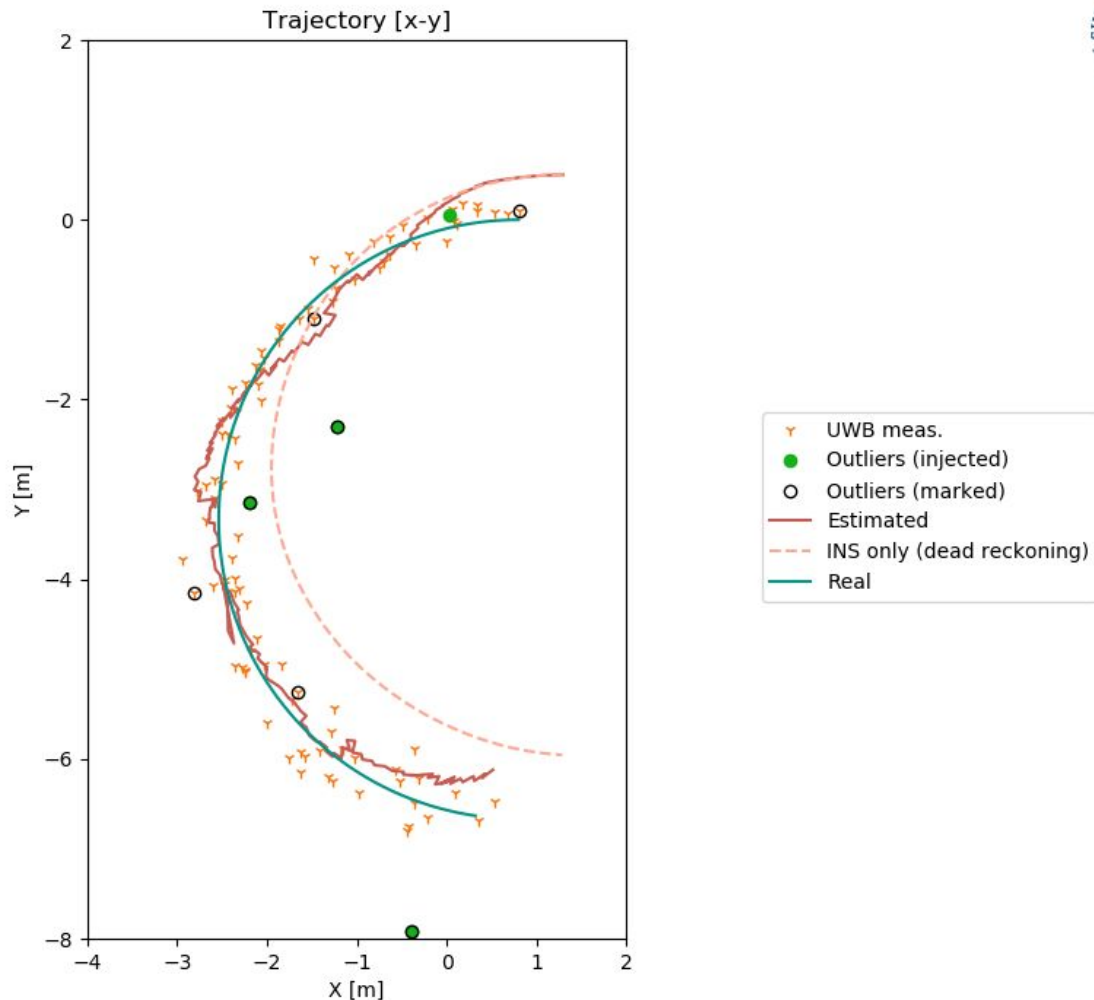
I **cerchi neri** marcano le misurazioni UWB per cui è avvenuta la **detezione** come outlier.

Poiché il rumore UWB è dato dalla somma di rumore bianco a una distribuzione uniforme, alcune misure “corrette” possono comunque essere indicate come outliers.

