

# Filtro di Kalman

Antonio Lanciotti, Lorenzo D'Agostino, Arment Pelivani

2019-2020



Rudolf E. Kalman

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Cenni di teoria della probabilità</b>	<b>4</b>
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	Vettori casuali . . . . .	5
2.2.1	Valore atteso . . . . .	5
2.2.2	Matrice di covarianza . . . . .	5
2.3	Variabili gaussiane . . . . .	6
2.3.1	Condizionamento di variabili gaussiane . . . . .	6
2.4	Teoria della stima . . . . .	7
2.4.1	Stimatore MMSE . . . . .	7
2.4.2	Stima MMSE nel caso Gaussiano . . . . .	8
<b>3</b>	<b>Automatica</b>	<b>10</b>
3.1	Sistemi dinamici a tempo discreto . . . . .	10
3.2	Sistemi stocastici . . . . .	11
<b>4</b>	<b>Osservatore ottimo</b>	<b>12</b>
4.1	Osservatore per sistemi stocastici . . . . .	12
4.2	Ottimizzazione . . . . .	14
<b>5</b>	<b>Equazioni del filtro</b>	<b>16</b>
5.1	Filtro di Kalman nella forma correzione-predizione . . . . .	18
<b>6</b>	<b>Documentazione software MATLAB</b>	<b>19</b>
6.1	<code>sistema.m</code> . . . . .	19
6.1.1	Proprietà . . . . .	19
6.1.2	Costruttore . . . . .	20
6.1.3	Evoluzione dello stato . . . . .	20
6.1.4	Lettura dell'uscita . . . . .	21
6.2	<code>filtrokalman.m</code> . . . . .	22
6.2.1	Proprietà . . . . .	22
6.2.2	Costruttore . . . . .	22
6.2.3	Evoluzione . . . . .	23
6.2.4	Lettura stima . . . . .	23
6.3	Main task : <code>filtraggio.m</code> . . . . .	24
6.4	Modelli dei generatori di segnale . . . . .	26

# 1 Introduzione

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

La sua versatilità ed utilità lo ha portato ad innumerevoli applicazioni quali il controllo di veicoli di ogni genere ([aerospaziali](#), [navali ...](#)), [robotica e object tracking](#), ricostruzione di segnali affetti da disturbi, stima dello State Of Charge ([SOC](#)) delle batterie e molto altro. Trova anche spazio in [applicazioni finanziarie](#). In questa relazione vengono presentati i procedimenti matematici che permettono il raggiungimento delle equazioni che lo descrivono.

Verrà poi discussa un' applicazione del filtro, inerente al problema della ricostruzione di segnali affetti da rumore, realizzata in ambiente *MATLAB*.

## 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

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 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.

### 2.2.1 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:

$$E[X] = (E[X_1] \ E[X_2] \ \dots \ E[X_n])^T \quad (2.6)$$

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

### 2.2.2 Matrice di covarianza

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

$$\Sigma_X = Cov(X, X) = E[(X - E[X])(X - E[X])^T] \quad (2.7)$$

Per come è definita, la matrice di covarianza è una matrice simmetrica semidefinita positiva i cui elementi  $\Sigma_{ij}$  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

$$\Sigma_{XY} = Cov(X, Y) = E[(X - E[X])(Y - E[Y])^T] \quad (2.8)$$

Due vettori casuali  $X$  e  $Y$  si dicono *incorrelati* se  $\Sigma_{XY} = 0$ .

Dati due vettori casuali  $X$  e  $Y$  l'operazione di *coniunzione* definisce un nuovo vettore casuale  $v = \begin{pmatrix} X \\ Y \end{pmatrix}$  tale che:

$$E[v] = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix} \quad \Sigma_v = \begin{pmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{pmatrix} \quad (2.9)$$

Si definisce inoltre l'operazione di condizionamento tra i vettori  $X$  e  $Y$ : essa produce una variabile aleatoria  $X|Y$  delle stesse dimensioni di  $X$  che modella il comportamento della variabile casuale  $X$  in seguito all'osservazione del valore assunto dalla variabile  $Y$ .

## 2.3 Variabili gaussiane

Le variabili gaussiane sono particolari variabili aleatorie caratterizzate da due parametri,  $\mu$  e  $\sigma^2$ , e sono indicate tradizionalmente con:

$$X \sim N(\mu, \sigma^2) \quad (2.10)$$

Sono caratterizzate dalla funzione densità di probabilità:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.11)$$

Si può dimostrare che per le variabili gaussiane vale che:

$$E[X] = \mu \quad Var[X] = \sigma^2 \quad (2.12)$$

Possiamo definire *vettore casuale gaussiano* un vettore di variabili gaussiane. Allo stesso modo delle variabili gaussiane scalari, un vettore gaussiano  $X$  è descritto completamente dalla media  $\mu_X \in \mathbb{R}^n$  e dalla matrice di covarianza  $\Sigma_X \in \mathbb{R}^{n \times n}$ .

### 2.3.1 Condizionamento di variabili gaussiane

Siano  $X$  e  $Y$  due variabili aleatorie congiuntamente gaussiane, ovvero tali che:

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \begin{pmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{pmatrix}\right) \quad (2.13)$$

Allora il condizionamento tra le due variabili  $X|Y$  è a sua volta una variabile gaussiana. Si dimostra[?] che media e covarianza di tale variabile valgono:

$$\mu_{X|Y} = \mu_X + \Sigma_{XY} \Sigma_Y^{-1} (y - \mu_Y) \quad (2.14)$$

$$\Sigma_{X|Y} = \Sigma_X - \Sigma_{XY} \Sigma_Y^{-1} \Sigma_{YX} \quad (2.15)$$

dove  $y$  rappresenta il valore dell'osservazione della variabile  $Y$ .

