



**UNIVERSITÀ DI PISA**

**DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE**

Corso di laurea magistrale in Ingegneria Robotica e  
dell'Automazione

**Relazione progetto – TracciaD**

**Modulo: Identificazione dei Sistemi Incerti (ISI)**

**Dario Lauretta – Francesco Borrello**

## Modello del sistema e validazione

Il sistema è rappresentato da un modello di “Segway” (Fig. 1) di massa  $M$  il cui moto avviene nel piano  $xz$ . Il sistema può essere schematizzato come un’asta di lunghezza  $L$  (rispetto al CoM – Centro di Massa) incernierata al centro di una ruota di raggio  $r$  e massa  $m$  che rotola senza strisciare lungo la direzione  $x$ . La ruota è dotata di un motore che applica su di essa una coppia  $\tau$ . L’angolo  $\varphi$  rappresenta la posizione angolare dell’asta rispetto alla verticale, il cui comportamento è assimilabile a quello di un pendolo inverso. Le grandezze che descrivono il sistema ed i relativi valori nominali sono indicati in tabella 1 [1].

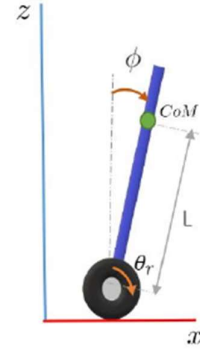


Fig. 1

Parametro	Valore nominale
Raggio ruota – $r$ [m]	0.2
Massa ruota – $m$ [kg]	2
Massa corpo – $M$ [kg]	0.5
Lunghezza asta – $L$ [m]	0.4
Gravità – $g$ [m/s <sup>2</sup> ]	9.81

Tabella 1

Le equazioni della dinamica vincolata sono le seguenti:

$$\begin{bmatrix} \tilde{M}_{\dot{v}} & \tilde{M}_{v\varphi} \\ \tilde{M}_{\varphi v} & \tilde{M}_{\ddot{\varphi}} \end{bmatrix} \begin{bmatrix} \dot{v} \\ \ddot{\varphi} \end{bmatrix} + \begin{bmatrix} \tilde{c}_v \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{g}_{\ddot{\varphi}} \end{bmatrix} = \begin{bmatrix} 1/r & 1/r \\ -1 & -1 \end{bmatrix} \begin{bmatrix} \tau \\ \tau \end{bmatrix}$$

Con:

$$\tilde{M}_{\dot{v}} = 2 \left( \frac{3}{2} m + \frac{1}{2} M \right), \quad \tilde{M}_{\ddot{\varphi}} = 2 \left( \frac{1}{2} m r^2 + \frac{2}{3} M L^2 \right), \quad \tilde{M}_{v\varphi} = \tilde{M}_{\varphi v} = -m r + M L \cos \varphi,$$

$$\tilde{c}_v = -M L \sin \varphi (\dot{\varphi}^2), \quad \tilde{g}_{\ddot{\varphi}} = -M g L \sin \varphi$$

L’implementazione in ambiente Simulink prevede la descrizione del sistema tramite il Subsystem Segway\_model che comprende la function eom\_segway contenente le equazioni della dinamica riportate sopra. Gli integratori utilizzati per implementare quest’ultime sono tutti inizializzati con le rispettive condizioni iniziali del sistema:  $x_0$ ,  $\varphi_0$ ,  $\dot{x}_0$ ,  $\dot{\varphi}_0$ .

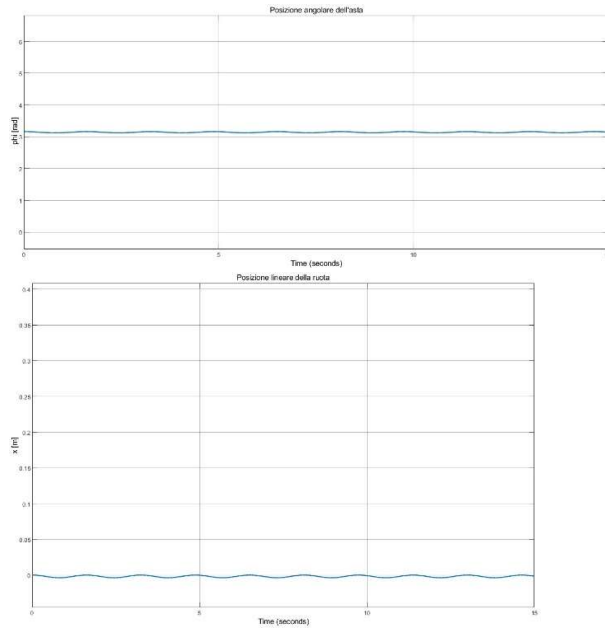
Le uscite nominali estratte dal Subsystem sono: la posizione lungo  $x$  [m] del centro della ruota, la posizione angolare  $\varphi$  [rad] dell’asta e le rispettive velocità  $\dot{x}$  [m/s] e  $\dot{\varphi}$  [rad/s] che vanno in ingresso al Subsystem Sensors\_models, il quale contiene le function dei 3 sensori utilizzati in seguito specificate.

L’unico ingresso deterministico è rappresentato dalla coppia  $\tau$  [Nm].

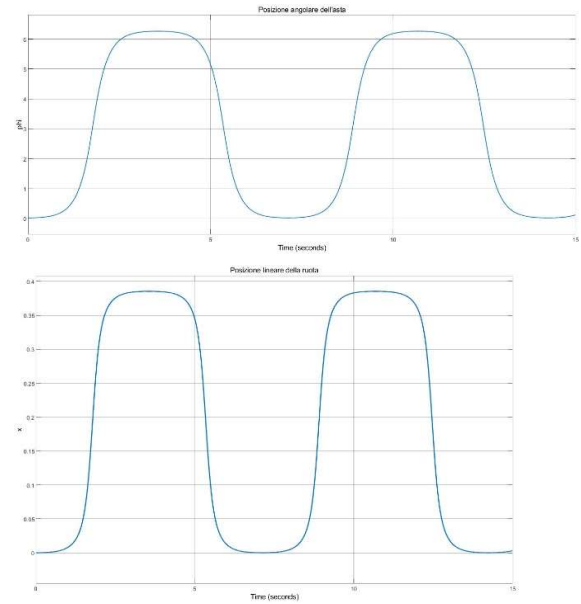
Sia i parametri del modello che le condizioni iniziali sono specificate nello script parameters.m.

Il modello è stato sottoposto ad una serie di combinazioni tra ingressi e condizioni iniziali al fine di testarne il comportamento per un tempo di simulazione pari a 15 secondi. I risultati dei valori nominali ottenuti sono mostrati in Tabella 2.

