



# Localizzazione e Tracking tramite il Filtro di Kalman Esteso

Università Degli Studi di L'Aquila  
Corso di Intelligent Systems And Robotics Laboratory  
Prof. Giovanni De Gasperis

*Valentina D'Orazio, mat. 252552*

## Indice

Introduzione.....	3
1. RSSI.....	3
2. L'errore nel canale radio .....	4
Modello Cinematico tempo discreto e Filtro di Kalman Esteso (EFK).....	5
1. Equazioni cinematiche.....	5
2. EKF .....	6
2.1 Modello Cinematico a tempo discreto nello spazio di stato .....	7
Simulazione Matlab.....	9
Simulazione Vrep e Python .....	11
1. Modellazione della scena .....	11
2. Collegamento Vrep-Python .....	12
Riferimenti: .....	15

# Introduzione

Il radio posizionamento viene adottato negli ultimi anni in moltissime applicazioni industriali, andando a sfruttare come sensori i nodi di una rete Wireless la quale non viene utilizzata solo per lo scambio di informazioni, ma anche come sensori attivi in grado di fornire misure utili alla localizzazione. In particolare, l'evoluzione tecnologica dei sistemi radio, come ad esempio le tecnologie Ultra Wideband, consentono di effettuare misurazioni di distanza affidabili anche se affette da rumore.

Il progetto sviluppato si basa su uno studio di fattibilità dell'applicazione del Filtro di Kalman al problema del positioning indoor. Lo scenario applicativo consiste in un'infrastruttura wireless composta da delle antenne posizionate in maniera fissa e nota, le quali fungono da emettitori di segnale, e da un'antenna ricevente che ha lo scopo di effettuare le misure di potenza (RSSI) dei segnali emessi dalle antenne. Essendo noto il legame tra potenza e distanza tra le antenne è possibile risalire alla posizione spaziale in tempo reale dell'oggetto in movimento, come ad esempio un drone, sul quale è montato il nodo ricevente.

La computazione dell'algoritmo del Filtro di Kalman è di tipo centralizzata in quanto le antenne posizionate nella struttura inviano solo i segnali.

## 1. RSSI

L'RSSI (Received Signal Strength Indicator) è una misura della potenza presente in un segnale radio allo scopo di fornire una stima della posizione di un nodo all'interno della rete.

La potenza del segnale ricevuto è definita come il rapporto tra la potenza ricevuta  $P_r$  e la potenza di riferimento  $P_{ref}$

$$RSSI = 10 \log_{10} \frac{P_r}{P_{ref}} = [dBm]$$

È facile intuire come ad un incremento della potenza ricevuta segue un incremento del valore di RSSI. In particolare, il legame tra la potenza del segnale ricevuto e la distanza è dato dalla seguente formula:

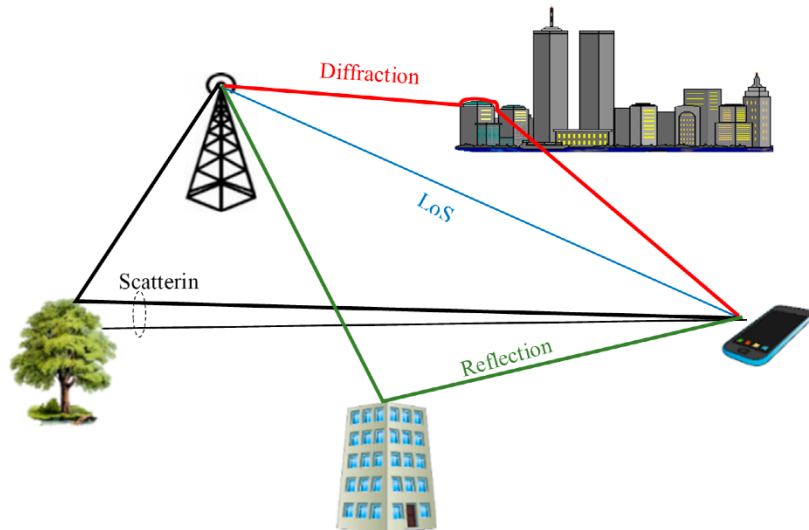
$$RSSI = -(10n \log_{10} d + A)$$

Da cui si ricava che:

$$d = 10^{\frac{A - RSSI}{10n}}$$

## 2. L'errore nel canale radio

Durante la trasmissione sul canale radio, il segnale è soggetto ad attenuazione dovuta a diversi fattori quali, ad esempio, shadowing e multipath. A causa del multipath, in ricezione si possono avere diverse repliche di uno stesso segnale provenienti da diversi percorsi come si evince in figura:



Le cause del multipath possono essere diverse tra cui la riflessione, la rifrazione, la presenza di edifici o oggetti terrestri come le montagne (questo è chiamato shadowing). Esistono diverse tecniche di espansione spettrale (come ad esempio frequency-hopping o direct-sequence) per ridurre l'effetto di tali fenomeni appena citati.

Shadowing e multipath, sono le principali fonti d'errore per l'RSSI.

L'incertezza della misura dell'RSSI può essere modellata tramite una variabile aleatoria log-normale (espressa in decibel). Solitamente la varianza dell'errore di misura è costante, ciò provoca (rispetto la stima della distanza) una covarianza dell'errore direttamente proporzionale alla distanza tra nodo mobile e i nodi ancora.

# Modello Cinematico tempo discreto e Filtro di Kalman Esteso (EFK)

## 1. Equazioni cinematiche

Le equazioni prese in considerazione nel progetto descrivono il moto uniformemente accelerato ossia è un tipo di moto in cui un corpo si muove con accelerazione costante. Di conseguenza l'accelerazione media e l'accelerazione istantanea coincidono in qualsiasi istante di tempo e assumono il medesimo valore a che per semplicità viene chiamato accelerazione senza distinzioni. Per questo tipo di moto le variazioni di velocità risultano essere direttamente proporzionali agli intervalli di tempo. Il sistema di equazioni a tempo discreto utilizzato è il seguente:

$$\left\{ \begin{array}{l} p_{x_{k+1}} = p_{x_k} + v_{x_k} \Delta T + \frac{1}{2} \Delta T^2 a_{x_k} \\ v_{x_{k+1}} = v_{x_k} + a_{x_k} \Delta T \\ a_{x_{k+1}} = a_{x_k} \\ p_{y_{k+1}} = p_{y_k} + v_{y_k} \Delta T + \frac{1}{2} \Delta T^2 a_{y_k} \\ v_{y_{k+1}} = v_{y_k} + a_{y_k} \Delta T \\ a_{y_{k+1}} = a_{y_k} \\ p_{z_{k+1}} = p_{z_k} + v_{z_k} \Delta T + \frac{1}{2} \Delta T^2 a_{z_k} \\ v_{z_{k+1}} = v_{z_k} + a_{z_k} \Delta T \\ a_{z_{k+1}} = a_{z_k} \end{array} \right.$$

In cui con

$v$  si intende la velocità lungo i tre assi,

$a$  indica l'accelerazione,

$p$  rappresenta la posizione,

$\Delta T$  è il tempo di campionamento.

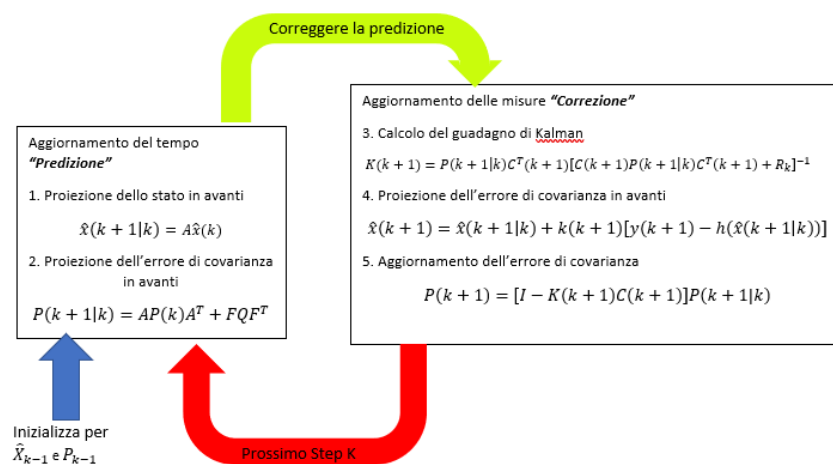
Ogni ancora fornirà la seguente uscita non lineare:

$$h_t(X) = -(10n \log_{10} \sqrt{(x - x_t)^2 + (y - y_t)^2 + (z - z_t)^2} + A)$$

## 2. EKF

Il filtro di Kalman è un algoritmo ricorsivo che risolve il problema della stima ottima dello stato per sistemi lineari a tempo discreto con rumore bianco gaussiano additivo che agisce sullo stato e sulle osservazioni dell'uscita. Per stima ottima si intende stima a minima varianza dell'errore.

