

Filtro di Kalman

Antonio Lanciotti, Lorenzo D'Agostino, Arment Pelivani

2019



Rudolf E. Kalman

Indice

1	Introduzione	3
2	Cenni di teoria della probabilità	4
2.1	Variabili aleatorie o casuali.	4
2.1.1	Valore atteso	4
2.1.2	Varianza	4
2.1.3	Covarianza	5
2.2	Variabili gaussiane e modellazione dei rumori.	5
2.3	Vettori casuali	6
2.3.1	Valore atteso	6
2.3.2	Matrice di covarianza	6
3	Automatica	7
3.1	Sistemi dinamici a tempo discreto	7
3.2	Sistemi stocastici	8
4	Osservatore ottimo	9
4.1	Ottimizzazione	9
5	Equazioni del filtro	11
6	Documentazione software matlab	12
6.1	sistema.m	12
6.1.1	Proprietà	12
6.1.2	Costruttore	13
6.1.3	Evoluzione dello stato	13
6.1.4	Lecture	14
6.2	kalmanfilter.m	15
6.2.1	Proprietà	15
6.2.2	Costruttore	15
6.2.3	Evoluzione	15
6.3	Main task : filtraggio.m	16
7	Conclusione	18

1 Introduzione

Il filtro di Kalman è un osservatore ottimo dello stato per sistemi lineari in presenza di rumori gaussiani.

2 Cenni di teoria della probabilità

Il filtro di Kalman è un algoritmo che mira alla ricostruzione dello stato interno di un sistema basandosi unicamente su una serie di misurazioni che, a causa di limiti costruttivi, sono soggette a rumore.

A causa della natura del problema, risulta necessario affrontare alcuni aspetti della teoria della probabilità, in particolare ci soffermeremo sul concetto di variabile aleatoria normale, o Gaussiana, con l'intento di fornire un modello matematico per gli errori di misura che siamo costretti ad affrontare.

2.1 Variabili aleatorie o casuali.

Una variabile casuale/aleatoria è una variabile che può assumere valori diversi in dipendenza da qualche fenomeno aleatorio. In particolare diremo che una variabile casuale X si dice continua se esiste una funzione $f(x)$ definita su tutto \mathbb{R} : $P(X \in B) = \int_B f(x)dx$ dove la funzione f si dice *densità di probabilità* della variabile casuale X .

L'integrale di tale funzione nel dominio di integrazione B rappresenta pertanto la probabilità che la variabile aleatoria assuma valori appartenenti a B .

Le variabili casuali risultano essere un valido strumento matematico per la modellazione dei rumori.

Alle variabili casuali sono associati i concetti di media/valore atteso, di varianza e di covarianza.

2.1.1 Valore atteso

Nella teoria della probabilità il valore atteso di una variabile casuale X , è un numero indicato con $E[X]$ che formalizza l'idea di valore medio di un fenomeno aleatorio.

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} xf(x)dx \quad (2.1)$$

Si noti che l'operatore valore atteso è lineare:

$$E[aX + bY] = aE[X] + bE[Y] \quad (2.2)$$

2.1.2 Varianza

La varianza di una variabile aleatoria è una funzione che fornisce una misura della variabilità dei valori assunti dalla variabile stessa; nello specifico, la misura di quanto essi si discostino dal valore atteso.

La varianza della variabile aleatoria X è definita come il valore atteso del quadrato della variabile aleatoria centrata sulla propria media: $X - E[X]$:

$$Var(X) = E[(X - E[X])^2] \quad (2.3)$$

2.1.3 Covarianza

In statistica e in teoria della probabilità, la covarianza di due variabili aleatorie è un numero che fornisce una misura di quanto le due varino dipendentemente l'una dall'altra.

La covarianza di due variabili aleatorie X e Y è il valore atteso dei prodotti delle loro distanze dalla media:

$$Cov(X, Y) = E[(X - E[X])(Y - E[Y])] \quad (2.4)$$

Due variabili casuali si dicono *incorrelate* se la loro covarianza è nulla.

