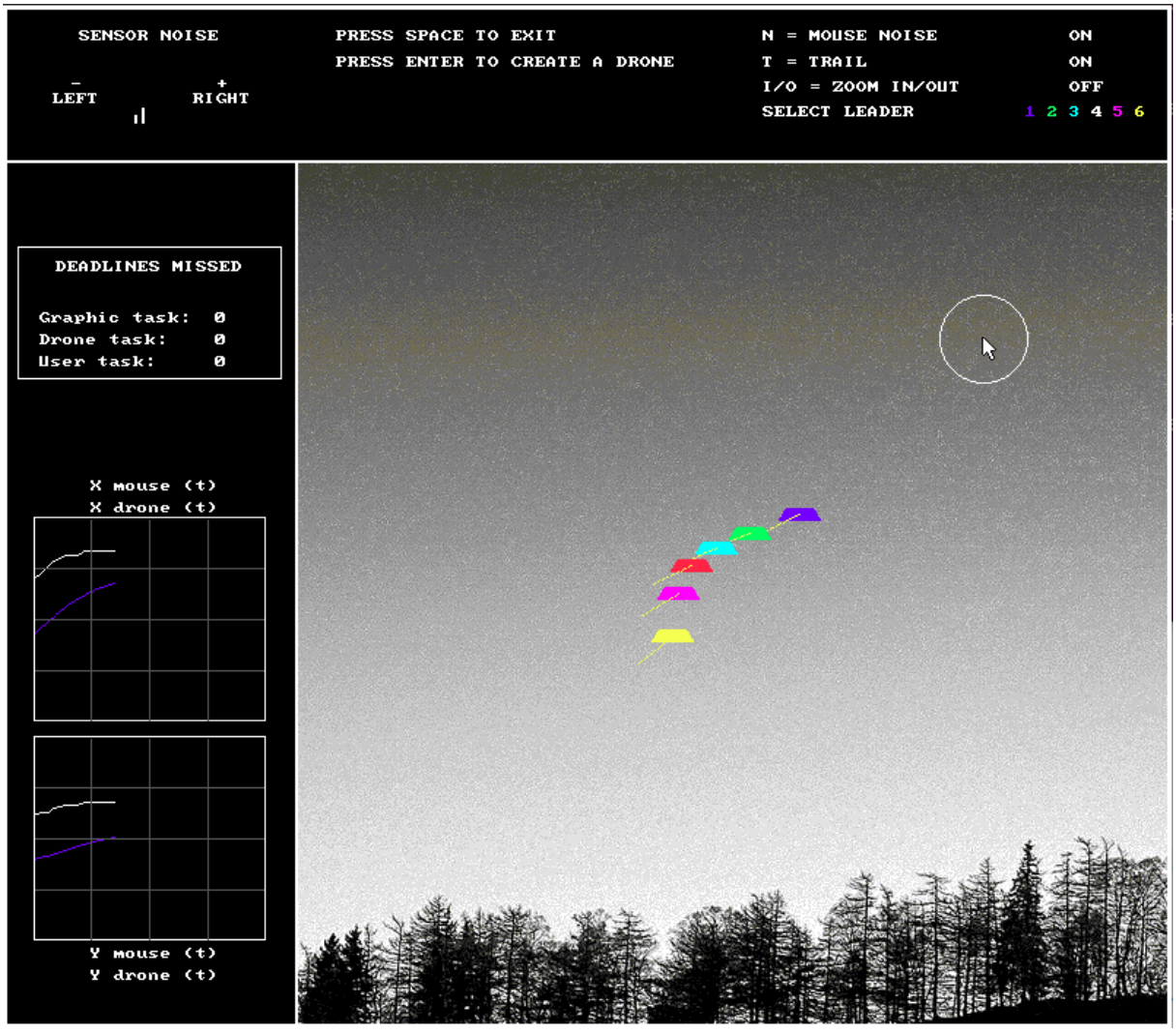


Noisy position reference tracking drones

Università di Pisa
Ingegneria Robotica e dell'Automazione
Real Time Systems

Bernacchi Lucrezia
Matricola: 603247
l.bernacchi2@studenti.unipi.it

Bezzini Riccardo
Matricola: 608798
r.bezzini1@studenti.unipi.it

Data di consegna: 17/06/2021

Indice

Descrizione del progetto

Il progetto assegnato prevede di sviluppare un'applicazione Real Time in linguaggio C sotto il sistema operativo Linux, con le librerie *pthread* e *allegro*, per simulare il volo di uno o più droni (a discrezione dell'utente) che hanno come obiettivo l'inseguimento di una posizione desiderata, individuata dalla posizione corrente del mouse. L'utente, oltre a generare più droni, ha la possibilità di aumentare o dimuinuire il rumore con cui è acquisista la misura della posizione del mouse, che in seguito è filtrata tramite un filtro di Kalman per poi essere fornita come ingresso al sistema dinamico controllato che rappresenta i droni stessi e ne simula la dinamica, fornendo in uscita la posizione effettiva di ogni drone. Sia la posizione del drone che del mouse sono graficate sulla sinistra dello schermo, insieme al contatore di deadline miss per i vari task. L'utente può inoltre rendere visibile una scia che tiene traccia dei movimenti dei droni, attivare uno zoom grafico, selezionare un drone leader e mostrare la posizione rumorosa del mouse.

Modello fisico

Oltre al modello del drone, controllato tramite un regolatore PD, sono state utilizzate per il progetto le equazioni del filtro di Kalman e quella del motore DC (implementato come filtro passa-basso tempo discreto).

2.1 Modello drone

Sono riportate in 2.1 le equazioni rappresentative della dinamica del drone, nelle quali si considera la y come variabile che indica lo spostamento verticale, mentre x si riferisce alla componente orizzontale

$$\begin{aligned} \ddot{x} &= \frac{u_2}{m} sin(\theta) \\ \ddot{y} &= -g + \frac{u_1}{m} cos(\theta) \end{aligned} \tag{2.1}$$