Il filtro di Kalman stima lo stato del processo in istanti di tempo determinati (tempo di campionamento) sulla base di misure affette da rumore gaussiano additivo.



Il filtro di Kalman Esteso, utilizzato per la stima dello stato di sistemi non lineari, consiste nell'applicare l'algoritmo ricorsivo sfruttando la linearizzazione del sistema nell'intorno della stima corrente. Un sistema non lineare a tempo discreto, formato dall'evoluzione dello stato e dalla sua osservazione, può essere scritto in forma generalizzata come segue:

$$x(k+1) = f(k, x(k), u(k)) + F(k)N_k$$

$$y(k) = h(k, x(k), u(k)) + G(k)N_k$$

dove,  $x(k)$  è lo stato,  $u(k)$  l'ingresso. Il rumore di stato all'istante  $k$  è pari a  $F(k)N_k$ , il rumore di osservazione all'istante  $k$  è  $G(k)N_k$ .  $f(\cdot)$  ed  $h(\cdot)$  sono funzioni non lineari dell'uscita e dello stato.

## 2.1 Modello Cinematico a tempo discreto nello spazio di stato

Considerando le equazioni cinematiche descritte nel paragrafo 1, si può determinare la rappresentazione matriciale nello spazio di stato che descrive il modello utilizzato nel filtro di Kalman.

Vettore di stato:

$$x = \begin{bmatrix} p_{x_k} \\ v_{x_k} \\ a_{x_k} \\ p_{y_k} \\ v_{y_k} \\ a_{y_k} \\ p_{z_k} \\ v_{z_k} \\ a_{z_k} \end{bmatrix}$$

Matrice A di evoluzione dello stato:

$$A = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & \frac{\Delta t^2}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

e la matrice F:

$$F = \begin{bmatrix} \frac{\Delta T^2}{2} & 0 & 0 \\ \Delta T & 0 & 0 \\ 1 & 0 & 0 \\ 0 & \frac{\Delta T^2}{2} & 0 \\ 0 & \Delta T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{\Delta T^2}{2} \\ 0 & 0 & \Delta T \\ 0 & 0 & 1 \end{bmatrix}$$

Il modello introdotto nel precedente paragrafo rappresenta un sistema a tempo discreto con l'uscita non lineare. Per poter applicare il Filtro di Kalman Esteso risulta necessario linearizzare l'uscita. In particolare, applicando gli sviluppi di Taylor arrestati al primo ordine, si ottiene:

$$y(k) \approx h(\hat{X}(k|k-1)) + \frac{dh}{dx} \big|_{x=\hat{X}(k|k-1)} (X(k) - \hat{X}(k|k-1)) + g_k$$

Data la formula dell'RSSI introdotta nella sezione 1, si ricava la distanza tra un'ancora e l'antenna posizionata sul drone che rappresenta l'uscita non lineare:

$$h_i(X) = (x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = d_i^2$$

L'uscita linearizzata  $C(x)$  si ottiene applicando lo Jacobiano:

$$C_i(X) = \frac{dh}{dx} \big|_{x=\hat{X}(k|k-1)} = \left[ \frac{dh_i(X)}{dp_x} \quad \frac{dh_i(X)}{dv_x} \quad \frac{dh_i(X)}{da_x} \quad \frac{dh_i(X)}{dp_y} \quad \frac{dh_i(X)}{dv_y} \quad \frac{dh_i(X)}{da_y} \quad \frac{dh_i(X)}{dp_z} \quad \frac{dh_i(X)}{dv_z} \quad \frac{dh_i(X)}{da_z} \right]$$

$$C(x) = \begin{bmatrix} 2(x - x_1) & 0 & 0 & 2(y - y_1) & 0 & 0 & 2(z - z_1) & 0 & 0 \\ 2(x - x_2) & 0 & 0 & 2(y - y_2) & 0 & 0 & 2(z - z_2) & 0 & 0 \\ 2(x - x_3) & 0 & 0 & 2(y - y_3) & 0 & 0 & 2(z - z_3) & 0 & 0 \\ 2(x - x_4) & 0 & 0 & 2(y - y_4) & 0 & 0 & 2(z - z_4) & 0 & 0 \end{bmatrix}$$