La covarianza può essere considerata una generalizzazione della varianza

$$Var(X) = Cov(X, X) \quad (2.5)$$

2.2 Variabili gaussiane e modellazione dei rumori.

Le variabili gaussiane sono particolari variabili aleatorie caratterizzate da due parametri, μ e σ^2 , e sono indicate tradizionalmente con:

$$X \sim N(\mu, \sigma^2) \quad (2.6)$$

Sono caratterizzate dalla funzione densità di probabilità:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.7)$$

Si può dimostrare che per le variabili gaussiane vale che:

$$E[X] = \mu \quad Var[X] = \sigma^2 \quad (2.8)$$

Come anticipato possiamo modellizzare i vettori di disturbo del sistema che consideriamo, attraverso l'utilizzo di variabili aleatorie gaussiane a media nulla e varianza σ^2 , di dimensioni conformi a quelle del sistema considerato.

2.3 Vettori casuali

Un vettore casuale è un vettore i cui elementi sono variabili casuali.

Risulta necessario estendere le definizioni date in precedenza per caratterizzare rumori che agiscono su sistemi non scalari.

2.3.1 Valore atteso

Si dice valore atteso del vettore casuale $x \in \mathbb{R}^n$ il vettore dei valori attesi delle variabili casuali che lo compongono:

$$E[x] = \begin{pmatrix} E[x_1] & E[x_2] & \dots & E[x_n] \end{pmatrix}^T \quad (2.9)$$

Si definisce *valore quadratico medio* di x come $E[x^T x]$.

2.3.2 Matrice di covarianza

Si definisce *matrice di covarianza* del vettore casuale $x \in \mathbb{R}^n$ la matrice $n \times n$:

$$Cov(x, x) = E[(x - E[x])(x - E[x])^T] \quad (2.10)$$

Per come è definita, la matrice di covarianza è una matrice simmetrica semidefinita positiva i cui elementi σ_{ij}^2 sono le covarianze tra gli elementi x_i e x_j del vettore x .

A sua volta si definisce la matrice di *cross-covarianza* tra due vettori casuali x e y , la matrice

$$Cov(x, y) = E[(x - E[x])(y - E[y])^T] \quad (2.11)$$

Due vettori x e y si dicono *incorrelati* se $Cov(x, y) = 0$.

3 Automatica

3.1 Sistemi dinamici a tempo discreto

Un sistema dinamico a tempo discreto è il modello matematico di un oggetto che interagisce con l'ambiente circostante attraverso canali di ingresso e di uscita rappresentati attraverso vettori u e y di variabili dipendenti dal tempo. Si differenziano dalla classe dei sistemi a tempo continuo dal fatto che in questo caso il tempo è rappresentato come una variabile intera $k \in \mathbb{Z}$.

Si avrà pertanto che in ogni istante di tempo k il sistema modificherà le proprie uscite sulla base dei segnali in ingresso.

Il vettore $u \in \mathbb{R}^m$ rappresenta i segnali che l'oggetto riceve in ingresso dall'esterno mentre il vettore $y \in \mathbb{R}^p$ rappresenta i segnali che l'oggetto fornisce in uscita.

In generale il comportamento del sistema non dipende esclusivamente da questi due vettori, ovvero non vi è un legame diretto tra ingresso e uscita: infatti il sistema può avere uno stato interno che evolve in funzione degli ingressi e degli stati precedenti. Lo stato di un sistema è rappresentato da un vettore $x \in \mathbb{R}^n$.

Il modello del sistema è pertanto costituito da equazioni che descrivono l'evoluzione dello stato del sistema in funzione dell'ingresso, dello stato e del tempo ed esprimono la relazione d'uscita:

$$x_{k+1} = f(x_k, u_k, k) \quad (3.1a)$$

$$y_k = g(x_k, u_k, k) \quad (3.1b)$$

dove f e g sono opportune funzioni vettoriali.

Consideriamo una particolare classe di sistemi, quelli lineari strettamente propri, in cui f e g sono funzioni lineari e l'uscita non dipende direttamente dall'ingresso ma solo dallo stato. In questo caso le equazioni generali del sistema sono:

$$x_{k+1} = A_k x_k + B_k u_k \quad (3.2a)$$

$$y_k = C_k x_k \quad (3.2b)$$

dove A, B, C sono matrici di coefficienti in generale variabili nel tempo. Se tali matrici sono costanti al variare di k il sistema si dice *tempo invariante*.

3.2 Sistemi stocastici

Il modello matematico di un sistema è un'astrazione che necessariamente deve trascurare alcuni fenomeni che sarebbero troppo complessi da descrivere. Nel caso in cui gli effetti di tali fenomeni non siano trascurabili, è possibile considerarli nel modello rappresentandoli come fenomeni stocastici, ovvero come variabili aleatorie.