dove si sfrutta le variabili di controllo u_i calcolate come

$$\begin{aligned} u_1 &= m(g - k_{d,y}(\dot{y}) + k_{p,y}(y_{des} - y) \\ u_2 &= m(-k_{d,x}(\dot{x}) + k_{p,x}(x_{des} - x) \end{aligned} \tag{2.2}$$

Trattandosi di una dinamica non lineare si è passati poi alla linearizzazione, utilizzando come equilibrio la configurazione descritta in 2.3 ed ottenendo le equazioni riportate in 2.4.

$$\begin{aligned} y_0 &= x_0 = 0 \\ u_{1,0} &= mg \end{aligned} \tag{2.3}$$

$$\begin{aligned} \ddot{y} &= (-g + \frac{u_1}{m}) \\ \ddot{x} &= +\frac{u_2}{m}\theta \end{aligned} \tag{2.4}$$

A questo punto sono state riscritte le equazioni diffirenziali nel dominio della variabile s, riportando poi il procedimento seguente solo per la variabile y (tenendo comunque conto che lo stesso si è fatto per la x) di cui si ricava la fdt (tra ingresso che corrisponde alla posizione desiderata e uscita che rappresenta la posizione effettiva).

$$\begin{aligned} s^2 y &= (-g + \frac{u_2}{m}) \\ s^2 x &= +\frac{u_1}{m}\theta \\ G(s) &= \frac{k_{p,y}}{s^2 + k_{d,y}s + k_{p,y}} \end{aligned} \tag{2.5}$$

Il passaggio successivo è stata la discretizzazione: per proseguire abbiamo ricavato la banda del sistema (c.ca $\omega_b = 0.88 rad/s$, che corrispondono a $0.14 Hz$) dalla quale, applicando il teorema di Shannon (nello specifico $\omega_s = 10\omega b$), è stata calcolata la frequenza di campionamento ed il conseguente $T_s = 0.71 s$ (periodo di campionamento) utilizzato poi per l'operazione di discretizzazione. Quest'ultima è stata effettuata con il metodo della trasformata trapezoidale (Tustin) tramite il quale è stata ottenuta l'equazione alle differenze rappresentativa dell'evoluzione della posizione del drone nei vari istanti di campionamento in risposta all'ingresso considerato (in questo caso la posizione desiderata, ossia quella che sarà trasmessa tramite il mouse).