Matrice di covarianza dell'errore di stima P:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrice di varianza dell'errore di stato Q:

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrice di varianza dell'errore di misura R:

$$R = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}$$



# Simulazione Matlab

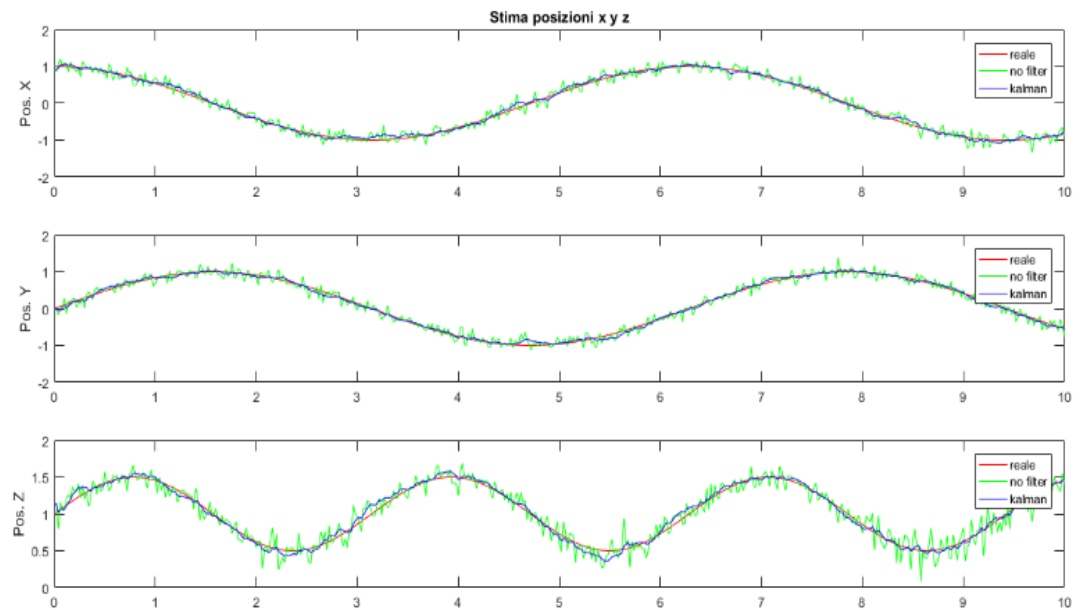
Per valutare le performance del filtro di Kalman, si è fatto ricorso a simulazioni effettuate mediante l'utilizzo di Matlab.

Sono state simulate delle misure affette da rumore gaussiano additivo per mostrare l'efficacia del filtro.