## 2.4 Teoria della stima

Per stima si intende il processo di inferire il valore di una variabile casuale  $X$  di interesse dall'osservazione di un'altra variabile casuale  $Y$  che dipenda in qualche modo da  $X$ . Le variabili coinvolte sono pertanto:

- la variabile da stimare  $X \in \mathbb{R}^n$
- la variabile osservata  $Y \in \mathbb{R}^p$

Si definisce *stimatore di  $X \in \mathbb{R}^n$  basato sull'osservazione  $Y \in \mathbb{R}^p$*  una funzione  $g : \mathbb{R}^p \rightarrow \mathbb{R}^n$  operante sull'osservazione  $Y$  per produrre una stima  $\hat{X} = g(Y)$ . Dato che le variabili in gioco sono casuali, anche la stima sarà una variabile casuale. Distinguiamo la *variabile casuale stima*  $\hat{X}$  da  $\hat{x}$  che è il particolare valore della stima ottenuto dall'applicazione dello stimatore all'osservazione  $y$  effettuata.

Si definisce l'*errore di stima*  $E = X - \hat{X} = X - g(Y)$  come la differenza tra la variabile da stimare e la sua stima. Un obiettivo naturale è quello di rendere tale errore di stima "piccolo" in accordo a qualche criterio, deterministico o probabilistico, da precisare opportunamente. A tale proposito, un criterio valido potrebbe essere quello di avere un errore di stima a media nulla (*stimatore non polarizzato*):

$$E[E] = E[X - \hat{X}] = 0 \implies E[\hat{X}] = E[X]$$

e di covarianza  $E[EE^T]$  in qualche modo minima.

Come indice di prestazione per valutare la qualità di uno stimatore si introduce l'errore quadratico medio (MSE = Mean Square Error):

$$MSE_g(y) = E[(X - g(Y))^T(X - g(Y))|Y = y] \quad (2.16)$$

Poiché uno stimatore con MSE inferiore è certamente da preferirsi, viene naturale porsi il problema della determinazione dello stimatore a minimo errore quadratico medio (stimatore MMSE = *Minimum Mean Squared Error*) ovvero di uno stimatore  $g^*(Y)$  il cui MSE sia inferiore a quello di ogni altro stimatore  $g(Y)$ .

### 2.4.1 Stimatore MMSE

Lo stimatore MMSE  $g^*(\cdot)$  è dato da

$$g^*(Y) = E[X|Y] \quad (2.17)$$

e l'associato MMSE è dato da

$$\begin{aligned} & E[(X - g^*(Y))^T(X - g^*(Y))|Y] \\ &= \text{tr}(\text{var}(X|Y)) \\ &= E[X^T X|Y] - E^T[X|Y]E[X|Y] \end{aligned} \quad (2.18)$$

*Dimostrazione* - Lo stimatore MMSE deve minimizzare, rispetto a tutti gli stimatori  $g^*(\cdot)$ , il funzionale di costo

$$\begin{aligned}
V(g) &= E[(X - g(Y))^T (X - g(Y)) | Y] = \\
&= E[X^T X + g^T(Y)g(Y) - X^T g(Y) - g^T(Y)X | Y] = \\
&= E[X^T X | Y] + g^T(Y)g(Y) - E^T[X | Y]g(Y) - g^T(Y)E[X | Y] = \\
&= \underbrace{(g(Y) - E[X | Y])^T (g(Y) - E[X | Y])}_{V_1(g)} + \underbrace{E[X^T X | Y] - E^T[X | Y]E[X | Y]}_{V_2}
\end{aligned} \tag{2.19}$$

dove il primo termine  $V_1(g)$ , dipendente da  $g^*(\cdot)$ , può essere reso nullo (quindi minimo, essendo non-negativo) scegliendo  $g(Y) = g^*(Y)$  uguale alla media condizionata come in (2.17), mentre il secondo termine  $V_2$ , indipendente da  $g^*(\cdot)$ , coincide con la traccia della varianza condizionata  $tr(var(X|Y)) = tr(E[XX^T | Y] - E[X | Y]E^T[X | Y]) = E[X^T X | Y] - E^T[X | Y]E[X | Y] \geq 0$  e rappresenta il costo minimo (MMSE)  $Vg^*(\cdot)$ .

#### 2.4.2 Stima MMSE nel caso Gaussiano

Se  $X$  e  $Y$  sono congiuntamente Gaussiane (vedi (2.13)) allora lo stimatore MMSE è dato da

$$\begin{aligned}
\hat{X} &= g^*(Y) = E[X | Y] = \\
&= \Sigma_{XY} \Sigma_Y^{-1} Y + \mu_X - \Sigma_{XY} \Sigma_Y^{-1} \mu_Y
\end{aligned} \tag{2.20}$$

con relativo MMSE dato da:

$$tr(\hat{\Sigma}_X) \triangleq tr(E[(X - g^*(Y))(X - g^*(Y))^T]) = tr(\Sigma_X - \Sigma_{XY} \Sigma_Y^{-1} \Sigma_{XY}^T) \tag{2.21}$$

*Dimostrazione* - Nelle ipotesi fatte, sfruttando le formule della media e covarianza condizionata di variabili aleatorie Gaussiane, si ha

$$\begin{aligned}
\hat{X} &= g^*(Y) = E[X | Y] = \\
&= \mu_X + \Sigma_{XY} \Sigma_Y^{-1} (Y - \mu_Y) = \\
&= \Sigma_{XY} \Sigma_Y^{-1} Y + [\mu_X - \Sigma_{XY} \Sigma_Y^{-1} \mu_Y]
\end{aligned} \tag{2.22}$$

Inoltre,

$$\hat{\Sigma}_X = var(X | Y) = \Sigma_X - \Sigma_{XY} \Sigma_Y^{-1} \Sigma_{XY}^T \tag{2.23}$$

come volevasi dimostrare.