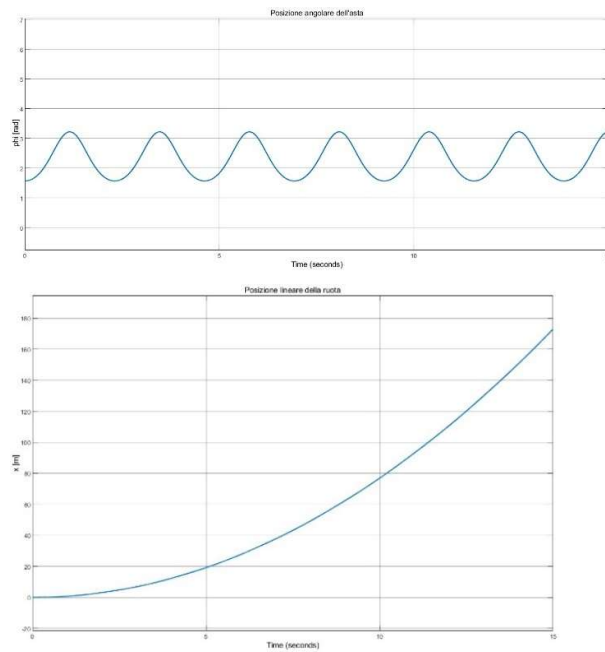
- a) Condizioni iniziali:  $x_0 = 0$ ,  $\varphi_0 = \pi + 0.02$ ,  $\dot{x}_0 = 0$ ,  $\dot{\varphi}_0 = 0$ .  
Ingresso:  $\tau = 0$ .



- b) Condizioni iniziali:  $x_0 = 0$ ,  $\varphi_0 = 0.02$ ,  $\dot{x}_0 = 0$ ,  $\dot{\varphi}_0 = 0$ .  
Ingresso:  $\tau = 0$ .



- c) Condizioni iniziali:  $x_0 = 0$ ,  $\varphi_0 = \pi$ ,  $\dot{x}_0 = 0$ ,  $\dot{\varphi}_0 = 0$ .  
Ingresso:  $\tau = 1$ .



- d) Condizioni iniziali:  $x_0 = 0$ ,  $\varphi_0 = \pi$ ,  $\dot{x}_0 = 0$ ,  $\dot{\varphi}_0 = 0$ .  
Ingresso:  $\tau = \sin(ft)$ ,  $f = 1\text{Hz}$ .

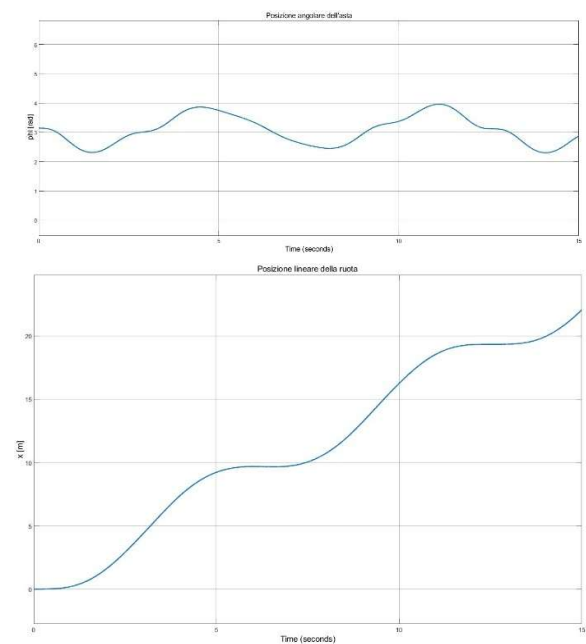


Tabella 2.

Le uscite relative alla posizione angolare  $\varphi$  dell'asta ed allo spostamento lineare  $x$  della ruota, riportate nei grafici, mostrano che il comportamento del modello matematico risulta coerente con il sistema fisico, in particolare:

- Se l'ingresso  $\tau$  è nullo, partendo da una condizione iniziale  $\varphi_0$  intorno al punto di equilibrio stabile ( $\varphi_0 = \pi$ ), l'asta si muove con piccole oscillazioni intorno ad esso mentre la ruota oscilla anch'essa intorno alla condizione iniziale. [Tabella 2 – a)];
- Se l'ingresso  $\tau$  è nullo, partendo da una condizione iniziale  $\varphi_0$  intorno al punto di equilibrio instabile ( $\varphi_0 = 0$ ), le oscillazioni dell'asta risultano molto ampie, compiendo quasi un giro completo per poi tornare indietro. La ruota anche in questo caso oscilla con media positiva poiché la perturbazione iniziale su  $\varphi_0$  è positiva. [Tabella 2 – b)];
- Applicando un ingresso  $\tau$  costante, con condizioni iniziali analoghe al caso a), l'asta oscilla a causa delle forze inerziali bilanciate dall'avanzare della ruota, con  $\varphi$  che inizialmente aumenta coerentemente con il verso di avanzamento della ruota. La posizione della ruota cresce in modo parabolico. [Tabella 2 – c)];
- Infine, applicando un ingresso  $\tau$  sinusoidale l'andamento della posizione angolare risulta periodica mentre la posizione lineare cresce anch'essa con andamento sinusoidale.

## Modello dei sensori e validazione

Le grandezze assegnate da misurare con opportuni sensori sono (Fig. 2):

- distanza  $D$
- velocità angolare  $\omega$  della ruota
- angolo  $\alpha_1$

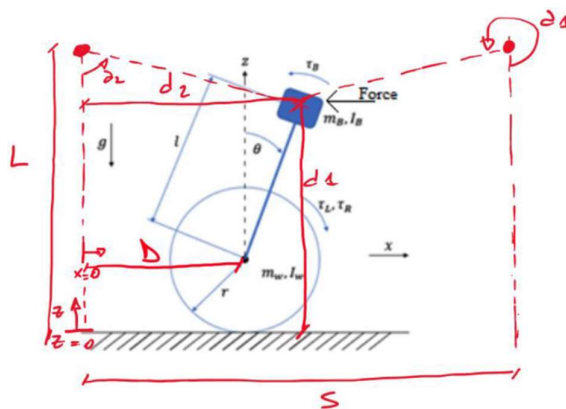


Fig. 2

I sensori scelti e le relative caratteristiche sono elencati di seguito:

- **Encoder [2]**

Grandezza misurata: velocità angolare  $\omega$

Modello matematico:  $\omega = \frac{\dot{x}}{r} + \sigma_{\omega} randn(1,1)$

Dispositivo commerciale: EPC MA3 Miniature Absolute Magnetic Shaft Encoder

Parametri:

$f \approx 683,733$  frequenza [Hz]

$T_{\omega} = 0.001$  periodo di campionamento [s]

$\sigma_{\omega} = 0.0025$  deviazione standard [rad/s]

Posizionamento: asse della ruota

- **Sensore laser [3]**

Grandezza misurata: distanza  $D$

Modello matematico:  $D = x + \sigma_D randn(1,1)$

Dispositivo commerciale: Hokuyo UTM-30LX-EW LiDAR

Parametri:

$f = 40$  frequenza [Hz]

$T_D = 0.025$  periodo di campionamento [s]

$\sigma_D = 0.0005$  deviazione standard [m]

Posizionamento:

$x = 0$

$z = r$

- **Camera [4]**

Grandezza misurata: angolo  $a_1$

Modello matematico:  $a_1 = atan2(x - S_{a_1} + L \sin \varphi, L_{a_1} - (L \cos \varphi + r)) + \sigma_{a_1} randn(1,1)$

Dispositivo commerciale: Intel RealSense D435

Parametri:

frame rate: 30 FPS, then  $T = 1/30$

$T_{a_1} = 0.033$  tempo di campionamento [s]

$\sigma_{a_1} = 0.07$  deviazione standard [rad]

Posizionamento:

$x = S_{a_1} = 1.3$  [m]

$z = L_{a_1} = 0.6$  [m]

Si noti che il sensore più veloce (tempo di campionamento più piccolo) è l'encoder, il quale gode anche di una maggiore precisione (deviazione standard più piccola).

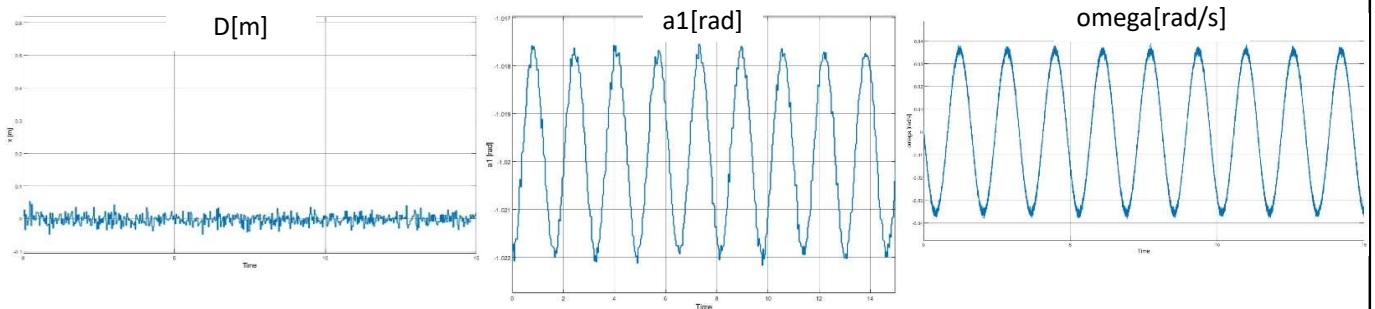
Ogni sensore è implementato in ambiente Simulink tramite un blocco function contenente il relativo modello matematico ed impostato a funzionare, tramite le proprietà del blocco, secondo lo specifico tempo di campionamento.

I sensori sono tutti e tre inclusi nel Subsystem Sensors\_models che prende in ingresso le grandezze nominali del sistema quali  $\varphi$ ,  $x$ ,  $\dot{x}$ .

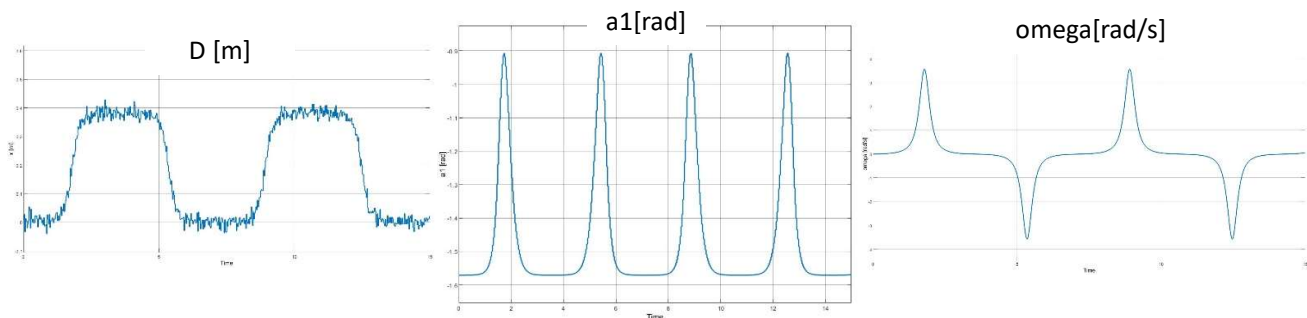
Tutti i parametri dei sensori sono specificati nello script parameters.m.

I sensori così implementati sono stati sottoposti alla misurazione delle relative grandezze in diverse condizioni di lavoro, così come è stato effettuato per la validazione del modello del sistema. I risultati ottenuti sono mostrati in tabella 3.

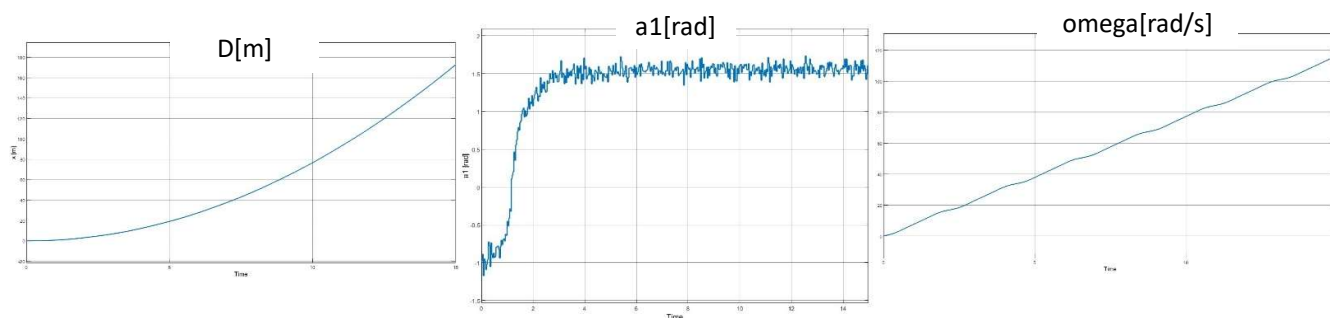
- a) Condizioni iniziali:  $x_0 = 0$ ,  $\varphi_0 = \pi + 0.02$ ,  $\dot{x}_0 = 0$ ,  $\dot{\varphi}_0 = 0$ .  
Ingresso:  $\tau = 0$ .



- b) Condizioni iniziali:  $x_0 = 0$ ,  $\varphi_0 = 0.02$ ,  $\dot{x}_0 = 0$ ,  $\dot{\varphi}_0 = 0$ .  
Ingresso:  $\tau = 0$ .