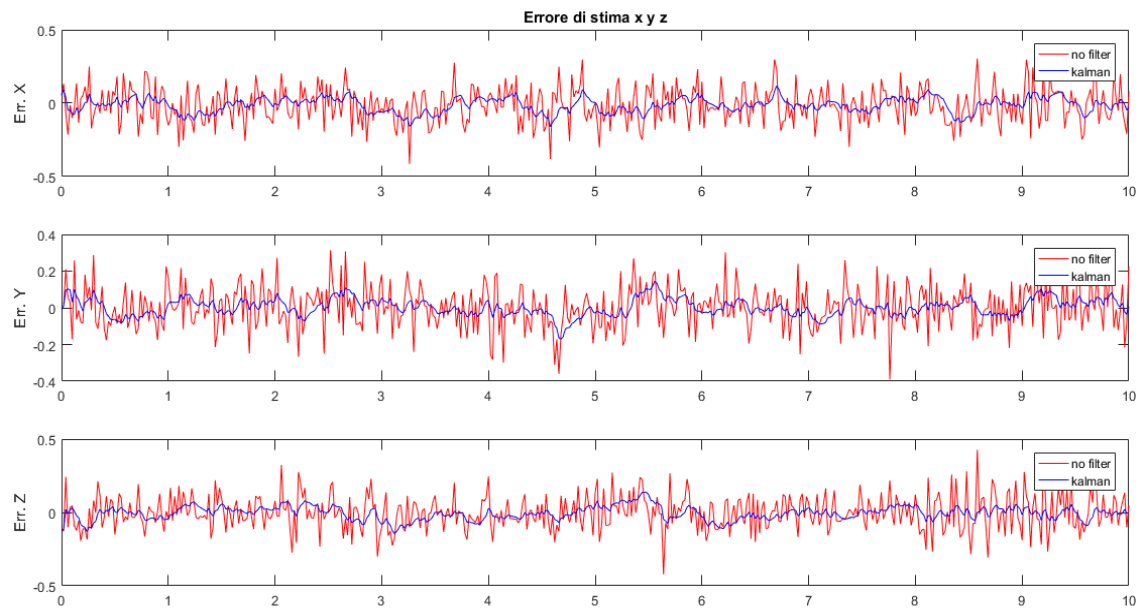
Dato che i sistemi Ultra Wideband hanno una precisione di misura con varianza 0.1.

La varianza di stato è pari ad 1 per far fronte a imprecisioni del modello.

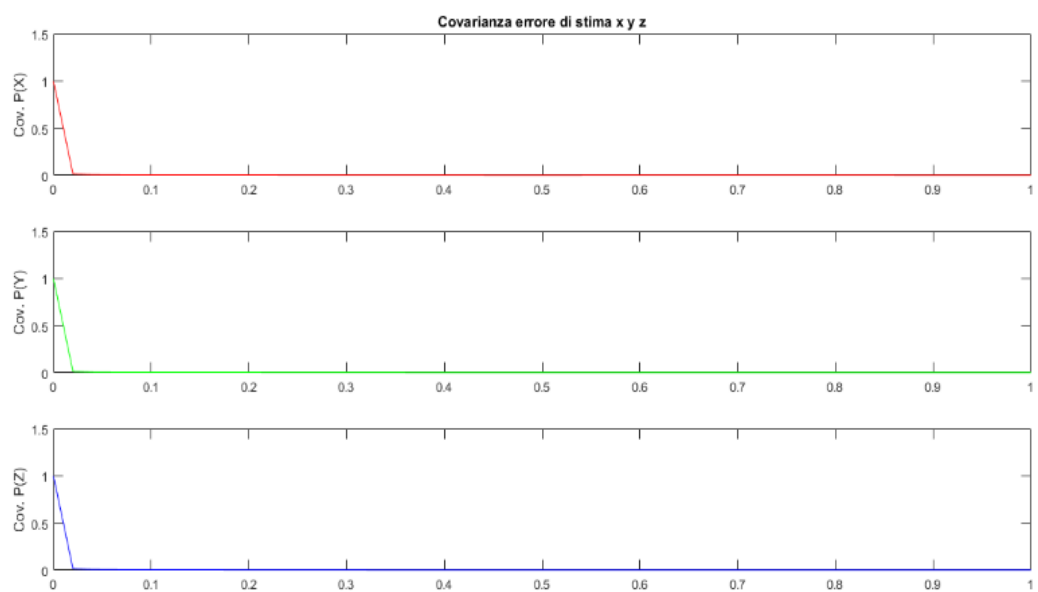
Il seguente grafico mostra una stima delle posizioni mettendo a confronto le misure senza applicare il filtro e quelle ottenute dopo aver applicato il filtro di Kalman:



Il seguente grafico mostra l'errore di stima. Si può notare come con il filtro di Kalman si ha una notevole riduzione dell'errore.



Il seguente grafico mostra che la covarianza dell'errore di stima  $p$  converge a zero: questo vuol dire che il filtro è ottimo.



# Simulazione Vrep e Phytion

Vrep è un ambiente di simulazione di robot che gode di strumenti interni per la simulazione della fisica necessaria per rendere quanto più possibile reali le interazioni degli oggetti modellati. Tali oggetti, nel progetto di nostro interesse, sono controllati attraverso le API di Phytion (vrep-api-python).