$$0.076u(k)+0.23u(k-1)+0.23u(k-2)+0.076u(k-3) = y(k)-1.14y(k-1)+0.14y(k-2) \tag{2.6}$$

Dove $u(k)$ rappresenta la coordinata desiderata, impostata come un segnale a gradino e che quindi risulta avere il valore effettivo passato dal mouse solo negli istanti in cui k assume valori maggiori di 0. La funzione (`control_drone`) che calcola i valori di spinta desiderati secondo la formula 2.7 (dove T è il tempo che intercorre fra due acquisizioni di posizione) prende in ingresso la posizione desiderata (in uscita dal filtro di Kalman) e restituisce la spinta desiderata T_des (che sarà il risultato del moto delle eliche collegate al motore).

$$T_{des} = k_d \frac{y(k) - y(k-1)}{T} + k_p(y(k) - y_{des}) \tag{2.7}$$

Questa spinta viene successivamente filtrata dall'azione del motore (implementato come filtro passa basso) per ottenere la spinta effettiva (funzione `motor_drone`):

$$T_{eff} = kT_{des} + (1-k)T_{eff\_previous} \tag{2.8}$$

Dal valore di spinta effettiva si ricava poi la coordinata desiderata risolvendo 2.7, e si fornisce come input all'equazione 2.6 (funzione `computes_coordinates`).

2.2 Filtro di Kalman

Partendo dalla formula seguente

$$X_k = K_k Z_k + (1 - K_k)X_{k-1} \tag{2.9}$$

dove

- X_k è la stima corrente della variabile di interesse
- X_{k-1} è la stima all'istante precendente
- K_k è il guadagno di Kalman
- Z_k è il valore misurato, contenente una componente rumorosa

si possono individuare due set di equazioni che consentono di compensare il fatto che la misura sia accostata ad un rumore, ossia le equazioni di Time Update (Predizione) e quelle di Measurement Update (Correction). Nel nostro caso, dato che le misure sono fornite in modo casuale dall'utente che muove il mouse ed acquisisite con una frequenza adeguata alle dinamiche di moto del drone, sono stati omessi i termini rappresentativi della

dinamica della variabile in esame, affidandosi alle doti predittive del filtro solo tramite l'aggiunta del termine di rumore di processo nella fase di *Update* della matrice P ($P_k^- = P_{k-1} + Q$) e mettendo invece in risalto la fase di correzione, le cui equazioni sono riportate in seguito:

$$\begin{aligned} K_k &= P_k^-(P_k^- + R)^{-1} \\ X_k &= X_k^- + K_k(Z_k - X_k^-) \\ P_k &= (I - K_k)P_k^- \end{aligned} \tag{2.10}$$

dove

- P_k^- è la stima a priori della covarianza dell'errore, inizializzata a P_0^-
- X_k^- è la stima a priori della variabile di interesse
- K_k è il guadagno di Kalman
- Z_k è il valore misurato, contenente una componente rumorosa
- R è la varianza dell'errore di misura, considerato gaussiano di media ϵ

Di seguito vengono riportati i valori utilizzati nel codice:

- $Q = 0.009$
- $R = 15$ + fattore dipendente dal numero del drone (è stata infatti prevista una diversità dei sensori su ciascun drone)
- P inizializzata a 0.9

Interfaccia utente

L'interfaccia utente è suddivisa in tre aree:

- Area centrale: adibita alla visualizzazione dei droni e dello scenario
- Area superiore: dedicata alla visualizzazione dei comandi disponibili all'utente
- Area laterale: adibita al monitoraggio

3.1 Area superiore - comandi

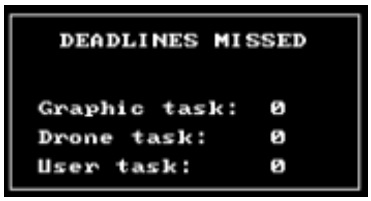
I comandi disponibili per l'utente sono i seguenti:

- `ENTER KEY`: creazione di un nuovo drone (massimo numero di droni = 6)
- `SPACE KEY`: uscita dall'applicazione
- `RIGHT/LEFT KEY`: aumento/riduzione del rumore dei sensori
- `N KEY`: visualizzazione di un cerchio d'incertezza rappresentativo della posizione rumorosa del mouse (dimensione coerente con il valore di rumore impostato)
- `T KEY`: visualizzazione della scia dei droni
- `I KEY`: attivazione della finestra di zoom (disonibili fino a 3 livelli di zoom ottenibili premendo più volte questo tasto)
- `O KEY`: disattivazione della finestra di zoom o riduzione di questo (se zoom presente a livelli 2 o 3).
- `1-6 KEY`: selezione/deselezione del drone leader. In assenza di drone leader tutti i droni seguiranno la posizione del mouse. In presenza di drone leader, solo quest'ultimo (evidenziato da un cerchio nella finestra centrale) seguirà la posizione del mouse mentre tutti gli altri droni seguiranno la posizione del drone leader.

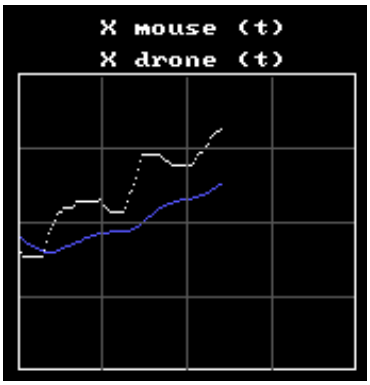
3.2 Area laterale - monitoraggio

Nell'area laterale è possibile visualizzare:

- -Il numero di deadline mancate da ciascun task
- -Oscilloscopio 1: coordinate x del mouse e del primo drone creato
- -Oscilloscopio 2: coordinate y del mouse e del primo drone creato



(a) Deadlines mancate

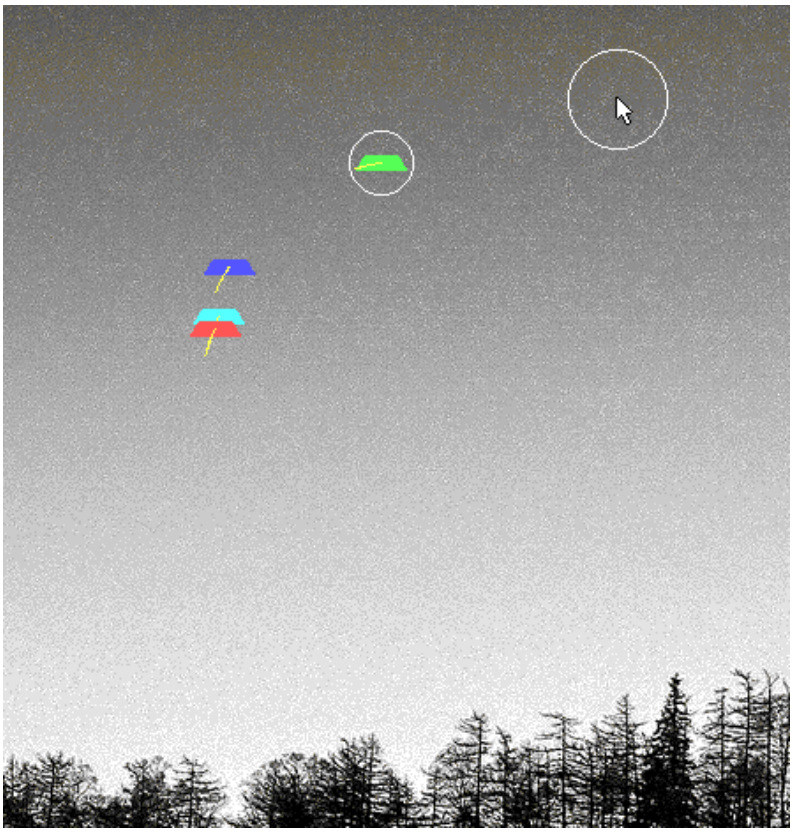


(b) Oscilloscopio 1 - andamento x

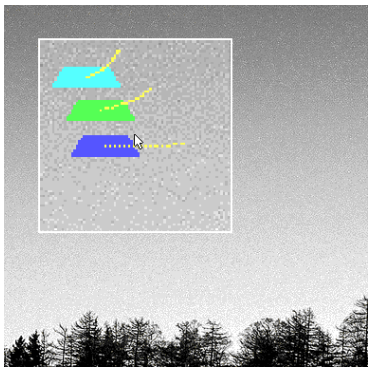
(c) Oscilloscopio 2 - andamento y

3.3 Area centrale - visualizzazione droni

Nelle figure sottostanti sono stati attivati contemporaneamente la maggior parte dei comandi in modo da poterli visualizzare. In particolare è stata attivata la scia, il rumore del mouse, lo zoom ed è stato selezionato il drone verde come drone leader.