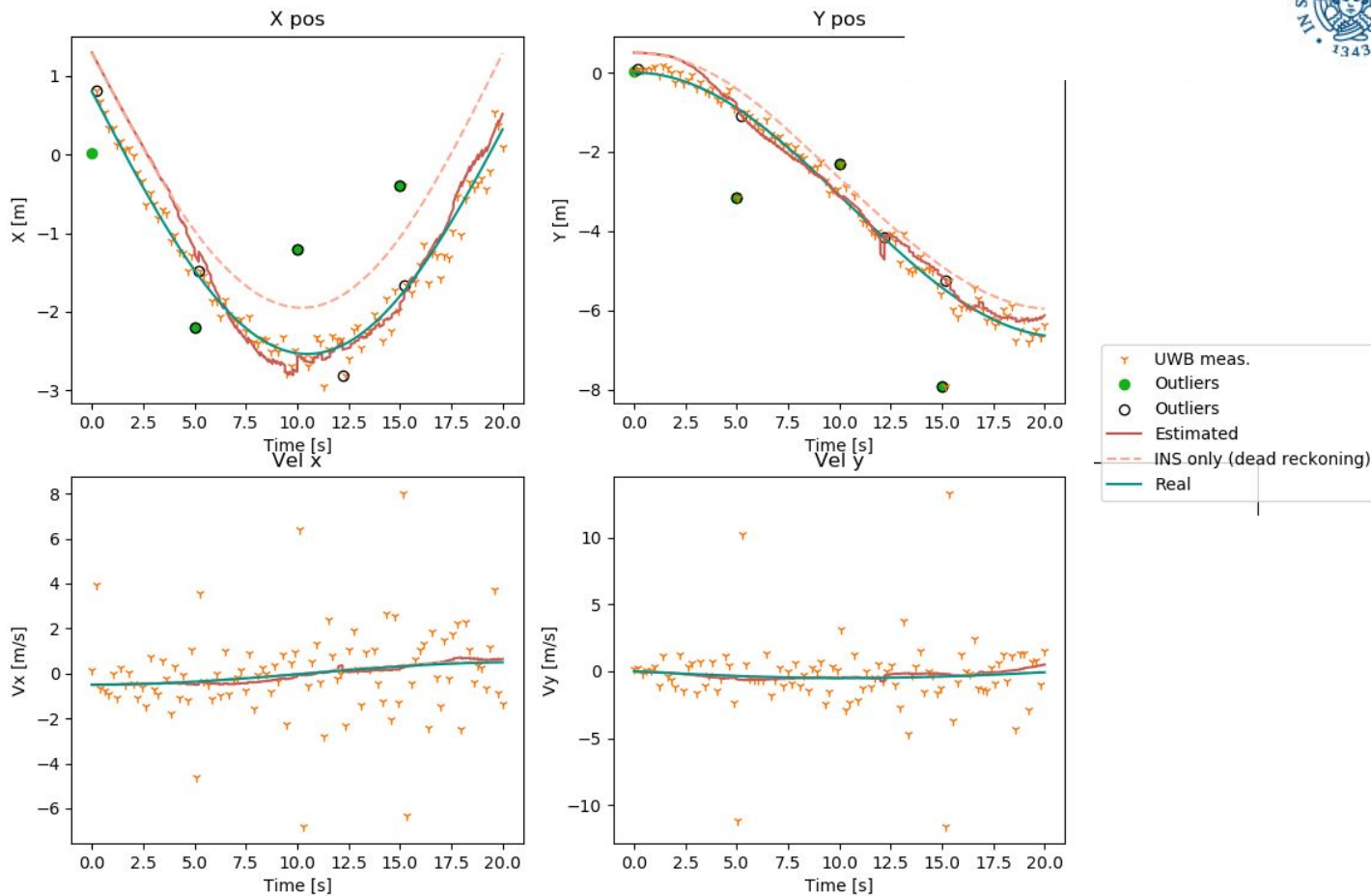
Parametri

$a=0.9784$, $l=12$, $\lambda=1.0119$,

$b=0.9976$, $\alpha=0.48$, $\xi=4.5$



Risultati

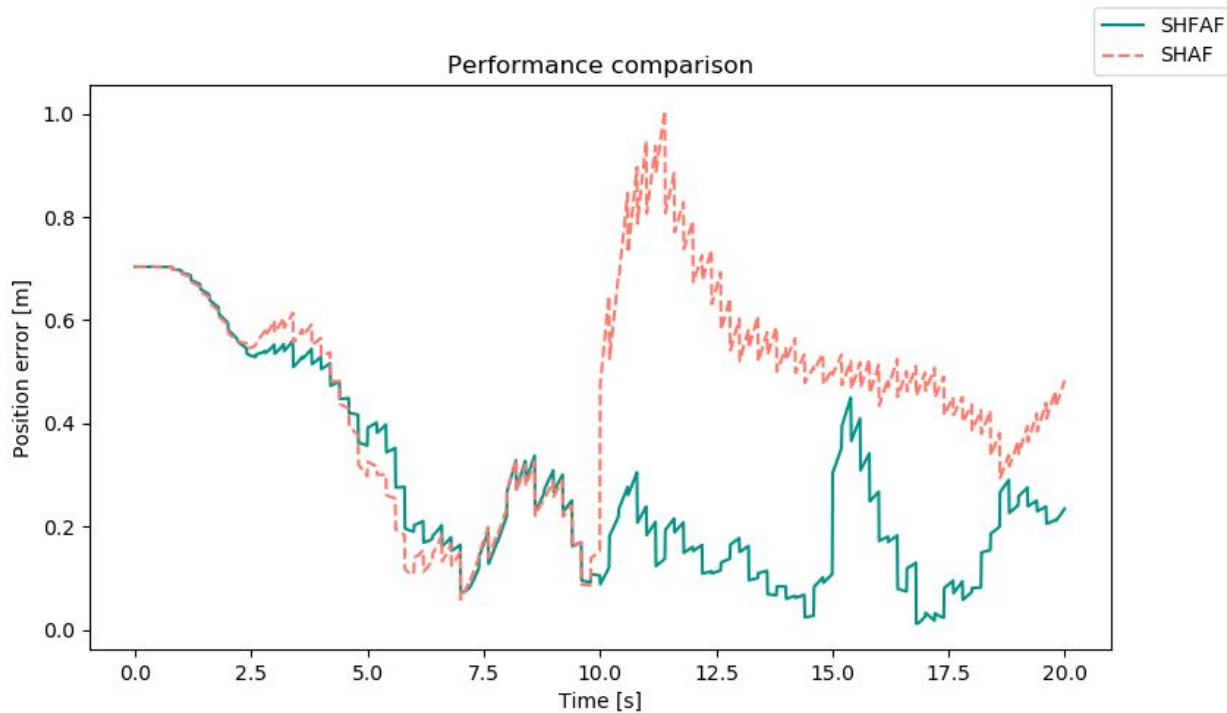


Risultati

Comparazione della norma dell'errore di posizione con (SHFAF) e senza (SHAF) fuzzy inference system ($s_k=1$).

In un caso tipico l'utilizzo della parte fuzzy migliora l'adattività del sistema: nel caso illustrato a $t=10$ s e $t=15$ s avviene l'ingresso di due outliers particolarmente grandi; la presenza della parte fuzzy migliora la rapidità di risposta.

Questo può essere utile nel caso di un numero elevato di outliers, per ristabilizzare il filtro prima dell'ingresso di un nuovo outlier di misura.



Bayesian optimization

Durante l'implementazione e il testing del filtro si è riscontrato il problema di fornire un opportuno **tuning dei parametri** che influenzano il funzionamento del filtro.

Si è scelto di implementare una procedura di ottimizzazione Bayesiana, in quanto essa si presta bene a questo tipo di problema dove **non è disponibile una relazione tra parametri e risultato** ottenuto durante la navigazione.

Adatta a funzioni di tipo “black-box”, tollera rumore nella valutazione della funzione da minimizzare, appropriata a funzioni dispendiose da valutare.

Il funzionamento si basa su una **funzione di acquisizione** che, basandosi sui risultati ottenuti dalle precedenti combinazioni di parametri (valutati su una funzione di costo), sceglie la successiva combinazione da testare, con un **tradeoff tra exploitation ed exploration** (comune agli algoritmi euristici di ottimizzazione).

La scelta risultante è una combinazione tra massimizzazione del miglioramento previsto ottenibile e massimizzazione della probabilità di miglioramento.

Bayesian optimization

Funzione da minimizzare: **norma L_2** del segnale di errore.

Le prove hanno un **risultato variabile**, dovuto alla casualità introdotta dal **rumore di misura**.