- c) Condizioni iniziali:  $x_0 = 0$ ,  $\varphi_0 = \pi$ ,  $\dot{x}_0 = 0$ ,  $\dot{\varphi}_0 = 0$ .  
Ingresso:  $\tau = 1$ .



- d) Condizioni iniziali:  $x_0 = 0$ ,  $\varphi_0 = \pi$ ,  $\dot{x}_0 = 0$ ,  $\dot{\varphi}_0 = 0$ .  
Ingresso:  $\tau = \sin(ft)$ ,  $f = 1\text{Hz}$ .

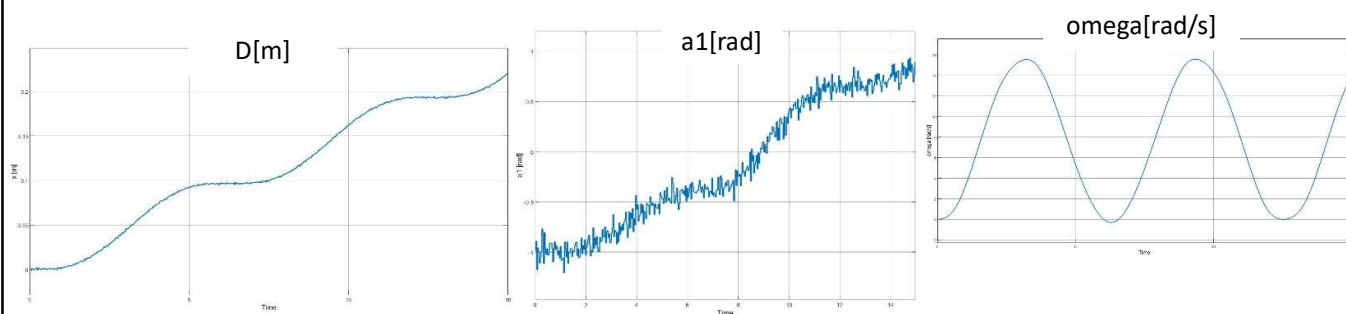


Tabella 3.

La combinazione di ingressi applicati e le rispettive condizioni iniziali adottate per testare il comportamento dei sensori sono analoghe a quelle adottate per la validazione del modello del sistema: in tutti i casi la misura della posizione e della velocità angolare risultano coerenti con le uscite nominali del modello a meno del rumore.

Per quanto riguarda la misura dell'angolo  $a_1$  questa fa uso della funzione  $\text{atan2}$  che restituisce valori tra  $-\pi$  e  $\pi$  il che giustifica i valori negativi assunti. Da notare inoltre che quando il dispositivo passa sotto il sensore la misura di  $a_1$  subisce uno sfasamento di circa  $\pi$ , ma tale comportamento non influisce la stima tramite EKF/UKF. Il sensore per la misura di  $a_1$  è stato testato in diverse posizioni: in corrispondenza dell'asta a destra e a sinistra di questa, con  $\varphi_0 = 0$ , restituendo i valori desiderati.

Nota: per i casi a) e b) si applica una  $\sigma_{a_1} = 1e - 4$  per rendere il grafico leggibile. L'uscita è analoga a quella del sensore reale a meno di un rumore meno accentuato.

## Stima dinamica dello stato

Tecniche di filtraggio assegnate: EKF, UKF.

Entrambe le tecniche sono state implementate mediante tecniche ricorsive basate su predizione e correzione:

La fase di predizione della stima del sistema (variabili di stato:  $x, \varphi$ ), è stata effettuata sulla base di un Modello Cinematico descritto dalla funzione  $f: z_{k+1} = f(z_k, w_k)$ :

$$\begin{cases} x_{k+1} = x_k + dT[(\omega_k + w_{\omega k})r] \\ \varphi_{k+1} = \varphi_k + dT(w_{\varphi k}) \end{cases}$$

$$\text{Con: } z_k = \begin{bmatrix} x_k \\ \varphi_k \end{bmatrix}, \quad w_k = \begin{bmatrix} w_{\omega k} \\ w_{\varphi k} \end{bmatrix} \sim \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, Q_k \right), \quad Q_k = \begin{bmatrix} \sigma_{w_{\omega}}^2 & 0 \\ 0 & \sigma_{w_{\varphi}}^2 \end{bmatrix},$$

$$\sigma_{w_{\omega}} = \sigma_{\omega}, \quad \sigma_{w_{\varphi}} = 1e-5 \text{ [rad/s]}$$

Per la fase di correzione della stima si utilizzano le equazioni dei sensori che costituiscono la funzione  $h: y_k = h(z_k, v_k)$ :

$$\begin{cases} D_k = x_k + v_{Dk} \\ a_{1k} = \text{atan2}\left(x - S_{a_1} + L \sin \varphi, L_{a_1} - (L \cos \varphi + r)\right) + v_{a_1k} \end{cases}$$

$$\text{Con: } y_k = \begin{bmatrix} D_k \\ a_{1k} \end{bmatrix}, \quad v_k = \begin{bmatrix} v_{Dk} \\ v_{a_1k} \end{bmatrix} \sim \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, R_k \right), \quad R_k = \begin{bmatrix} \sigma_D^2 & 0 \\ 0 & \sigma_{a_1}^2 \end{bmatrix}.$$

Poiché il rumore  $v_k$  è additivo, possiamo scrivere  $h: y_k = h'(z_k) + v_k$

Nota: l'equazione del terzo sensore non è inclusa nella funzione  $h$  poiché  $\omega$  nominale viene già utilizzata nel Modello Cinematico e non avrebbe alcun senso correggere la stima con la misura di tale grandezza.

Per entrambe le tecniche di filtraggio (EKF e UKF) si assume:

- periodo di discretizzazione  $dT = T_{\omega}$  (sensore più veloce)
- tempo di simulazione  $t_{max} = 15s$
- numero di iterazioni  $n = \frac{t_{max}}{dT}$  (inteso come divisione intera)
- inizializzazione:  $P_0 = \begin{bmatrix} \sigma_{init}^2 & 0 \\ 0 & \sigma_{init}^2 \end{bmatrix},$

### ○ Stima tramite Extended Kalman Filter (EKF)



Il Filtro di Kalman Esteso permette di stimare lo stato del sistema utilizzando un modello linearizzato.