(a) trail-leader-rumore



(b) zoom

Strutture Dati e Task

4.1 Strutture dati

Sono state implementate le seguenti strutture dati per la gestione dei droni e dei rispettivi: controllori, filtri di Kalman e motori. E' stata inoltre creata una struttura dati per la rappresentazione della scia dei droni.
Di seguito le strutture nel dettaglio:

- DRONE
  - `x_pos_now`: coordinata x attuale del drone
  - `y_pos_now`: coordinata y attuale del drone
  - `x_pos_prev`: coordinata x al passo precedente
  - `y_pos_prev`: coordinata y al passo precedente
  - `color`: colore del drone
  - `leader`: numero del drone leader
- CONTROLLER:
  - `x_des`: coordinata x desiderata
  - `y_des`: coordinata y desiderata

  Le coordinate desiderate sono le coordinate stimate dal filtro di kalman, ovvero le coordinate di riferimento per il controllore.
- KALMAN FILTER (KF):
  - `R`: rumore di misura
  - `P`: matrice di covarianza
  - `Q`: rumore di processo
- MOTOR:
  - `T_des`: spinta del motore desiderata
  - `T_eff`: spinta del motore effettiva
  - `T_eff_prev`: spinta del motore effettiva del passo precedente
- TRAIL:
  Questa struttura dati è un buffer circolare creato per conservare TL=20 posizioni precedenti dei droni al fine di creare la scia, la struttura è composta dai seguenti campi:
  - `top`: elemento corrente

– `x[TL]`: vettore che memorizza fino a 20 coordinate x precedenti

– `y[TL]`: vettore che memorizza fino a 20 coordinate y precedenti

Le variabili globali condivise tra i vari task sono protette da mutex, in particolare le strutture che prevedono una condivisione di variabili tra i task sono: DRONE, KF e TRAIL. Ciascuna di queste strutture è protetta da un mutex diverso (mux_drone, mux_kalman, mux_trail).