Tali variabili possono anche modellizzare le incertezze nella misura delle uscite del processo, ad esempio nel caso in cui esse siano affette da rumore.

Il modello del sistema tenendo conto di tali fenomeni può essere riscritto come:

$$x_{k+1} = A_k x_k + B_k u_k + W_k w_k \quad (3.3a)$$

$$y_k = C_k x_k + v_k \quad (3.3b)$$

Le ipotesi che facciamo per caratterizzare i termini w e v sono:

- w e v sono vettori casuali gaussiani a media nulla:

$$w_k \sim \mathcal{N}(0, Q_k), \quad Q_k = Q_k^T > 0 \quad (3.4a)$$

$$v_k \sim \mathcal{N}(0, R_k), \quad R_k = R_k^T > 0 \quad (3.4b)$$

- w e v sono incorrelati:

$$E[w_{k_1} v_{k_2}^T] = 0 \quad \forall k_1, k_2 \geq 0 \quad (3.5)$$

- w e v sono bianchi:

$$E[w_{k_1} w_{k_2}^T] = 0 \quad \forall k_1 \neq k_2 \quad (3.6a)$$

$$E[v_{k_1} v_{k_2}^T] = 0 \quad \forall k_1 \neq k_2 \quad (3.6b)$$

- x_0 è un vettore casuale gaussiano con media e covarianza note:

$$x_0 \sim \mathcal{N}(\bar{x}_0, P_0), \quad P_0 = P_0^T > 0 \quad (3.7)$$

- w e v sono incorrelati con x_0 :

$$E[x_0 w_k^T] = 0 \quad \forall k \geq 0 \quad (3.8a)$$

$$E[x_0 v_k^T] = 0 \quad \forall k \geq 0 \quad (3.8b)$$

Con le ipotesi fatte è possibile passare al problema della progettazione di un osservatore che restituisca una stima dello stato interno del sistema filtrando i rumori e le incertezze sull'evoluzione dello stato e sulla misura dell'uscita.

4 Osservatore ottimo

Nella teoria del controllo, l'osservatore è un sistema dinamico che ha lo scopo di stimare lo stato di un altro sistema. L'osservatore è utile in quanto la conoscenza dell'evoluzione dello stato di un processo permette di risolvere problemi come la stabilizzazione e il controllo.

Nel caso di sistema lineare, l'osservatore può essere progettato come una copia del processo (del quale deve essere pertanto noto il modello) con l'aggiunta di un termine correttivo proporzionale alla differenza tra le uscite del processo e dell'osservatore.

Tale osservatore prende il nome di *Osservatore di Luenberger* ed ha la seguente espressione:

$$\hat{x}_{k+1} = A_k \hat{x}_k + B_k u_k + K_k (y_k - C \hat{x}_k) \quad (4.1)$$

Si considera il sistema lineare con disturbi di processo e misura (3.3), con stato iniziale $x(k_0) = x_0$ e con ingresso u_k misurabile per ogni $k \geq k_0$.

Per valutare la precisione dell'osservatore si definisce l'errore di stima, $e_k \triangleq x_k - \hat{x}_k$, il quale è regolato da un'equazione dinamica che si può ottenere valutando tale espressione per $k = k + 1$ e sostituendo sfruttando le equazioni (3.3a) e (4.1):

$$\begin{aligned} e_{k+1} &= A_k x_k + B_k u_k + W_k w_k - [A_k \hat{x}_k + B_k u_k + K_k (y_k - C \hat{x}_k)] = \\ &= (A_k - K_k C_k) e_k + W_k w_k - K_k v_k \end{aligned} \quad (4.2)$$

Osserviamo che l'errore è a sua volta un sistema stocastico, dato che la sua espressione dipende dai termini v_k e w_k , pertanto ne definiamo la matrice di covarianza all'istante $k + 1$:

$$P_{k+1} = E[e_{k+1} e_{k+1}^T] \quad (4.3)$$

Tale matrice rappresenta l'*errore quadratico medio* di stima all'istante $k + 1$.

Per procedere con l'analisi ricordiamo le ipotesi (3.4)-(3.8) fatte sui termini stocastici presenti nelle equazioni del sistema.

4.1 Ottimizzazione

L'obiettivo che ci si prefigge è quello di determinare la matrice K_k tale che la stima fornita dall'osservatore sia il più attendibile possibile, ovvero ad ogni istante $k = 0, 1, \dots$ si vuole determinare ricorsivamente il guadagno K_k dell'osservatore in modo da minimizzare l'errore quadratico medio di stima dello stato.