Si può osservare come nel caso gaussiano lo stimatore MMSE sia a tutti gli effetti uno stimatore affine, ovvero della forma  $g(Y) = AY + b$ , dove:

$$A = \Sigma_{XY} \Sigma_Y^{-1} \tag{2.24}$$

$$b = \mu_X - A\mu_Y = \mu_X - \Sigma_{XY} \Sigma_Y^{-1} \mu_Y \tag{2.25}$$



In generale lo stimatore MMSE non è affine, infatti esiste il problema della ricerca del miglior stimatore affine detto stimatore BLUE (*Best Linear Unbiased Estimator* con una imprecisione terminologica essendo lo stimatore cercato affine e non lineare) che è meno preciso dello stimatore MMSE. Nel caso gaussiano tuttavia essi coincidono, ovvero lo stimatore MMSE appena determinato è esso stesso lo stimatore BLUE per  $X$  sulla base delle osservazioni di  $Y$ .

## 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 di ogni genere, (dai [sistemi industriali](#) al [controllo del traffico](#)).

L'osservatore più utilizzato nel caso di sistemi lineari prende il nome di *Osservatore di Luenberger*[?] ed ha la seguente espressione:

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

Ad ogni istante di tempo, tale sistema calcola la nuova stima dello stato a partire dalla precedente sfruttando il modello noto del processo e le informazioni date dalle quantità conosciute  $u_k$  e  $y_k$ , ovvero ingresso e uscita del processo.

In particolare il termine  $A_k \hat{x}_k + B_k u_k$  altro non è che l'applicazione dell'equazione di aggiornamento dello stato (3.2a) alla stima calcolata nel passo precedente sfruttando la conoscenza dell'ingresso, mentre il termine  $K_k (y_k - C_k \hat{x}_k)$  è la correzione che viene fatta sulla base della differenza tra l'uscita del processo  $y_k$  e quella stimata dall'osservatore  $C_k \hat{x}_k$ .

Il fattore  $K_k$  è il tassello essenziale per il corretto funzionamento dell'osservatore, dato che va a determinarne il comportamento (stabilità e velocità di convergenza).

In particolare, si può riscrivere l'equazione dell'osservatore nella forma:

$$\hat{x}_{k+1} = (A_k - K_k C_k) \hat{x}_k + B_k u_k + K_k y_k \quad (4.2)$$

Si osserva come la dinamica dell'osservatore sia determinata dalla matrice  $A_k - K_k C_k$ , pertanto la matrice  $K_k$  va scelta in modo opportuno, ad esempio in modo tale che garantisca la stabilità del sistema, ovvero che  $A_k - K_k C_k$  abbia autovalori  $\lambda$  tali che  $\lambda < 1$  (nel caso di tempo discreto). Tale specifica può essere soddisfatta se e solo se il processo risulta *osservabile*.

Nel caso di sistemi lineari stocastici, come si vedrà in seguito, il problema della determinazione della matrice  $K_k$  è affrontato in modo da minimizzare l'errore quadratico medio di stima.

La scelta della stima  $\hat{x}_0$  non è critica, dato che se l'osservatore è progettato correttamente gli effetti di una stima iniziale poco precisa si estinguono in un tempo relativamente breve grazie all'effetto del termine correttivo precedentemente descritto.

### 4.1 Osservatore per sistemi stocastici

Si considera il sistema lineare con disturbi di processo e misura (3.3), con stato iniziale  $x_0$  e con ingresso  $u_k$  misurabile per ogni  $k \geq k_0$  e costruiamo l'osservatore come in (4.1).

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 all'istante  $k + 1$  e sfruttando le equazioni (3.3a) e (4.1):

$$\begin{aligned} e_{k+1} &= A_k x_k + B_k u_k + W_k w_k - [A_k \hat{x}_k + B_k u_k + K_k (y_k - C \hat{x}_k)] = \\ &= (A_k - K_k C_k) e_k + W_k w_k - K_k v_k \end{aligned} \quad (4.3)$$

Tale errore, essendo definito sulla base dello stato  $x_k$  del processo non è noto, tuttavia la sua definizione ci permetterà di caratterizzarlo dal punto di vista statistico, infatti 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  $k + 1$ :

$$P_{k+1} = E[e_{k+1} e_{k+1}^T] \quad (4.4)$$

La traccia di 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. In particolare, sfruttiamo le ipotesi di incorrelazione tra i rumori e lo stato iniziale fatte in precedenza per dimostrare che anche l'errore di stima è incorrelato con i rumori. Dalla teoria dei sistemi sappiamo che, risolvendo l'equazione alle differenze (4.3) l'errore di stima  $e_k$  è una combinazione lineare dell'errore iniziale  $e_0 = x_0 - \hat{x}_0$  e dei rumori  $w_i$  e  $v_i$  per  $i = 0, 1, \dots, k - 1$ :

$$e_k = \Phi_k e_0 + \sum_{i=0}^{k-1} \Phi_{k-i} W_i w_i - \sum_{j=0}^{k-1} \Phi_{k-j} K_j v_j \quad (4.5)$$

dove  $\Phi_k = \prod_{i=0}^{k-1} A_i - K_i C_i$ .

