Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"
Corso di Laurea Magistrale in Matematica

Stima con metodi subspace di parametri fisici in modelli differenziali alle derivate parziali

Candidato:
Ludovica Salvi Bentivoglio
Matricola 1132947

Relatore:
Prof. Fabio Marcuzzi

6 luglio 2018

Indice

Introduzione

I metodi *subspace* si collocano nel panorama di algoritmi per la risoluzione di problemi inversi. Da un punto di vista matematico il concetto di ""problema inverso" è ambiguo, ma spesso viene citata come definizione la seguente frase di J.B.Keller [1]: ""Diciamo che due problemi sono uno l'inverso dell'altro se la formulazione dell'uno coinvolge tutta o parte della soluzione dell'altro. Spesso, per ragioni storiche, uno dei due é stato studiato a lungo ed esaurientemente, mentre l'altro non è stato mai studiato o pienamente compreso". I due problemi sono quindi, in un certo senso, uno il duale dell'altro, nel senso che uno dei due puó essere ottenuto dall'altro scambiando i ruoli di dati e incognite. Se ci si restringe all'ambito della fisica, con problema diretto si intende di solito quello che segue il naturale ordinamento di causa ed effetto, ossia il problema che riguarda la deduzione degli effetti a partire dalle cause; viceversa, il problema inverso corrisponde alla ricostruzione delle cause dati gli effetti. Nel nostro caso ad esempio, il problema fisico sará l'analisi della trasmissione del calore attraverso un mezzo fisico: per problema diretto si intende capire come si propaga il calore nel mezzo noti parametri fisici che caratterizzano il mezzo, mentre per problema inverso si intende capire le caratteristiche del mezzo a partire dalla conoscenza di come il calore si trasmette in esso.
I metodi *subspace* sono appunto metodi pensati per stimare parametri di modelli DLTI *state-space*. Questo tipo di modelli ha fondamentale importanza per diversi motivi, uno dei quali é sicuramente il fatto che permette di includere nella descrizione non solo variabili di input e output ma anche le cosidette variabili di stato, le quali permettono di tenere nota anche dello stato interno del sistema. Un altro motivo è che modelli di questo tipo si possono ottenere, come avviene nel nostro caso, dalla discretizzazione nello spazio e nel tempo di modelli di equazioni differenziali alle derivate parziali.
Una delle tecniche piú utilizzate per la stima dei parametri di modelli *state-space* è il *Prediction Error Method* (PEM), che consiste nel trovare il minimo di un funzionale dell'errore di predizione. Le debolezze di questo metodo sono: in primo luogo che, come ogni problema di minimizzazione, c'è sempre il rischio di incorrere in minimi locali; in secondo luogo, il funzionale da minimizzare è spesso una complicata funzione dei parametri del sistema che richiede metodi iterativi e non lineari; infine, questo metodo richiede che si stabilisca a priori la struttura del modello. I metodi subspace sono nati proprio dall'esigenza di aggirare questi problemi.
Si possono evidenziare innanzitutto alcuni aspetti di questi metodi. Il primo è che questi algoritmi utilizzano principalmente la fattorizzazione QR e la decom-

[1] Joseph B. Keller. The American Mathematical Monthly

posizione a valori singolari, vengono cioè utilizzate tecniche di algebra lineare, il che costituisce un vantaggio. Il secondo vantaggio è che non richiedono imposizioni a priori sulla struttura del modello: ad esempio non si richede che sia fissato l'ordine del modello, che viene invece calcolato dal metodo *subspace* stesso. Ultimo aspetto da sottolineare è l'approccio alternativo dei metodi *subspace* rispetto alle altre tecniche di stima dei parametri; questi algoritmi pongono infatti l'accento sulle variabili di stato: prima di tutto fanno una stima degli stati e poi sfruttano tali stime per ricavare le matrici del sistema; adottano cioè un approccio inverso rispetto alle altre tecniche, che invece stimano prima le matrici del sistema e poi da esse ricostruiscono gli stati.
Lo scopo della tesi é analizzare le proprietá numeriche di questi metodi applicandoli al problema della stima dei parametri dell'equazione del calore.
Il Capitolo 1 riassume le nozioni principali sui sistemi DLTI *state-space* necessarie alla comprensione dei contenuti successivi. Il Capitolo 2 si propone invece di descrivere in modo sintetico il filtro di Kalman, un algoritmo che permette una stima ricorsiva degli stati del sistema e su cui si basano i metodi *subspace*. Il Capitolo 3 cerca di trattare in modo sintetico la teoria dei metodi *subspace* nei suoi tratti e caratteristiche principali. Nel Capitolo 4 si vuole velocemente ricordare il metodo degli elementi finiti e i θ-metodi che verranno applicati poi all'equazione del calore per passare da un problema alle derivate parziali a un DLTI *state-space*; in questa fase, per motivi di sintesi, verranno omesse tutte le giustificazioni dell'analisi funzionale che stanno dietro al metodo degli elementi finiti. Nel Capitolo 5 si procderá ad applicare i metodi *subspace* a modelli di forma molto semplice per poterne analizzare nel dettaglio le caratteristiche numeriche ed individuare possibili problemi nell'identificazione del modello. Infine nei Capitoli 6 e 7 vedremo nel dettaglio l'applicazione dei metodi *subspace* alla stima dei parametri per l'equazione del calore su dominio 1-dimensionale e 2-dimensionale, cercando nello specifico un metodo per garantire la stabilitá dei modelli stimati.