Sostituendo nella (4.3) la (4.2) e sfruttando le ipotesi (3.4)-(3.8) si ottiene:

$$\begin{aligned} P_{k+1} &= E \{ [(A_k - K_k C_k) e_k + W_k w_k - K_k v_k] [(A_k - K_k C_k) e_k + W_k w_k - K_k v_k]^T \} = \\ &= (A_k - K_k C_k) E[e_k e_k^T] (A_k - K_k C_k)^T + W_k E[w_k w_k^T] W_k^T + K_k E[v_k v_k^T] K_k^T = \\ &= (A_k - K_k C_k) P_k (A_k - K_k C_k)^T + W_k Q_k W_k^T + K_k R_k W_k^T \end{aligned} \quad (4.4)$$

Il problema si riduce quindi alla seguente ottimizzazione quadratica:

$$\begin{aligned}
K_k &= \underset{K}{\operatorname{argmin}} \left\{ \underbrace{(A_k - KC_k)P_k(A_k - KC_k)^T + W_k Q_k W_k^T + KR_k W_k^T}_{P_{k+1}} \right\} \\
&= \underset{K}{\operatorname{argmin}} \left\{ K \underbrace{(R_k + C_k P_k C_k^T)}_{S_k} K^T - K \underbrace{C_k P_k A_k^T}_{V_k^T} - \underbrace{A_k P_k C_k^T}_{V_k} K^T + W_k Q_k W_k^T + A_k P_k A_k^T \right\} \\
&= \underset{K}{\operatorname{argmin}} \{ K S_k K^T - K V_k^T - V_k K^T + A_k P_k A_k^T + W_k Q_k W_k^T \}
\end{aligned} \tag{4.5}$$

Essendo $S_k \triangleq R_k + C_k P_k C_k^T > R_k > 0$ la matrice S_k risulta invertibile, pertanto si può scrivere:

$$\begin{aligned}
K_k &= \underset{K}{\operatorname{argmin}} \{ (K - V_k S_k^{-1}) S_k (K - V_k S_k^{-1})^T - V_k S_k^{-1} V_k^T + A_k P_k A_k^T + W_k Q_k W_k^T \} \\
&= V_k S_k^{-1} \\
&= A_k P_k C_k^T (R_k + C_k P_k C_k^T)^{-1}
\end{aligned} \tag{4.6}$$

L'ultima espressione fornisce quindi il guadagno ottimo K_k , all'istante k , a cui corrisponde il minimo errore quadratico medio all'istante $k + 1$ dato da:

$$\begin{aligned}
P_{k+1} &= A_k P_k A_k^T - V_k S_k^{-1} V_k^T + W_k Q_k W_k^T \\
&= A_k P_k A_k^T - A_k P_k C_k^T (R_k + C_k P_k C_k^T)^{-1} C_k P_k A_k^T + W_k Q_k W_k^T
\end{aligned} \tag{4.7}$$

5 Equazioni del filtro

Il filtro di Kalman è un'implementazione ricorsiva degli algoritmi di stima che risolve il problema della ricostruzione dello stato di un sistema lineare.

Sia \hat{x}_k la stima k -esima dello stato x_k , e sia questa stima gaussiana a error medio nullo ($E[e_k] = 0$) con covarianza $P_k = E[e_k e_k^T]$.

Desiderando costruire una stima \hat{x}_{k+1} , dovremo tenere conto di due sorgenti di informazione, la conoscenza del modello (mediante l'utilizzo della legge di propagazione dello stato) e la conoscenza delle misure. Distinguiamo quindi due diverse stime di x_k , una prima \hat{x}_k^- costruita conoscendo le misure sino y_{k-1} , ed una seconda \hat{x}_k , che utilizza anche la misura y_k .

Basandoci sui risultati della sezione precedente, costruiamo quindi le stime:

$$\hat{x}_k^- = A_k \hat{x}_{k-1} + B_k u_k \quad (5.1)$$

$$P_k^- = A_k P_{k-1} A_k^T + B_k^w Q_k (B_k^w)^T \quad (5.2)$$

$$L_k = P_k^- C_k^T (C_k P_k^- C_k^T + R_k)^{-1} \quad (5.3)$$

$$\hat{x}_k = \hat{x}_k^- + L_k (y_k - C_k \hat{x}_k^-) \quad (5.4)$$

$$P_k = (I - L_k C_k) P_k^- \quad (5.5)$$

L'algoritmo così descritto prende il nome di filtro di Kalman discreto.