Utilizzando tale espressione possiamo dimostrare l'incorrelazione tra l'errore di stima e i rumori di processo e misura.

$$\begin{aligned} E[e_k w_k^T] &= E[(\Phi_k e_0 + \sum_{i=0}^{k-1} \Phi_{k-i} W_i w_i - \sum_{j=0}^{k-1} \Phi_{k-j} K_j v_j) w_k^T] = \\ &= \Phi_k E[e_0 w_k^T] + \sum_{i=0}^{k-1} \Phi_{k-i} W_i E[w_i w_k^T] - \sum_{j=0}^{k-1} \Phi_{k-j} K_j E[v_j w_k^T] = \\ &= \Phi_k E[x_0 w_k^T] - \Phi_k \hat{x}_0 E[w_k^T] = 0 \end{aligned} \quad (4.6)$$

Allo stesso modo si dimostra che anche  $E[e_k v_k^T] = 0$ .

## 4.2 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 (4.4) la (4.3) e sfruttando le ipotesi (3.4)-(3.8) e il risultato ottenuto alla fine del precedente paragrafo si ottiene:

$$\begin{aligned} P_{k+1} &= E \left\{ [(A_k - K_k C_k) e_k + W_k w_k - K_k v_k] [(A_k - K_k C_k) e_k + W_k w_k - K_k v_k]^T \right\} = \\ &= (A_k - K_k C_k) E[e_k e_k^T] (A_k - K_k C_k)^T + W_k E[w_k w_k^T] W_k^T + K_k E[v_k v_k^T] K_k^T = \\ &= (A_k - K_k C_k) P_k (A_k - K_k C_k)^T + W_k Q_k W_k^T + K_k R_k W_k^T \end{aligned} \quad (4.7)$$

Il problema si riduce quindi alla seguente ottimizzazione quadratica:

$$\begin{aligned} K_k &= \underset{K}{\operatorname{argmin}} \{ \operatorname{tr} [P_{k+1}] \} = \\ &= \underset{K}{\operatorname{argmin}} \left\{ \operatorname{tr} \left[ (A_k - K C_k) P_k (A_k - K C_k)^T + W_k Q_k W_k^T + K R_k W_k^T \right] \right\} \\ &= \underset{K}{\operatorname{argmin}} \left\{ \operatorname{tr} \left[ K \underbrace{(R_k + C_k P_k C_k^T)}_{S_k} K^T - K \underbrace{C_k P_k A_k^T}_{V_k^T} - \underbrace{A_k P_k C_k^T}_{V_k} K^T + W_k Q_k W_k^T + A_k P_k A_k^T \right] \right\} \\ &= \underset{K}{\operatorname{argmin}} \left\{ \operatorname{tr} [K S_k K^T - K V_k^T - V_k K^T + A_k P_k A_k^T + W_k Q_k W_k^T] \right\} \end{aligned} \quad (4.8)$$

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:

$$\begin{aligned} K_k &= \underset{K}{\operatorname{argmin}} \left\{ \operatorname{tr} [(K - V_k S_k^{-1}) S_k (K - V_k S_k^{-1})^T - V_k S_k^{-1} V_k^T + A_k P_k A_k^T + W_k Q_k W_k^T] \right\} \\ &= V_k S_k^{-1} \\ &= A_k P_k C_k^T (R_k + C_k P_k C_k)^{-1} \end{aligned} \quad (4.9)$$

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:

$$\begin{aligned} \operatorname{tr}[P_{k+1}] &= \operatorname{tr}[A_k P_k A_k^T - V_k S_k^{-1} V_k^T + W_k Q_k W_k^T] \\ &= \operatorname{tr}[A_k P_k A_k^T - A_k P_k C_k^T (R_k + C_k P_k C_k)^{-1} C_k P_k A_k^T + W_k Q_k W_k^T] \end{aligned} \quad (4.10)$$

In definitiva si ha il seguente risultato:

dato il sistema LTV (3.3) che soddisfa le ipotesi (3.4)-(3.8), l'osservatore di

Luenberger che minimizza ad ogni istante  $k \geq 0$  l'errore quadratico medio di stima dello stato  $P_k \triangleq E[(x_k - \hat{x}_k)^T(x_k - \hat{x}_k)]$  è fornito dal seguente algoritmo ricorsivo (che procede per  $k = 0, 1, \dots$ ):

$$K_k = A_k P_k C_k^T (R_k + C_k P_k C_k)^{-1} \quad (4.11)$$

$$\hat{x}_{k+1} = A_k \hat{x}_k + B_k u_k + K_k (y_k - C \hat{x}_k) \quad (4.12)$$

$$P_{k+1} = A_k P_k A_k^T - A_k P_k C_k^T (R_k + C_k P_k C_k^T)^{-1} C_k P_k A_k^T + W_k Q_k W_k^T \quad (4.13)$$

## 5 Equazioni del filtro

Come osservato nel precedente paragrafo, l'algoritmo (4.11)-(4.13) propaga una stima  $\hat{x}_k$  e la relativa covarianza  $P_k$  dello stato  $x_k$  basata sulle osservazioni precedenti  $y_{0:k-1}$ , ovvero effettua la predizione ad un passo dello stato. Per questo motivo, nel seguito le quantità  $\hat{x}_k$  e  $P_k$  in (4.11)-(4.13) verranno indicate in modo più appropriato come  $\hat{x}_{k|k-1} = \hat{x}_k$ , e rispettivamente,  $P_{k|k-1} = P_k$ . In particolare come abbiamo visto la matrice di covarianza  $P_k$  ci permette di valutare la precisione della stima  $\hat{x}_k$ .