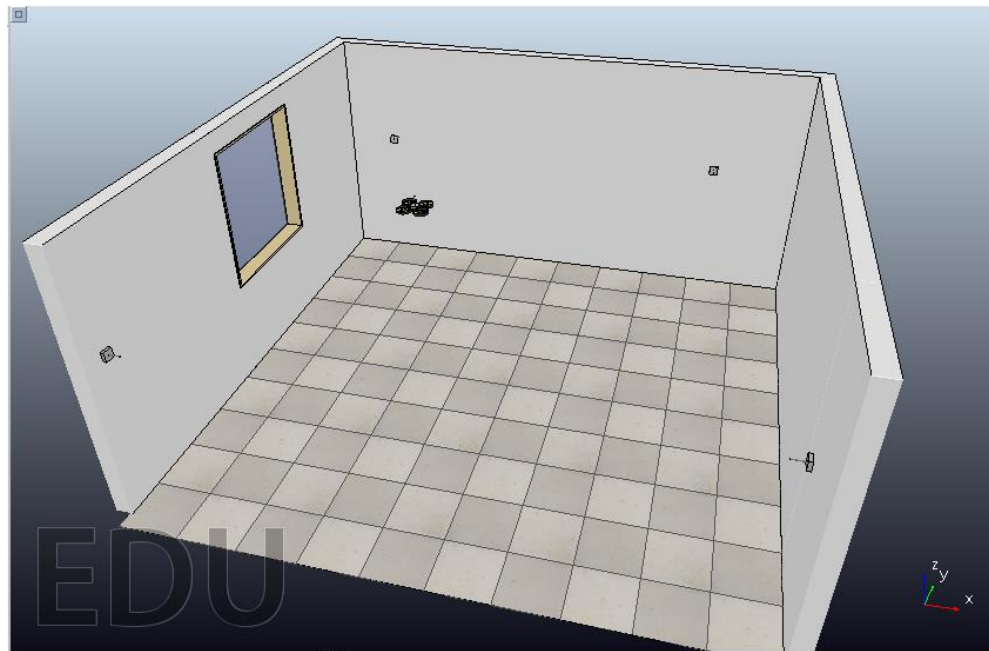
La simulazione è composta da quattro step:

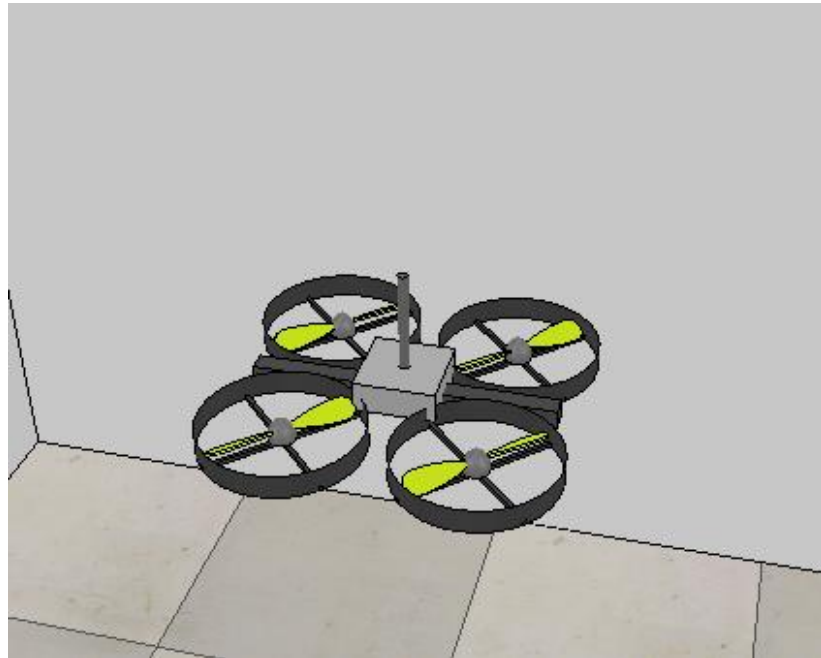
- modellazione della scena
- collegamento Vrep-Pycharm
- algoritmo del filtro di Kalman Esteso in Python
- Simulazione e risultato

## 1. Modellazione della scena

Dopo la creazione di una nuova scena in Vrep, lo spazio operativo contiene solo la pavimentazione e l'indicazione del sistema di riferimento.