Il guadagno della innovazione nel filtro, L_k è fondamentalmente un rapporto tra la incertezza nella stima dello stato P e la incertezza nella misura R : conseguentemente, se le misure sono molto accurate (R piccola), la nuova stima \hat{x}_k sarà poco legata alla precedente. Se viceversa sono disponibili misure poco affidabili ma vecchie stime relativamente buone, si propagheranno queste nel futuro appoggiandosi sostanzialmente al modello del sistema.

6 Documentazione software matlab

In questo paragrafo viene descritta l'implementazione del filtro di Kalman come sistema dinamico ed il task implementato dal nostro gruppo in ambiente di programmazione *MATLAB*.

6.1 sistema.m

MATLAB presenta già una sua implementazione dei modelli dinamici ma in questo frangente si è preferita una sua nuova implementazione che includesse anche le matrici di covarianza degli errori di processo e di misura in modo da adattare più facilmente il problema alle nostre esigenze.

In particolare nel file *sistema.m* viene implementata la **classe** dei sistemi dinamici che necessitiamo.

6.1.1 Proprietà

Le proprietà di cui dispone un oggetto di questo tipo sono :

```
properties (Access = protected) % private, non modificabili.
    A,B,C,D,Q,R,x; % A,B,C,D matrici del sistema
                    % Q matrice di covarianza del rumore di processo
                    % R matrice di covarianza del rumore di misura
    n,m,p; % dimensioni rispettivamente di stato, ingresso
           e uscita
    xold; % vettore degli stati vecchi (per plot)
    u; % ULTIMO ingresso ricevuto
end
```

I metodi che implementa riguardano il costruttore dell'oggetto, l'evoluzione del suo stato interno e la lettura dell'uscita del sistema.

6.1.2 Costruttore

La creazione dell'oggetto *sistema* avviene tramite l'inizializzazione delle proprietà dell'oggetto in questione :

```
function obj = sistema(A,B,C,D,Q,R,x0)
```

Al costruttore vanno passate tutte le matrici relative al caso preso in analisi (comprese le covarianze) ed il suo stato iniziale.

Al suo interno vengono effettuati svariati controlli sulle dimensioni delle matrici (e non solo) per far sì che queste ultime rispettino le seguenti proprietà :

- A deve essere quadrata (dimensione $n \times n$);
- B deve avere n righe (dimensione $n \times m$);
- C deve avere n colonne (dimensione $p \times n$);
- D deve essere di dimensione $p \times m$;
- Q deve essere quadrata, di ordine m e definita positiva;
- R deve essere quadrata, di ordine p e semi-definita positiva;
- x0 vettore riga/colonna di dimensione n .

6.1.3 Evoluzione dello stato

In *MATLAB* nei metodi delle classi che utilizzano le proprietà delle stesse, risulta necessario passare come argomento l'oggetto corrente. Questo è possibile attraverso la parola chiave *obj*.

```
function update(obj, u) % aggiorna lo stato del sistema
    if (nargin<2)
        u = zeros(obj.m,1); % se u viene omissso si considera
                               nullo
    end
    obj.u=u;
    obj.xold(:,end+1)=obj.x; % salva il vecchio stato
    xn=obj.A*obj.x + obj.B*obj.u + obj.B*mvnrand(zeros(obj.m1),
        obj.Q)'; % calcola il nuovo stato x(t) = Ax(t-1) + Bu +
        v : v = rumore di processo
    obj.x = xn; % aggiorna lo stato con quello nuovo
end
```

La funzione accetta come parametro esterno l'ingresso dato al sistema.

Esso può essere omissso, in tal caso viene considerato nullo.

All'interno del metodo vengono inoltre aggiornate le variabili di stato del sistema.

Implementa l'equazione di stato $x(t+1) = Ax(t) + Bu(t) + v$

6.1.4 Letture

Aggiornato lo stato interno del sistema è ovviamente possibile leggerne la risposta. Questo avviene tramite il metodo *leggiUscita* che implementa l'equazione $y(t) = Cx(t) + Du(t) + w$.