Per questo motivo si è scelto di considerare come valore il **risultato medio su 7 prove**; in questo modo si ha inoltre una stima della varianza, del risultato ottenuto con ogni set di parametri, che può essere sfruttato dalla funzione di acquisizione per un risultato più accurato.

$$f_{min} = \|e(k)\|_2 = \left(\sum_{k=0}^T e_k^T e_k \Delta t \right)^{\frac{1}{2}}$$

Bayesian optimization

Implementata con la libreria di Python *HyperOpt*.

Dopo ogni valutazione i risultati delle combinazioni di parametri provate vengono salvate in un file *.csv, in modo da poter essere recuperati agevolmente.

È stato inoltre implementato il salvataggio della variabile contenente le prove effettuate (oltre che dei risultati), per poter riprendere la stessa ottimizzazione senza perdere l'informazione sui parametri provati in precedenza.

Questo si è dimostrato utile per poter procedere con un numero basso di prove effettuate in ogni sessione, perchè nel caso di combinazioni di parametri “estreme” il sistema diverge numericamente, ed è possibile che la simulazione venga interrotta per valori inf o NaN nelle matrici del filtro.

Nel caso presentato si è dimostrata uno strumento efficace.

Conclusioni

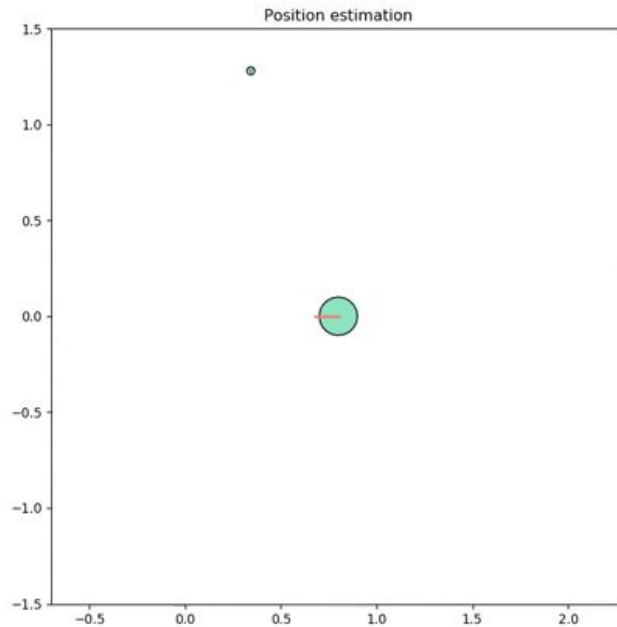
Il sistema implementato si è rivelato efficace nella determinazione della posizione di navigazione; la presenza di una **parte fuzzy dimostra nella maggior parte dei casi maggiore responsività all'ingresso di outlier**.

Nonostante questo sono comunque ottenibili prove in cui il filtro a cui viene rimossa la parte fuzzy performa meglio; globalmente (dai risultati di varie prove) sembra che la presenza della parte fuzzy rappresenti comunque un vantaggio .

Un vantaggio decisivo è stato inoltre rappresentato dall'utilizzo della tecnica di ottimizzazione proposta per il tuning dei parametri del filtro, che ha concesso di esplorare combinazioni di parametri diverse.

Si riporta in particolare l'**influenza di L** , lunghezza della sliding window utilizzata **per la stima di S** ; il suo valore rappresenta un trade-off tra velocità di convergenza della stima e stabilità del valore stimato.

Animazione



Il codice scritto per il progetto è disponibile a: https://github.com/fraiori0/SGN_project

References

- *Liu J, Pu J, Sun L, He Z. An Approach to Robust INS/UWB Integrated Positioning for Autonomous Indoor Mobile Robots. Sensors (Basel). 2019;19(4):950. Published 2019 Feb 23. doi:10.3390/s19040950*
- Trawny, N., & Roumeliotis, S.I. (2005). Indirect Kalman Filter for 3D Attitude Estimation
- Zhen, W., Zeng, S., & Soberer, S. (2017). Robust localization and localizability estimation with a rotating laser scanner. 2017 IEEE International Conference on Robotics and Automation (ICRA), 6240-6245.
- Wang, L., & Li, S. (2018). Enhanced Multi-sensor Data Fusion Methodology based on Multiple Model Estimation for Integrated Navigation System. International Journal of Control, Automation and Systems, 16, 295-305.

Il codice scritto per il progetto è disponibile a: https://github.com/fraiori0/SGN_project