Il primo oggetto introdotto è stato il muro per simulare una stanza, successivamente sono state introdotte le ancore sui muri utilizzando l'oggetto transceiver (components → sensors → transceiver). Infine è stato posizionato il drone e un transceiver sul drone per simulare la ricezione dei segnali. Lo scenario completo è mostrato nella figura sottostante:





## 2. Collegamento Vrep-Python

Per poter realizzare un progetto in Pycharm utilizzando le vrep-api-python, si imposta Vrep come server aggiungendo al drone un “Associated child script” di tipo “Thread”. Aprendo lo script appena creato, bisogna aggiungere `simRemoteApi.start(19999)` nella funzione `sysCall_threadmain()` come si evince in figura:

```
1 Main script
2 -- use the require-directive, e.g.:
3 --
4 -- require 'myExternalFile'
5 --
6 -- Above will look for <V-REP executable path>/myExternalFile.lua or
7 -- <V-REP executable path>/lua/myExternalFile.lua
8 -- (the file can be opened in this editor with the popup menu over
9 -- the file name)
10
11 function sysCall_threadmain()
12     -- Put some initialization code here
13     simRemoteApi.start(19999)
14
15
```

Su Pycharm si crea un nuovo progetto inserendo nella cartella appena creata, i seguenti file presi dalla cartella d’installazione di Vrep:

- RemoteApi.dll
- vrep.py
- vrepConst.py

Successivamente è necessario creare un nuovo file di Python e inserire il seguente codice che permette la connessione tra Vrep e Pycharm:

```
# Connect to V-REP
clientID = vrep.simxStart('127.0.0.1', 19999, True, True, 5000, 5)
```

Poiché le ancore sono situate in una posizione fissa e nota, in *parametri.py* è presente la variabile BEACON che contiene le posizioni delle quattro ancore.

```
# modulo parametri.py
import numpy as np

# tempo di campionamento
TC = 0.02

# varianza rumore di stato
VAR_STATE = 1

# varianza misura distanza da RSSI
VAR_RSSI = 0.1

# numero di ancore
N_BEACON = 4

# configurazione spaziale ancore [x, y, z]
BEACON = np.array([[-2.45, -2.15, -1.4],
                    [-1.9, 2.3, -1.4],
                    [1.6, 2.3, -1.4],
                    [2.4, -2.07, -1.4]])
```

In *traiettoria.py* è presente la traiettoria che seguirà il drone. È stata scelta una traiettoria di tipo curvilinea.

```
j=0
#il tempo varia da 0 a 30 sec con un passo di 0.02 (tempo di campionamento)
for tempo in np.arange(0, 30, 0.02):
    pos[j, 0] = np.cos(tempo)
    pos[j, 1] = np.sin(tempo)
    pos[j, 2] = 1 + 0.5*np.sin(tempo*2)
    j = j + 1
```