Predizione:  $(k|k \rightarrow k+1|k)$

$$\hat{z}_{k+1|k} = f\left(\hat{z}_{k|k}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right) \text{ (stima predetta)}, \quad P_{k+1|k} = F_k P_{k|k} F_k^T + D_k Q_k D_k^T \text{ (matr. di cov.)}$$

$$\text{Con: } F = \frac{\partial f}{\partial z_k} \Big|_{z_k=\hat{z}_{k|k}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad D = \frac{\partial f}{\partial w_k} \Big|_{z_k=\hat{z}_{k|k}} = \begin{bmatrix} r d^T & 0 \\ 0 & d^T \end{bmatrix}$$

$$w_k = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Correzione:  $(k|k-1 \rightarrow k|k)$

$$e_k = y_k - h(\hat{z}_{k|k-1}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}) \text{ (innovazione)}$$

$$S_k = H P_{k|k-1} H^T + M R M^T \text{ (matrice di cov. dell'innovazione)}$$

$$L_k = P_{k|k-1} H_k^T S_k^{-1} \text{ (guadagno di correzione)}$$

$$\hat{z}_{k|k} = \hat{z}_{k|k-1} + L e_k \text{ (stima corretta)}$$

$$P_{k|k} = (I - L H) P_{k|k-1} (I - L H)^T + L M R M^T L \text{ (matrice di cov. stima corretta - forma di Joseph)}$$

$$\text{Con: } H = \frac{\partial h}{\partial z_k} \Big|_{z_k=\hat{z}_{k|k-1}} = \begin{bmatrix} 1 & 0 & 0 \\ a & b & 0 \end{bmatrix}, \quad M = \frac{\partial h}{\partial v_k} \Big|_{z_k=\hat{z}_{k|k-1}} = I_{2 \times 2},$$

$$v_k = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$a = - \frac{L_{a_1} - (L \cos \varphi_k + r)}{(L_{a_1} - (L \cos \varphi_k + r))^2 + (S_{a_1} - (L \sin \varphi_k + x))^2}$$

$$b = \frac{N1}{(L_{a_1} - (L \cos \varphi_k + r))^2 + (S_{a_1} - (L \sin \varphi_k + x))^2}$$

Nota: rumore additivo  $\Rightarrow M = I$

Poiché i sensori hanno periodi di campionamento differenti e poiché la frequenza di filtraggio è impostata secondo  $T_{\text{sensor\_omega}}$  (più veloce), le misure non sono sempre tutte disponibili ad ogni istante. Per questo la fase di correzione della stima viene eseguita solo quando le ultime misure effettuate per  $D$  e  $a_1$  sono entrambe non utilizzate. Poiché le misure non utilizzate sono quelle provenienti dal sensore  $D$  che anche con meno correzioni di quelle possibili riesce comunque a correggere correttamente. In particolare, sono stati creati due vettori di flag associati alle misure ed al relativo istante, quindi di lunghezza pari al numero di misure effettuate rispettivamente dai due sensori. I flag vengono impostati ad 1 solo quando l'ultima misura effettuata dal sensore non è ancora stata utilizzata per la fase di correzione, questo è possibile utilizzando la funzione find che trova l'ultimo indice associato alla misura effettuata in un tempo minore o uguale al tempo corrente. Tale indice viene valutato rispetto al vettore dei flag del

sensore per capire se la misura è stata utilizzata o meno. Quando entrambi i flag dei sensori relativi ad una misura sono impostati ad 1, allora viene eseguita la fase di correzione.

#### ○ **Stima tramite Unscented Kalman Filter (UKF)**

Il Filtro di Kalman Unscented permette di stimare lo stato del sistema utilizzando direttamente le funzioni non lineari dalle quali è possibile estrarlo (insieme alla matrice di covarianza) applicando la Trasformata Unscented.

Predizione:  $(k|k \rightarrow k+1|k)$

$$z_{k+1} = f(z_k, w_k), \quad z'_k = \begin{bmatrix} z_k \\ w_k \end{bmatrix},$$

$$\begin{bmatrix} \hat{z}_{k+1|k} & P_{k+1|k} & * \end{bmatrix} = UT \left[ \begin{bmatrix} \hat{z}_{k|k} \\ 0^2 \end{bmatrix}, \begin{bmatrix} P_{k|k}^{2 \times 2} & 0^{2 \times 2} \\ 0^{2 \times 2} & Q_k^{2 \times 2} \end{bmatrix}, f(z_k, w_k) \right]$$

\*: il momento incrociato restituito dalla UT non è di interesse.

Algoritmo Trasformata Unscented per la funzione  $f$ :

- Scelta parametri:

$\alpha = 1$  (Trasformata UT non scalata)

$\beta = 2$  (Ottimo nel caso Gaussiano)

$k = 0$

$n = \dim(z'_k) = 4$

$\lambda = \alpha^2(n + k) - n$

- Calcolo pesi:

$$w_0 = \frac{\lambda}{n + \lambda}, \quad w'_0 = w_0 + 1 - \alpha^2 + \beta, \quad w_i = w'_i = \frac{1}{2(n + \lambda)} \text{ per } i = \pm 1, \pm 2, \dots, \pm n$$

- Calcolo sigma point:

fattorizzazione matrice di covarianza a priori tramite decomposizione a valori singolari (SVD):

$$\Sigma_{z'_k} = \begin{bmatrix} P_{k|k}^{2 \times 2} & 0^{2 \times 2} \\ 0^{2 \times 2} & Q_k^{2 \times 2} \end{bmatrix} = V \Lambda V^T = \Gamma \Gamma^T = \sum_{i=1}^n \gamma_i \gamma_i^T$$

con:  $\Gamma = V \Lambda^{1/2}$  e  $\gamma_i = i\text{-esima colonna di } \Gamma$

generazione dei  $2n + 1$  sigma point:

$$\begin{cases} z_0 = \begin{bmatrix} \hat{z}_{k|k} \\ 0^2 \end{bmatrix} \\ z_{\pm i} = \begin{bmatrix} \hat{z}_{k|k} \\ 0^2 \end{bmatrix} \pm \sqrt{n + \lambda} \gamma_i \quad \text{con } i = 1, 2, \dots, n \end{cases}$$

% sigma\_point=matrice 4x(2n+1), ogni colonna è un sigma point

- Calcolo sigma point trasformati tramite la funzione  $f$ :

$$\hat{z}_{k+1|k\ i} = f(z_i) \quad \text{con } i = -n, \dots, 0, 1, 2, \dots, n$$

- Calcolo stima predetta e relativa covarianza tramite media campionaria:

$$\hat{z}_{k+1|k} = \sum_{i=-n}^n w_i \hat{z}_{k+1|k\ i}$$

$$P_{k+1|k} = \sum_{i=-n}^n w_i' (\hat{z}_{k+1|k\ i} - \hat{z}_{k+1|k}) (\hat{z}_{k+1|k\ i} - \hat{z}_{k+1|k})^T$$

Correzione: ( $k|k-1 \rightarrow k|k$ )

$$y_k = h(z_k, v_k) = h'(z_k) + v_k$$

$$[\hat{y}_{k|k-1} \quad \Sigma'_k \quad \Sigma'_{zy,k}] = UT[\hat{z}_{k|k-1} \quad P_{k|k-1} \quad h'(z_k)]$$

$$e_k = y_k - \hat{y}_{k|k-1} \quad (\text{innovazione})$$

$$S_k = \Sigma'_k + R_k \quad (\text{matrice di cov. dell'innovazione})$$

$$L_k = \Sigma'_{zy,k} S_k^{-1} \quad (\text{guadagno di correzione})$$

$$\hat{z}_{k|k} = \hat{z}_{k|k-1} + L e_k \quad (\text{stima corretta})$$

$$P_{k|k} = P_{k|k-1} - L_k S_k L_k^T$$

Algoritmo Trasformata Unscented per la funzione  $h$ :