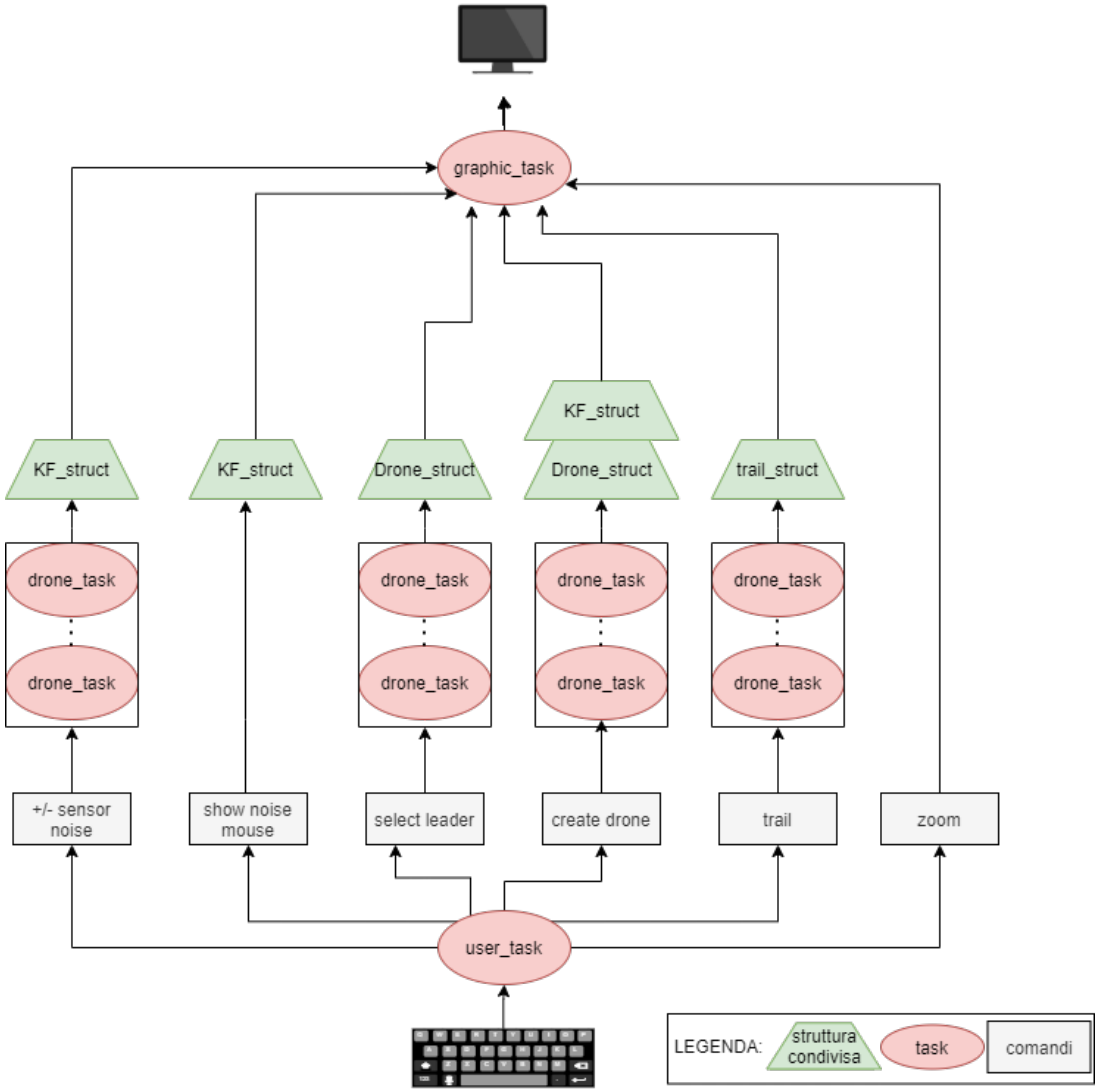
4.2 Task

Sono stati implementati tre task: `drone_task` per la parte inerente alla stima della posizione del mouse e controllo del drone, `graphic_task` per la gestione della grafica e `user_task` per la gestione dei comandi utente. Le priorità dei task, periodi e deadline relative sono state assegnate nel seguente modo:

Task	**Priorità**	**Periodo[ms]**	**Deadline rel.[ms]**
drone_task	23	30	30
graphic_task	23	50	50
user_task	20	30	30

Struttura software

La struttura del software può essere schematizzata come segue:



La cartella del progetto contiene i seguenti file:

- `rts.c`: codice principale
- `Makefile`: per la compilazione del codice
- `plib.c - plib.h`: libreria utile per la gestione dei task
- `tlib.c - tlib.h`: libreria utile per la gestione del tempo
- `foresta24bit.bmp`: sfondo

Risultati

I droni riescono a seguire il mouse (o il drone leader) con un piccolo margine di errore, l'implementazione del filtro risulta quindi essere soddisfacente; l'effetto della presenza del filtro si nota in particolare nel momento in cui la misura fornita ai droni risulta costante (mouse fermo) ed essi, con una velocità correlata al rumore presente, si avvicinano effettivamente alla posizione del mouse. Alla luce delle simulazioni fatte andando a variare il rumore di misura, si nota che all'aumentare di questo si ha un rallentamento dei droni nel raggiungere il valore desiderato, effetto atteso in conseguenza al maggior rumore presente. Inoltre è possibile notare come l'implementazione del motore conferisca un certo ritardo nei cambi di direzione dei droni rendendo quindi il modello più realistico. La funzione motore rende possibile l'analisi di prestazioni di motori DC con specifiche diverse, in quanto proprio tali specifiche sono quelle che concorrono alla generazione del valore che determina la costante che caratterizza il filtro passa basso.