In molte applicazioni pratiche tuttavia l'obiettivo è quello di ottenere, ad ogni istante  $k$ , una stima dello stato  $x_k$ , e la relativa covarianza, basata su tutte le osservazioni  $y_{0:k} = y_{0:k-1} \cup y_k$ , inclusa quella presente, acquisite fino all'istante  $k$ .

Il problema in oggetto, noto come filtraggio, è finalizzato alla determinazione della cosiddetta stima filtrata  $\hat{x}_{k|k}$  dello stato  $x_k$  basata sulle osservazioni  $y_{0:k}$  e della covarianza dell'errore di stima  $P_{k|k} = E[e_{k|k}^x (e_{k|k}^x)^T]$ , con  $e_{k|k}^x \triangleq x_k - \hat{x}_{k|k}$ .

A tale scopo, si possono considerare  $\hat{x}_{k|k}$  e  $P_{k|k}$  come stima e covarianza a-posteriori della variabile aleatoria  $x_k$ , a partire da stima e covarianza a priori  $\hat{x}_{k|k-1}$  e  $P_{k|k-1}$ , sulla base dell'osservazione  $y_k$  in (3.3b), utilizzando il metodo di stima BLUE.

In altri termini, date le statistiche a priori  $(\hat{x}_{k|k-1}, P_{k|k-1})$  e l'osservazione lineare  $y_k = C_k x_k + v_k$  di  $x_k$  si vogliono determinare le statistiche a posteriori  $(\hat{x}_{k|k}, P_{k|k})$  di  $x_k$  mediante stima BLUE. In particolare le statistiche a priori sono interpretate come valori attesi di quelle a posteriori. Pertanto stima e covarianza a posteriori BLUE sono fornite da:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \underbrace{E[e_{k|k-1}^x (e_{k|k-1}^y)^T] E[e_{k|k-1}^y (e_{k|k-1}^y)^T]^{-1}}_{L_k} (y_k - \hat{y}_{k|k-1}) \quad (5.1)$$

$$P_{k|k} = P_{k|k-1} - E[e_{k|k-1}^x (e_{k|k-1}^y)^T] E[e_{k|k-1}^y (e_{k|k-1}^y)^T]^{-1} E[e_{k|k-1}^x (e_{k|k-1}^y)^T]^T \quad (5.2)$$

dove

$$\begin{aligned} \hat{y}_{k|k-1} &= E[y_k] = C_k \hat{x}_{k|k-1} \\ e_{k|k-1}^y &= y_k - \hat{y}_{k|k-1} = C_k e_{k|k-1}^x + v_k \\ E[e_{k|k-1}^x (e_{k|k-1}^y)^T] &= P_{k|k-1} C_k^T \\ E[e_{k|k-1}^y (e_{k|k-1}^y)^T] &= R_k + C_k P_{k|k-1} C_k^T \end{aligned} \quad (5.3)$$

Sostituendo le (5.3) in (5.1)-(5.2) si ottengono le seguenti equazioni di aggiornamento da  $\hat{x}_{k|k-1}, P_{k|k-1}$  a  $\hat{x}_{k|k}, P_{k|k}$ :

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L_k (y_k - C_k \hat{x}_{k|k-1}) \quad (5.4)$$

$$P_{k|k} = P_{k|k-1} - P_{k|k-1} C_k^T (R_k + C_k P_{k|k-1} C_k^T)^{-1} C_k P_{k|k-1} \quad (5.5)$$

con

$$L_k = P_{k|k-1} C_k^T (R_k + C_k P_{k|k-1} C_k^T)^{-1} \quad (5.6)$$



Le equazioni (5.4)-(5.5) consentono di correggere la stima predittiva  $\hat{x}_{k|k-1}$  e la relativa covarianza  $P_{k|k-1}$  con l'ultima osservazione  $y_k$ , per ottenere la stima filtrata  $\hat{x}_{k|k}$  e relativa covarianza  $P_{k|k}$ .

Per questo motivo (5.4)-(5.5) vengono dette *equazioni di correzione* della stima e della covarianza, e la matrice  $L_k$  in (5.6) prende il nome di *guadagno di correzione*.