- Scelta parametri:

$$\alpha = 1 \quad (\text{Trasformata UT non scalata})$$

$$\beta = 2 \quad (\text{Ottimo nel caso Gaussiano})$$

$$k = 0$$

$$n = \dim(z'_k) = 6$$

$$\lambda = \alpha^2(n + k) - n$$

- Calcolo pesi:

$$w_0 = \frac{\lambda}{n+\lambda}, \quad w'_0 = w_0 + 1 - \alpha^2 + \beta, \quad w_i = w'_i = \frac{1}{2(n+\lambda)} \text{ per } i = \pm 1, \pm 2, \dots, \pm n$$

- Calcolo sigma point:

fattorizzazione matrice di covarianza a priori tramite decomposizione a valori singolari (SVD):

$$\Sigma_{z_k} = [P_{k|k-1}] = V \Lambda V^T = \Gamma \Gamma^T = \sum_{i=1}^n \gamma_i \gamma_i^T$$

$$\text{con: } \Gamma = V \Lambda^{1/2} \quad \text{e} \quad \gamma_i = i\text{-esima colonna di } \Gamma$$

generazione dei  $2n + 1$  sigma point:

$$\begin{cases} z_0 = [\hat{z}_{k|k-1}] \\ z_{\pm i} = [\hat{z}_{k|k-1}] \pm \sqrt{n + \lambda} \gamma_i \quad \text{con } i = 1, 2, \dots, n \end{cases}$$

% sigma\_point=matrice 2x(2n+1), ogni colonna è un sigma point

- Calcolo sigma point trasformati tramite la funzione  $h$ :

$$\hat{y}_{k|k-1 i} = h(z_i) \quad \text{con } i = -n, \dots, 0, 1, 2, \dots, n$$

- Calcolo media, covarianza e covarianza incrociata:

$$\hat{y}_{k|k-1} = \sum_{i=-n}^n w'_i \hat{y}_{k|k-1 i}$$

$$\Sigma'_k = \sum_{i=-n}^n w'_i (\hat{y}_{k|k-1 i} - \hat{y}_{k|k-1})(\hat{y}_{k|k-1 i} - \hat{y}_{k|k-1})^T$$

$$\Sigma'_{zy,k} = \sum_{i=-n}^n w'_i (z_i - \hat{z}_{k|k-1})(\hat{y}_{k|k-1 i} - \hat{y}_{k|k-1})^T$$

## Risultati (stima)

Entrambi gli stimatori (EKF e UKF) sono stati testati in differenti condizioni di lavoro. Di seguito sono riportati tre casi di maggiore interesse:

Caso 1 :

Condizioni iniziali:  $x_0 = 0$ ,  $\varphi_0 = \pi + 0.02$

Ingresso:  $\tau = 0$

Dev standard iniziale:  $\sigma_{init} = 0.01$

Disturbi di processo:  $\sigma_{w_\varphi} = 0.1 \text{ [rad/s]}$

Sensore a1:  $\sigma_{a_1} = 1e - 4 \text{ [rad]}$

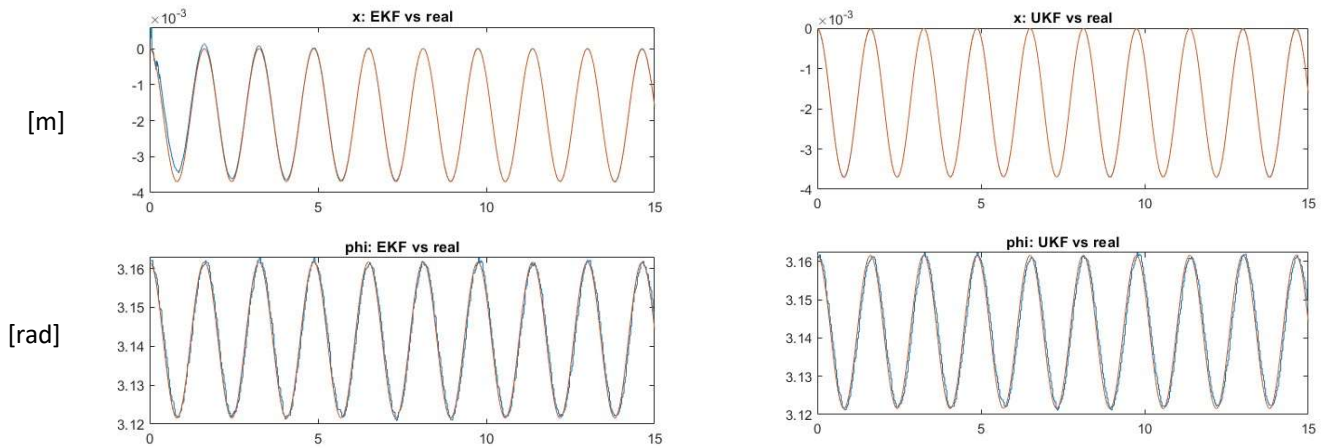


Fig. 3

Entrambe le stime seguono in maniera corretta l'andamento reale sia della posizione  $x$  che dell'angolo  $\varphi$ , da notare come l'errore di stima negli istanti iniziali per la  $x$  nell'EKF (approssimazione al 1° ordine) si annulla con l'UKF poiché in questo caso l'approssimazione è al 2° ordine.

Tale comportamento si presenta solo se  $\sigma_{init}$  e  $\sigma_{w\varphi}$  sono dell'ordine di grandezza sopra specificati.

In particolare, all'aumentare dell'incertezza iniziale  $\sigma_{init}$  l'errore di stima aumenta: per valori di  $\sigma_{init}$  di ordini superiori a 0.1 la stima non risulta coerente con l'andamento reale dello stato.

Importante specificare inoltre che l'incertezza del sensore per la misura dell'angolo  $a_1$  non è quella del sensore reale inquanto anche in questo caso la stima dello stato non risulta coerente. Tale comportamento è legato alla complessità del modello del sensore per il calcolo di  $a_1$  che a differenza degli altri due sensori utilizzati non risulta essere una misura diretta.

Se invece si diminuisce  $\sigma_{w\varphi}$  di un ordine di grandezza, la stima della posizione  $x$  peggiora mentre quella dell'angolo  $\varphi$  presenta uno sfasamento rispetto all'andamento reale come mostrato in Fig4. Tale comportamento è legato alle caratteristiche del modello cinematico utilizzato.