Mentre in *sistema.py* sono presenti tutte le matrici, illustrate nei precedenti capitoli, utili al calcolo del filtro di Kalman. In *Ex\_Kalman\_filter.py* saranno presenti le equazioni per l'implementazione del filtro di Kalman Esteso.

```
# modulo Ex_Kalman_Filter.py
# filtro di kalman esteso
import numpy as np
import sistema as s

def kalman( x, P, misure):

    x_pred = s.A @ x.T #X(k+1|k)

    p_new = s.A @ P @ s.A.T + s.F @ s.Q @ s.F.T #P(k+1|k)

    C = s.clineare(x_pred[0], x_pred[3], x_pred[6])

    PCT = p_new @ C.T

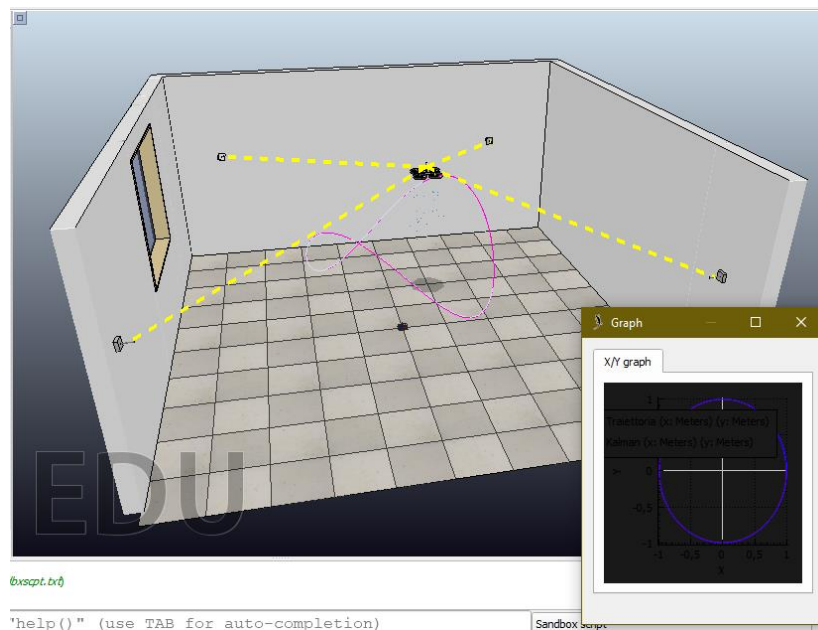
    K = PCT @ np.linalg.inv(C @ PCT + s.R) #K(k+1)
    x_pred = x_pred + K @ (misure.T - s.cnonlineare(x_pred[0], x_pred[3], x_pred[6]).T) #X(k+1)
    p_new = (np.eye(9) - K @ C) @ p_new #p(k+1)
    return x_pred, p_new
```

Nel *main.py* innanzitutto si imposta lo stato iniziale ad un punto ben preciso: il punto di partenza e si inizializza la covarianza di errore di stima  $p$  con una matrice identità  $9 \times 9$  e si procede applicando il filtro alle misure. Dopo la connessione con Vrep, inseriamo l'handle dell'oggetto quadricottero e nel ciclo for si catturano le misure e si applica il filtro di Kalman. È da notare che vengono considerate le distanze tra ogni ancora fissa e l'antenna mobile posizionata sul drone.

```
for jk in range(1):
    for i in range(0, 1500):
        pos_real = [traiettoria.pos[i, 0], traiettoria.pos[i, 1], traiettoria.pos[i, 2]]
        err = vrep.simxSetObjectPosition(clientID, handle, -1, pos_real, vrep.simx_opmode_blocking)

        distanza = s.distanze(traiettoria.pos[i, 0], traiettoria.pos[i, 1], traiettoria.pos[i, 2])
        misure = np.array([distanza[0] + np.random.normal(0, pa.VAR_RSSI),
                           distanza[1] + np.random.normal(0, pa.VAR_RSSI),
                           distanza[2] + np.random.normal(0, pa.VAR_RSSI),
                           distanza[3] + np.random.normal(0, pa.VAR_RSSI)])
        x, P = kalman.kalman(x, P, misure)
        pos_stima = [x[0], x[3], x[6]]
        err = vrep.simxSetObjectPosition(clientID, handle, -1, pos_stima, vrep.simx_opmode_blocking)
```

Di seguito, uno screenshot catturato durante la simulazione



## Riferimenti:

<https://www.youmath.it/lezioni/fisica/cinematica/2950-moto-rettilineo-uniformemente-accelerato.html>

<http://www.ce.unipr.it/people/medici/geometry/node56.html>

Alessandro Borri – Il PEKF per la stima simultanea di stato e parametri

A. Germani, C. Manes, P. Palumbo – Polynomial Extended Kalman filtering for discrete-time non linear stochastic systems.

T. Singhal, A. Harit, D. N. Vishwakarma – Kalman filter implementation on an accelerometer sensor data for three state estimation of a dynamic system.

A. Benini, A. Mancini, E. Frontoni, P. Zingaretti, S. Longhi – Adaptive extended filter for indoor/outdoor localization using a 802.15.4 wireless network.

Chenchen Zhang, Xiaoyu Bao, Qilu Wei, Qun Ma, Yuan Yang and Qing Wang - A Kalman Filter for UWB Positioning in LOS/NLOS Scenarios