Si noti che, posto  $K_k = A_k L_k$ , l'equazione di aggiornamento della stima (4.12) può essere espressa come segue:

$$\hat{x}_{k+1|k} = A_k \underbrace{[\hat{x}_{k|k-1} + L_k(y_k - C_k \hat{x}_{k|k-1})]}_{\hat{x}_{k|k}} + B_k u_k \quad (5.7)$$

La precedente relazione esprime la stima predittiva ad un passo  $\hat{x}_{k+1|k}$  come il risultato dell'applicazione del modello di stato (3.3a), per  $w_k = 0$ , alla stima filtrata  $\hat{x}_{k|k}$ . Analogamente, l'equazione di aggiornamento della covarianza (4.13) può essere riscritta come:

$$P_{k+1|k} = A_k \underbrace{[P_{k|k-1} - P_{k|k-1} C_k^T (R_k + C_k P_{k|k-1} C_k^T)^{-1} C_k P_{k|k-1}]}_{P_{k|k}} A_k^T + W_k Q_k W_k^T \quad (5.8)$$

Pertanto, si ottengono le seguenti equazioni di predizione da  $(\hat{x}_{k|k}, P_{k|k})$  a  $(\hat{x}_{k+1|k}, P_{k+1|k})$ :

$$\hat{x}_{k+1|k} = A_k \hat{x}_{k|k} + B_k u_k \quad (5.9)$$

$$P_{k+1|k} = A_k P_{k|k} A_k^T + W_k Q_k W_k^T \quad (5.10)$$

Riassumendo i precedenti sviluppi, la ricorsione (4.12)-(4.13) del filtro di Kalman può essere suddivisa in due fasi diverse: la correzione (5.4)-(5.5) seguita dalla predizione (5.9)-(5.10). La forma correzione-predizione del filtro di Kalman viene riportata nella pagina seguente.

## 5.1 Filtro di Kalman nella forma correzione-predizione

**Dati:**

- matrici :  $A_k, B_k, C_k, W_k, Q_k, R_k$ ;
- stima iniziale :  $\hat{x}_{1|0}$ ;
- covarianza iniziale :  $P_{1|0}$ ;
- $k = 1, 2, \dots$

**Correzione:**

$$\begin{aligned}
 S_k &= R_k + C_k P_{k|k-1} C_k^T && \text{covarianza dell'innovazione} \\
 L_k &= P_{k|k-1} C_k^T S_k^{-1} && \text{guadagno di correzione} \\
 e_k &= y_k - C_k \hat{x}_{k|k-1} && \text{innovazione} \\
 \hat{x}_{k|k} &= \hat{x}_{k|k-1} + L_k e_k && \text{correzione della stima} \\
 P_{k|k} &= P_{k|k-1} - L_k S_k L_k^T && \text{correzione della covarianza} \\
 &= (I - L_k C_k) P_{k|k-1} (I - L_k C_k)^T + L_k R_k L_k^T
 \end{aligned}$$

**Predizione:**

$$\begin{aligned}
 \hat{x}_{k+1|k} &= A_k \hat{x}_{k|k} + b_k && \text{predizione della stima} \\
 P_{k+1|k} &= A_k P_{k|k} A_k^T + W_k Q_k W_k^T && \text{predizione della covarianza}
 \end{aligned}$$

## 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*, usando l'approccio della programmazione orientata agli oggetti.

### 6.1 sistema.m

*MATLAB* presenta già un'implementazione dei modelli di sistemi dinamici lineari ma si è preferito realizzarne una nuova implementazione che considerasse anche gli errori di processo e di misura in modo da rispettare le ipotesi del problema.

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

Per semplicità abbiamo implementato soltanto sistemi tempo invarianti, per cui tutte le matrici che definiscono il sistema sono costanti.

#### 6.1.1 Proprietà

Le proprietà di cui dispongono gli oggetti di questa classe sono :

```
classdef sistema < handle
%SISTEMA
%Classe che descrive un sistema dinamico lineare tempo
    invariante

    properties (Access = protected)
        A,B,C,D,W,Q,R,x;    %A,B,C,D matrici del sistema
        % W matrice di guadagno del rumore di processo
        % Q matrice di covarianza del rumore di processo
        % R matrice di covarianza del rumore di misura
        n,m,p,q;            %n dim stato, m dim ingresso, p dim uscita, q
                               dim rumore di processo
        u;                    %ultimo ingresso ricevuto
    end
```

I metodi implementati sono il costruttore dell'oggetto che va ad inizializzarlo e le due equazioni di evoluzione dello stato interno e di uscita.

### 6.1.2 Costruttore

La creazione dell'oggetto **sistema** avviene tramite l'inizializzazione dei suoi parametri:

```
function obj = sistema(A,B,C,D,W,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 tutti i controlli necessari a verificare che i parametri 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  $p \times m$ ;
- $W$  deve avere  $n$  righe (dimensione  $n \times q$ );
- $Q$  deve essere quadrata (dimensione  $q \times q$ ) e definita positiva;
- $R$  deve essere quadrata (dimensione  $p \times p$ ) e definita positiva;
- $x0$  vettore di lunghezza  $n$ .

### 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; % salva l'ultimo ingresso ricevuto
    xn = obj.A*obj.x+obj.B*obj.u+obj.W*mvnrnd(zeros(obj.q,1),obj.Q,1);
    % calcola il nuovo stato x(k+1) = Ax(k) + Bu(k) + Ww(k) : w = 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.

Implementa l'equazione di stato  $x_{k+1} = Ax_k + Bu_k + Ww_k$  andando ad aggiornare la variabile di stato  $x$  dell'oggetto.

#### 6.1.4 Lettura dell'uscita

Il metodo *leggiUscita* implementa l'equazione  $y(t) = Cx(t) + Du(t) + w$  restituendo in output il valore di  $y$ .

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 integrità, consentito unicamente all'oggetto stesso. La lettura dello stato del sistema non sarebbe possibile nella realtà, infatti tale metodo viene utilizzato solo per monitorare il comportamento del sistema, tali dati non verranno utilizzati direttamente.

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

## 6.2 filtrokalman.m

La classe `filtrokalman` è stata implementata come un'estensione della precedente classe `sistema`, infatti il *filtro di Kalman* essendo un osservatore dello stato è a sua volta un sistema dinamico.

Tale estensione si realizza attraverso il concetto di ereditarietà delle classi, infatti `kalmanfilter` eredita proprietà e metodi di `sistema` e ciò si indica attraverso il simbolo `<` :