Capitolo 1

Sistemi DLTI *state-space*

Un'importante classe di modelli matematici è quella dei modelli *state-space*. Tali modelli servono a descrivere matematicamente la relazione di causa-effetto che intercorre tra gli input e gli output di un sistema dinamico. In particolare tali modelli si collocano nel panorama dei modelli DLTI (a tempo discreto, lineari, tempo-invarianti), rappresentano cioè relazioni di tipo lineare e invarianti per traslazioni temporali. Nello specifico la rappresentazione *state-space* ha la particolarità di appoggiarsi a un vettore di variabili di stato che descrivono il comportamento interno del sistema. Matematicamente assumono la seguente forma

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + Du_k + v_k \end{aligned}$$

con

$$\mathbb{E}\Big[\begin{pmatrix} w_p \\ v_p \end{pmatrix}\begin{pmatrix} w_q \\ v_q \end{pmatrix}^T\Big] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix}\delta_{pq} \geq 0$$

Nel modello abbiamo:

Input e output: i vettori $u_k \in \mathbb{R}^m$ e $y_k \in \mathbb{R}^l$ rappresentano rispettivamente le misurazioni all'istante temporale discreto k degli m input e degli l output del sistema.

Vettori di stato: sono i vettori $x_k \in \mathbb{R}^n$. Gli stati spesso non hanno un'interpretazione fisica e sono invece solo variabili funzionali alla trascrizione del modello;tuttavia, se hanno significato fisico, è sempre possibile trovare una trasformazione per similitudine del modello che permetta di ricondursi alla base opportuna per interpretarli in tale senso. Il numero n delle varibili di stato si dice ordine del sistema.

Vettori di rumore: sono i vettori $w_k \in \mathbb{R}^m$ e $v_k \in \mathbb{R}^l$ e rappresentano rispettivamente il rumore di modello e il rumore di misura. Sono segnali non misurabili di tipo rumore bianco, stazionari a media nulla [1].

Matrici del sistema: sono le matrici $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{l\times n}, D \in \mathbb{R}^{l\times m}$; la matrice A è la matrice che regola la dinamica del sistema in

[1] si veda il Capitolo 2 per dettagli sui segnali stazionari

quanto esprime il legame che c'è tra i vettori di stato di due istanti temporali consecutivi ed è completamente caratterizzata dai suoi autovalori; la matrice B rappresenta invece la trasformazione lineare attraverso la quale gli input influenzano lo stato successivo; la matrice C indica come gli stati interni del sistema si ripercuotano sugli output; infine la matrice D nei sistemi a tempo continuo il più delle volte è nulla e compare a causa della discretizzazione delle equazioni.

Matrici di covarianza: ossia $Q \in \mathbb{R}^{n\times n}, R \in \mathbb{R}^{l\times l}, S \in \mathbb{R}^{n\times l}$.

Nel caso in cui w_k, v_k siano nulli per ogni k, parleremo di *sistema deterministico*; invece, nel caso non vi sia dipendenza dagli ingressi u_k, (cioé $B, D = 0$) parleremo di *sistema stocastico*. Infine, nel caso ci fosse dipendenza sia dai rumori che dagli ingressi, parleremo talvolta di *sistema combinato*.
In questo capitolo parleremo in generale di questo tipo di sistemi, nello specifico parleremo di stabilità, raggiungibilità e osservabilità. Per approfondimenti di vedano [2] e [3].