Il metodo non necessita di ulteriori argomenti in ingresso:

```
function y = leggiUscita(obj) % restituisce in output l'uscita  
    del sistema
```

In più è stata implementato il metodo per la lettura dello stato interno in quanto l'accesso diretto alle proprietà del sistema è, per ragioni di sicurezza, privato all'oggetto stesso (consentito unicamente ad esso).

```
function esc = leggiStato(obj) % get dello stato per plot.
```

6.2 kalmanfilter.m

La classe *kalmanfilter* è stata pensata come un'estensione di un sistema dinamico (in particolare del sistema dinamico da osservare) con in più le matrici di guadagno caratteristiche.

Questo è implementabile attraverso il concetto di ereditarietà delle classi, infatti *kalmanfilter* eredita proprietà e metodi di *sistema*:

```
classdef kalmanfilter < sistema
```

6.2.1 Proprietà

Oltre alle proprietà caratteristiche dei sistemi dinamici, vengono introdotte le matrici di guadagno e di covarianza dello stato :

```
properties (Access = protected)
    P,K; %rispettivamente covarianza stato e guadagno
end
```

6.2.2 Costruttore

Come in *sistema.m* la classe *kalmanfilter* accetta come argomenti in ingresso le matrici relative al sistema dinamico da osservare (così da non dover costruire prima un oggetto di tipo *sistema*) ed il suo stato iniziale ed in più, opzionalmente, una stima iniziale della covarianza dello stato (P_0). Se non fornita, come stima iniziale viene considerata la matrice identità di ordine n :

```
function obj = kalmanfilter(A, B, C, D, Q, R, x0, P0)
```

All'interno del costruttore viene chiamato il costruttore della superclasse (*sistema*) al fine di inizializzare le variabili relative ad esse nell'oggetto *kalmanfilter* :

```
obj.sistema(A, B, C, D, Q, R, x0);
```

6.2.3 Evoluzione

Come per la superclasse corrispondente, la classe *kalmanfilter* avrà un metodo per il calcolo dell'evoluzione del sistema ma non sarà quello ereditato dal *sistema* da osservare ma un override di essa in quanto necessita di un metodo che implementi le equazioni di predizione, guadagno e correzione descritte nei paragrafi precedenti.

Questo avviene attraverso il metodo :

```
function update(obj, u, y) % stima lo stato
```

i cui parametri d'ingresso sono rispettivamente l'ingresso e l'uscita del sistema da osservare al tempo t .

Anche qui si necessitano dei metodi di *GET* delle variabili interne all'oggetto.

6.3 Main task : filtraggio.m

Il task che ci siamo prefissati di raggiungere è quello di ricostruire un segnale disturbato da rumore bianco. Questa applicazione risulta molto frequente in ambito ingegneristico in quanto anche i migliori trasduttori, per limiti costruttivi, presentano delle variazioni nelle misure seppur piccole.

Oltre a questo i trasduttori migliori sono reperibili ad un costo elevato, per cui si può pensare di risparmiare sulla sensoristica applicando, alle misure più rumorose di un eventuale trasduttore economico, il filtro di Kalman così da ottenere dei valori affidabili a prezzi più accessibili.

Una volta cliccato il pulsante *Run* di matlab la prima schermata permette la scelta del segnale da filtrare. A destra il layout di tale schermata con tutte le possibili scelte cliccabili. Cliccando uno dei segnali il programma provvederà alla creazione del sistema relativo alla generazione di tale segnale ed alla sua successiva discretizzazione.

```
sys = ss(A,B,C,D);  
| sysd = c2d(sys,dt); % discretizzazione  
[Ad,Bd,Cd,Dd] = ssdata(sysd);
```

Successivamente viene assegnata la matrice di covarianza del rumore di processo. Nel nostro caso il valore è scelto in maniera casuale e può essere variato permettendoci di osservarne gli effetti sulle stime :