```
classdef kalmanfilter < sistema
```

### 6.2.1 Proprietà

Oltre alle proprietà della classe `sistema` da essa ereditate, vengono introdotte la matrice di guadagno, la matrice di covarianza dello stato corretto e la predizione del prossimo stato e della relativa covarianza:

```
properties %(Access = protected)
    L;          % matrice guadagno di Kalman
    P;          % matrice di covarianza dello stato corretto
    xPr, PPr;   % predizione dello stato e relativa covarianza
end
```

### 6.2.2 Costruttore

Come in `sistema.m` la classe `filtrokalman` accetta come argomenti in ingresso le matrici relative al modello del sistema da osservare e la stima iniziale dello stato (`x0`) con la relativa covarianza (`P0`); se quest'ultima è omessa viene considerata come valore di default la matrice identità di ordine  $n$ :

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

All'interno del costruttore viene richiamato il costruttore della superclasse `sistema` al fine di inizializzare le variabili relative al modello nell'oggetto `filtrokalman`:

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

Inoltre, viene verificato che stima iniziale e covarianza siano valide.

### 6.2.3 Evoluzione

Come per la superclasse corrispondente, la classe *filtrokalman* ha un metodo per il calcolo dell'evoluzione del sistema. Il metodo ereditato dalla classe **sistema** viene sovrascritto (*override*) in modo da implementare l'algoritmo ricorsivo di stima descritto nel capitolo precedente:

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

    %calcolo guadagno di Kalman
    obj.L = obj.PPr*obj.C'/(obj.C*obj.PPr*obj.C'+obj.R);

    %correzione
    obj.x = obj.xPr+obj.L*(y-obj.C*obj.xPr);
    I_LC = (eye(obj.n)-obj.L*obj.C);
    obj.P = I_LC*obj.PPr*I_LC'+obj.L*obj.R*obj.L';

    %predizione
    obj.xPr = obj.A*obj.x + obj.B*u;
    obj.PPr = obj.A*obj.P*obj.A'+obj.W*obj.Q*obj.W';
end
```

I parametri d'ingresso, oltre al riferimento all'oggetto, sono rispettivamente l'ingresso e l'uscita del sistema da osservare al tempo  $k$ .

### 6.2.4 Lettura stima

Una volta effettuato l'aggiornamento del filtro, per ottenere il valore della stima dello stato calcolata si utilizza il metodo **leggiStima** che restituisce il vettore dello stato stimato.

```
function x = leggiStima(obj)
    x = obj.x;
end
```

Se si è interessati al filtraggio dell'uscita del sistema, si può utilizzare il metodo **leggiUscitaStimata** che restituisce l'uscita del sistema calcolata sulla base dello stato stimato.

Entrambi i metodi non necessitano di alcun parametro esterno.

```
function y = leggiUscitaStimata(obj)
    y = obj.C*obj.x + obj.D*obj.u;
end
```

Sono stati implementati anche i metodi **leggiL** e **leggiP** che permettono di ottenere le matrici  $L$  e  $P$  del filtro all'istante corrente. Ciò sarà utile per osservarne l'evoluzione attraverso le funzioni grafiche di *MATLAB*.

```
function L = leggiL(obj)
    L = obj.L;
end
function P = leggiP(obj)
    P = obj.P;
end
```

### 6.3 Main task : filtraggio.m

Il task che ci siamo prefissati di raggiungere è quello di ricostruire un segnale disturbato da rumore bianco gaussiano. Questa applicazione risulta molto utile 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 soltanto ad un costo elevato, per cui si può pensare in certe condizioni 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.

All'inizio dello script vengono definiti il tempo di campionamento e la durata della simulazione in secondi.

Viene poi visualizzato un menu che permette di scegliere la natura del segnale da filtrare; i segnali possibili sono tutti e soli quelli ottenibili come uscite da sistemi lineari.

I modelli dei generatori di segnale utilizzati vengono riportati nella sezione successiva. Cliccando uno dei segnali il programma provvederà alla creazione del modello del generatore ed alla sua successiva discretizzazione attraverso le funzioni *built-in* di *MATLAB*.