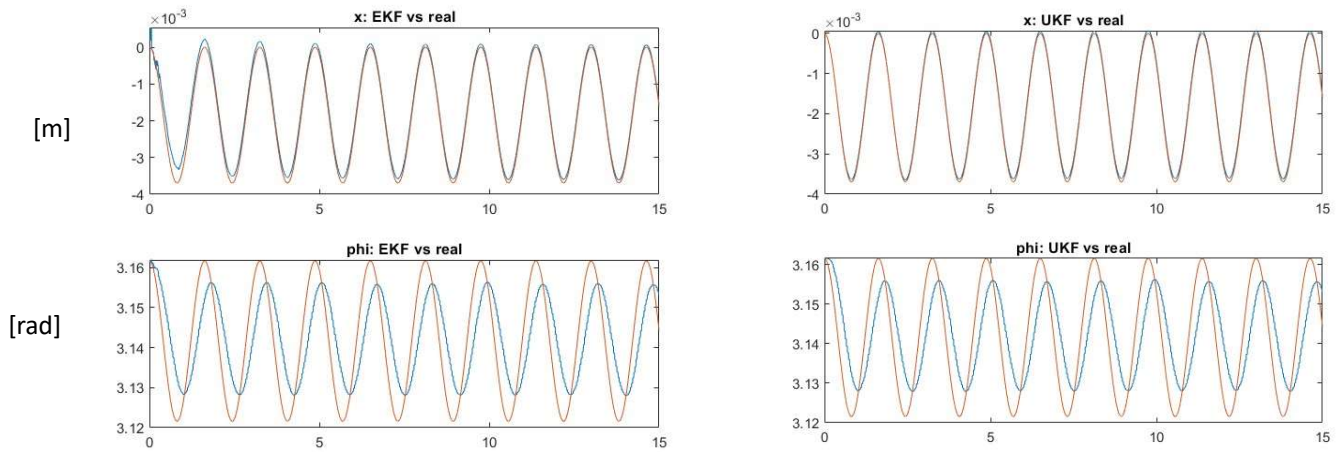


Fig. 4

Caso 2:

Condizioni iniziali:  $x_0 = 0$ ,  $\varphi_0 = \pi$

Ingresso:  $\tau = 0.01$  [N/m]

Dev standard iniziale:  $\sigma_{init} = 0.01$

Disturbi di processo:  $\sigma_{w\varphi} = 0.1$  [rad/s]

Sensore a1:  $\sigma_{a_1} = 1e - 4$  [rad]

In questo caso, la stima della posizione  $x$  risulta esatta per entrambi i filtri, mentre per l'angolo  $\varphi$  si presenta un errore di stima che aumenta con il tempo dovuto alle approssimazioni del modello cinematico Fig. 5.

Anche in questo caso l'incertezza del sensore per  $a_1$  non è quella del dispositivo reale.

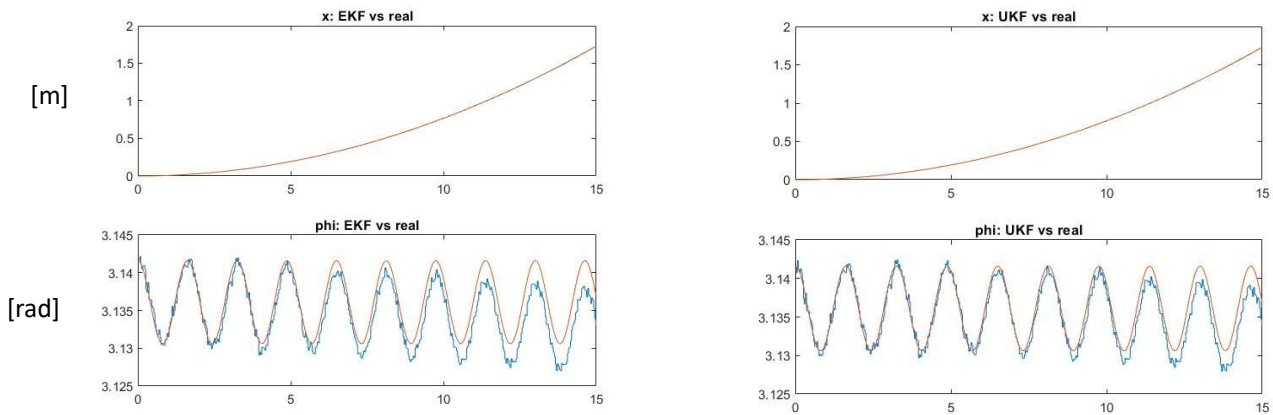


Fig. 5

Caso 3:

Condizioni iniziali:  $x_0 = 0, \varphi_0 = 0.02$

Ingresso:  $\tau = 0$

Dev standard iniziale:  $\sigma_{init} = 0.01$

Disturbi di processo:  $\sigma_{w\varphi} = 0.1 \text{ [rad/s]}$

Sensore a1:  $\sigma_{a_1} = 1e - 4 \text{ [rad]}$

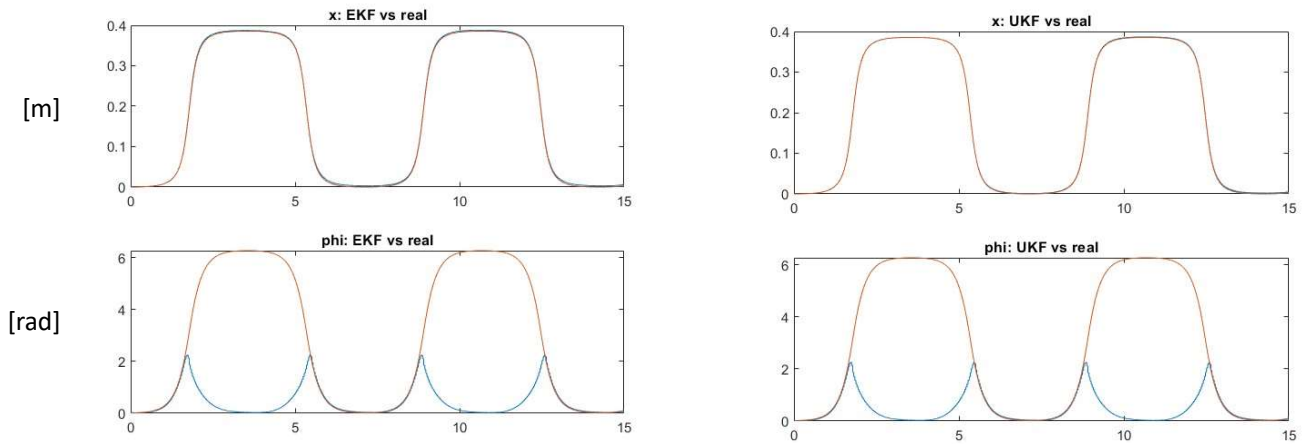


Fig. 6

A causa delle ampie oscillazioni dell'asta (che parte dalla condizione di equilibrio instabile) la stima dell'angolo  $\varphi$  non risulta corretta. Si presume che tale comportamento sia legato ad una errata valutazione dell'angolo  $\alpha_1$  per ampie oscillazioni il quale non risulta avere valori univoci se rilevato con un solo sensore. La stima della posizione lineare risulta invece esatta.