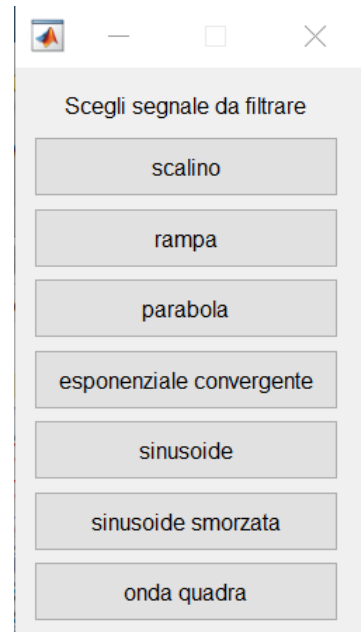
```
Q = 1e-5;
```

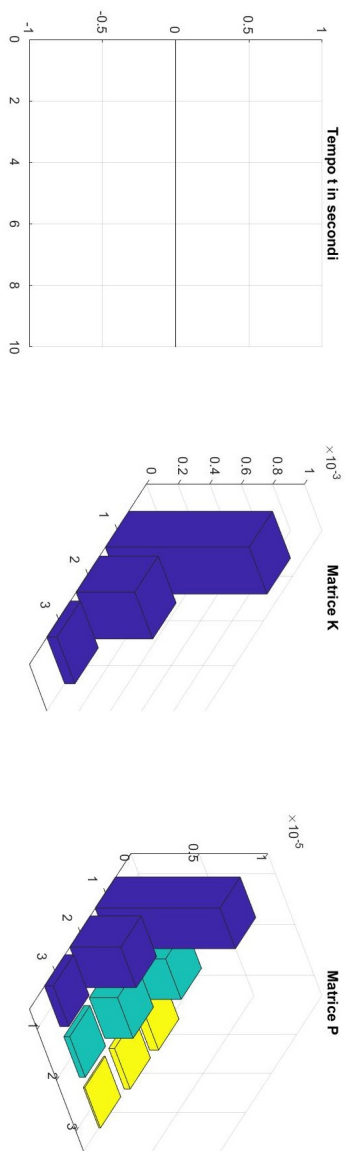
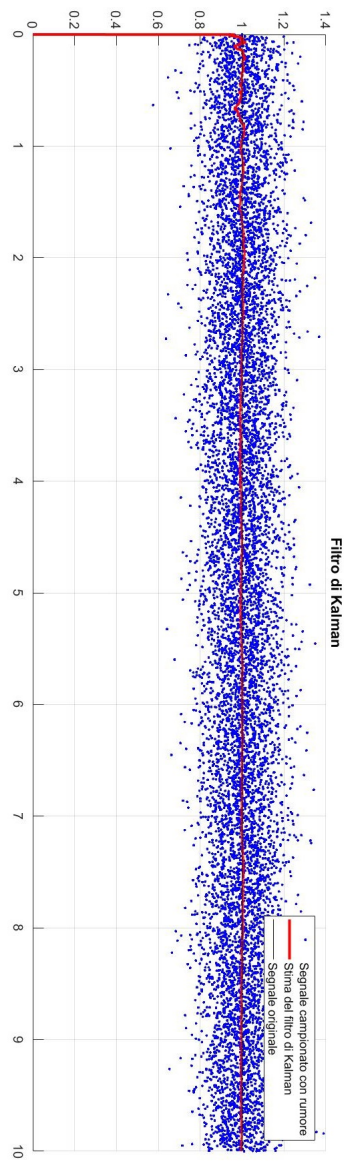
A questo punto si procede alla generazione dell'oggetto della classe `kalmanfilter`, che verrà utilizzato per la stima ed il filtraggio del segnale, utilizzando le matrici relative alla discretizzazione del modello precedente e lo stato 0 come stato iniziale.

```
k=kalmanfilter(Ad,Bd,Cd,Dd,Q,R,x0,P0);
```

A questo punto il programma procede con il calcolo, con il plotting e l'animazione della stima effettuata dal filtro, il tempo dell'animazione e la variazione delle matrici di covarianza dello stato e del guadagno.

Di seguito viene riportato un esempio del filtraggio di uno scalino affetto da rumore :





7 Conclusion

"A nonlinear differential equation of the Riccati type is derived for the covariance matrix of the optimal filtering error. The solution of this 'variance equation' completely specifies the optimal filter for either finite or infinite smoothing intervals and stationary or non-stationary statistics. The variance equation is closely related to the Hamiltonian (canonical) differential equations of the calculus of variations. Analytic solutions are available in some cases. The significance of the variance equation is illustrated by examples which duplicate, simplify, or extend earlier results in this field. The duality principle relating stochastic estimation and deterministic control problems plays an important role in the proof of theoretical results. In several examples, the estimation problem and its dual are discussed side-by-side. Properties of the variance equation are of great interest in the theory of adaptive systems. Some aspects of this are considered briefly." [1]

Riferimenti bibliografici

- [1] Rudolf E. Kalman and Richard S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, pages 95–108, 1961.