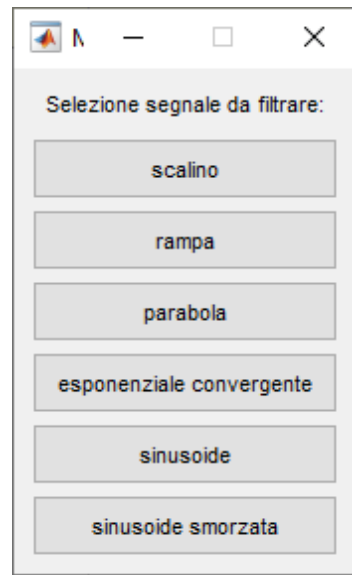
```
sys = ss(A,B,C,D);  
sysd = c2d(sys,dt);  
[Ad,Bd,Cd,Dd] = ssdata(sysd);
```

Successivamente vengono definite le matrici di covarianza dei rumori di processo e misura. I valori di tali matrici possono essere variati per aumentare o diminuire la rumorosità del segnale da filtrare.

```
Q=1e-3;  
R=1e-1*eye(p);
```

Vengono a questo punto inizializzati gli oggetti relativi al generatore di segnale e al filtro di Kalman e si definisce il ciclo che esegue la simulazione.

```
% inizializzazione sistema generatore di segnale rumoroso  
sys=sistema(Ad,Bd,Cd,Dd,W,Q,R,x0);  
  
% inizializzazione filtro di kalman  
P0=eye(n);  
k=filtrokalman(Ad,Bd,Cd,Dd,W,Q,R,zeros(n,1),P0);
```



Menu di scelta dei segnali



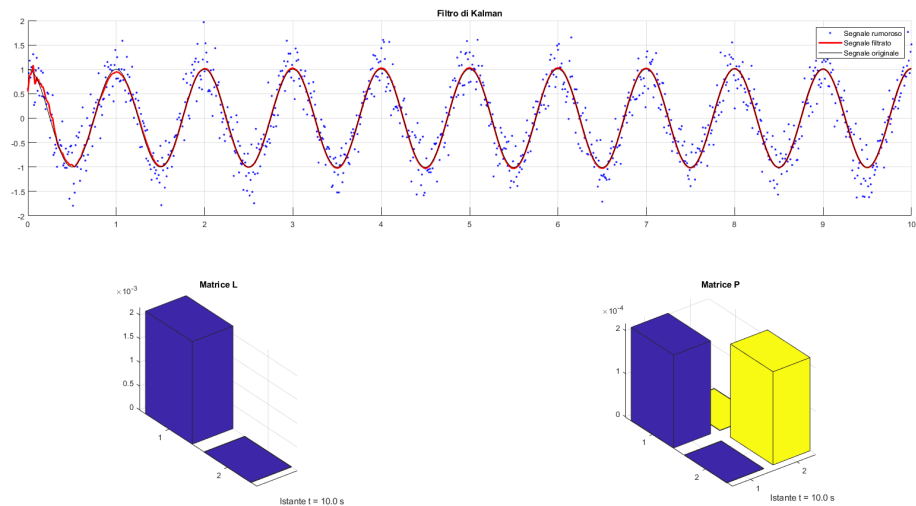
```

%% simulazione
for i=1:length(t)
    x(:,i)=sys.leggiStato(); % lettura stato sistema (segnale
                             da ricostruire, per plot)
    y(:,i)=sys.leggiUscita(); % lettura uscita sistema (segnale
                             rumoroso)
    sys.update(); % calcolo del nuovo stato del
                  sistema
    k.update(0,y(:,i)); % aggiornamento filtro -->
                        parametri u=0 e y(segnale rumoroso)
    xs(:,i)=k.leggiStima(); % lettura stima kalman
    Lplot(:,i)=k.leggiL(); % lettura matrice L per
                            animazione
    Pplot(:,i)=k.leggiP(); % lettura matrice P per
                            animazione
end

```

Una volta completata la simulazione, viene visualizzato il plot con il confronto tra segnale da ricostruire, campioni rumorosi e segnale ricostruito dal filtro. Nella stessa finestra viene visualizzata l'animazione delle matrici di covarianza dello stato e del guadagno.

Di seguito viene riportato un esempio di filtraggio di un segnale sinusoidale:



## 6.4 Modelli dei generatori di segnale

I generatori di segnale sono sistemi lineari autonomi ( $B = 0$ ) ad uscita scalare ( $p = 1$ ). La loro evoluzione a partire dallo stato iniziale  $x(0)$  genera segnali diversi in base alla natura della matrice dinamica  $A$ .

In particolare il segnale viene prodotto nell'ultima componente dello stato  $x(t)$ , quindi la matrice di uscita sarà del tipo:  $C = \begin{pmatrix} 0 & \dots & 0 & 1 \end{pmatrix}$ .

Di seguito sono riportati i modelli utilizzati nell'applicazione, che per semplicità sono scritti a tempo continuo e poi discretizzati da MATLAB.

- Segnali polinomiali di grado  $n$ :

$$x(t) = a_n t^n + \dots + a_1 t + a_0$$

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}, \quad x(0) = \begin{pmatrix} a_0 \\ a_1 \\ \dots \\ a_n \end{pmatrix}$$

– Scalino ( $n = 0$ ):  $A = 0$

– Rampa ( $n = 1$ ):  $A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$

– Parabola ( $n = 2$ ):  $A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

- Segnale esponenziale:

$$x(t) = x_0 e^{\alpha t}$$

$$A = \alpha \in \mathbb{R}, \quad x(0) = x_0$$

- Segnale sinusoidale ad ampiezza costante:

$$x(t) = a \cos(\omega t)$$

$$A = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix} \in \mathbb{R}^{2 \times 2}, \quad x(0) = \begin{pmatrix} a \\ 0 \end{pmatrix}$$

- Segnale sinusoidale smorzato:

$$x(t) = a e^{\alpha t} \cos(\omega t)$$

$$A = \begin{pmatrix} \alpha & -\omega \\ \omega & \alpha \end{pmatrix} \in \mathbb{R}^{2 \times 2}, \quad x(0) = \begin{pmatrix} a \\ 0 \end{pmatrix}$$