## Regolarizzazione

L'algoritmo di regolarizzazione di Ranch, Tung e Striebel (1965) è un metodo utilizzato per migliorare la stabilità e la precisione della stima nei sistemi dinamici, soprattutto quando si utilizzano il filtro di Kalman esteso (EKF). L'obiettivo della regolarizzazione è mitigare i problemi di instabilità numerica e di propagazione dell'errore nel tempo. È composto da due fasi:

1. In avanti:  
si salvano durante ogni passo i dati dell'EKF, in particolare: la stima dello stato, la varianza predette e corrette, e la matrice F.
2. All' indietro:

$C_k = P_{k|k} F_{k+1}^T P_{k+1|k}$  , *parametro di regolarizzazione* che determina l'importanza della regolarizzazione rispetto la stima originale

$\hat{z}_{k|n} = \hat{z}_{k|k} + C_k (\hat{z}_{k+1|n} - \hat{z}_{k+1|k})$ , aggiornamento della stima con regolarizzazione

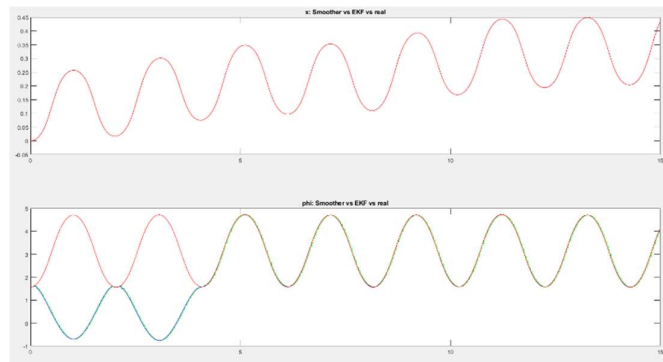
$$P_{k|n} = P_{k|k} - C_k (P_{k+1|n} - P_{k+1|k})$$

L'algoritmo partirà dagli ultimi valori dello stimatore fino ai primi:

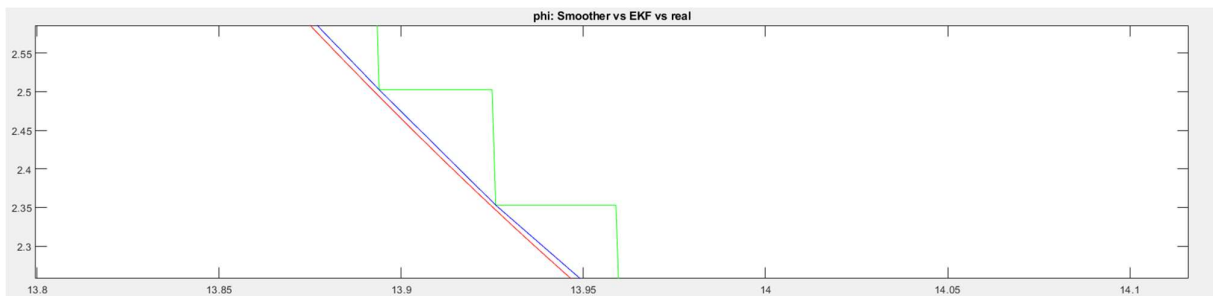
$$n|n \rightarrow n-1|n \rightarrow n-2|n \rightarrow \dots \rightarrow 1|n$$

### Risultati (regolarizzazione)

In base ai risultati della stima, la regolarizzazione avrà più o meno rilevanza; a seguire un esempio dove si ha una maggiore rilevanza nel caso della variabile di stato *phi*.



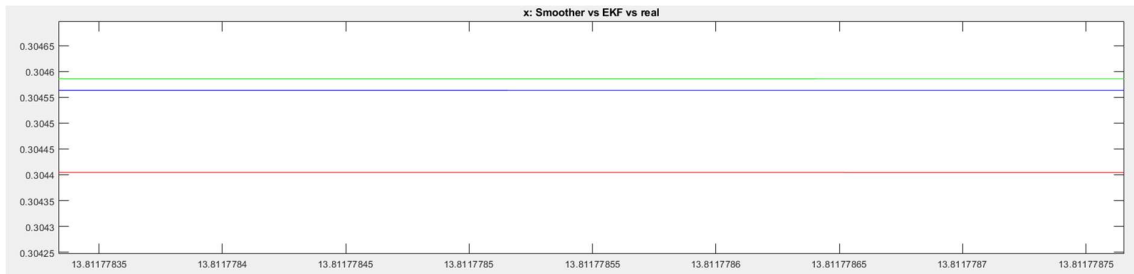
Esempio di regolarizzazione riguardante la variabile di stato *phi*. In figura il paragone tra stima regolarizzata (blu), stima con EKF (verde) e sistema reale (rosso).



Il risultato precedente mette in risalto come la regolarizzazione è riuscita a 'sistemare' il risultato ottenuto dall'EKF, avendo a disposizione le stime totali.



A seguire la regolarizzazione ottenuta con la variabile di stato  $x$ . Poiché di per sé era già precisa, la regolarizzazione 'sistema' meno rispetto al caso precedente della variabile di stato  $\phi$ .



Nota: esempio effettuato con i seguenti parametri:  $C\_input = 0.01$ ,  $\phi_0 = \pi/2$ ,  $std\_dev\_phi = 1e-4$ ,  $std\_dev\_init = 1e-20$ .

## Note e bibliografia

$$N1 = -(L*(L - L\_a1*\cos(\phi\_curr) - S\_a1*\sin(\phi\_curr) + r*\cos(\phi\_curr) + x\_curr*\sin(\phi\_curr)))/((r - L\_a1 + L*\cos(\phi\_curr))^2 + (x\_curr - S\_a1 + L*\sin(\phi\_curr))^2);$$

- [1] P. Purohit, P. Modi, and U. Vyas, "Kinematic Control of 2-wheeled Segway." [Online]. Available: [https://github.com/uditvyas/Segway\\_Controller.git](https://github.com/uditvyas/Segway_Controller.git)
- [2] [https://cdn.usdigital.com/assets/datasheets/MA3\\_datasheet.pdf?k=635549412619806250](https://cdn.usdigital.com/assets/datasheets/MA3_datasheet.pdf?k=635549412619806250)
- [3] <https://www.reichelt.com/it/it/hokuyo-utm-30lx-ew-lidar-utm-30lx-ew-p320261.html>
- [4] <https://www.intelrealsense.com/depth-camera-d435/>