1.1 La trasformata Z

Si consideri una successione bilatera infinita $a = (a_n)_{n\in\mathbb{Z}}$ Possiamo allora definire la seguente serie di potenze complesse bilatera

$$A(z) = \sum_{n=-\infty}^{\infty} a_n z^{-n}$$

Tale serie prende il nome di *serie formale*.
È noto dalla teoria sulle serie di potenze che tale serie converge assolutamente in un dominio anulare $\rho_1 < |z| < \rho_2$. Quando tale serie converge si parla di *trasformata Z di a* e si indica col simbolo

$$\zeta[a](z) := A(z)$$

Nel caso a sia una successione unilatera è sufficiente applicare la definizione alla successione bilatera coincidente con a per $n \geq 0$ e nulla per $n < 0$. Tale trasformata ha le seguenti proprietà:

Linearità: date due successioni a, b e due scalari α, β vale

$$\zeta[\alpha a + \beta b] = \alpha\zeta[a] + \beta\zeta[b]$$

Traslazioni temporali: La trasformare una successione traslata corrisponde a moltiplicare la trasformata della successione per una potenza; più precisamente: sia a una successione e sia σ l'operatore di traslazione temporale tale che $(\sigma(a))_n = a_{n+1}$. Allora:

$$\zeta[\sigma^k(a)](z) = z^k A(z)$$

Convoluzione: Siano date due successioni a, b le cui trasformate $\zeta[a], \zeta[b]$ sono definite rispettivamente sugli anelli $\rho_a^- < |z| < \rho_a^+, \rho_b^- < |z| < \rho_b^+$, la sua

trasformata; definita la convoluzione delle due successioni nel seguente modo

$$c_m = \sum_{n=-\infty}^{\infty} a_n b_{m-n}$$

la trasformata della convoluzione è

$$\zeta[c] = \zeta[a]\zeta[b]$$

ed è definita sull'intersezione degli anelli di definizione delle altre due trasformate, cioè $\max\{\rho_a^-, \rho_b^-\} < |z| < \min\{\rho_a^+, \rho_b^+\}$.

Somme parziali: sia $s_n = a_0 + a_1 + \cdots + a_n$; vale allora

$$\zeta[s](z) = \frac{1}{1 - z^{-1}} A(z)$$

Differenze: sia $d_n = a_n - a_{n-1} = (a - \sigma^{-1}(a))_n$ vale

$$\zeta[d](z) = (1 - z^{-1})A(z)$$

Nel caso il generico elemento della successione a_n non sia uno scalare ma un vettore o una matrice, la trasformata è definita componente per componente; ad esempio, data una successione di matrici $A = (A_n)_{n\in\mathbb{Z}}$ con $A_n = (a_{ij}^n)$, la sua trasformata sarebbe $\zeta[A] = (\zeta[a_{ij}])$.

Esempio 1. Mostriamo un esempio di applicazione dlla trasformata z. Supponiamo di avere dei dati di input e di output tra i quali intercorre la relazione

$$y(k) + a_1 y(k-1) + a_2 y(k-2) + a_3 y(k-3) = b_1 u(k-1) + b_2 u(k-2) + a_3 u(k-3)$$

Definite le successioni (finite) $y_k = y(k-3)$ e $u_k = u(k-3)$, vediamo che ai due lati dell'uguale abbiamo due convoluzioni, quindi applicando la trasformata z ad ambo i lati dell'equazione, otteniamo

$$A(z)Y(z) = B(z)U(z)$$

$$Y(z) = G(z)U(z) \qquad \text{con } G(z) := \frac{B(z)}{A(z)} = \frac{b_1 z^2 + b_2 z + b_3}{z^3 + a_1 z^2 + a_2 z + a_3}$$

$G(z)$ si dice *funzione di trasferimento*.

Esempio 2. Cosideriamo stavolta un sistema DLTI *state-space*.

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned}$$

Facendo una sostituzione ricorsiva, si può ottenere la seguente espressione

$$y_k = CA^k x_0 + Du_k + \sum_{i=1}^{k} CA^{i-1} B u_{k-i}$$

Supponiamo che $x_0 = 0$ (y_k si dice allora risposta a stato nullo) e definiamo la successione di matrici

$$G_n := \begin{cases} D & \text{se } n = 0 \\ CA^{n-1}B & \text{se } n > 0 \end{cases}$$

Gli elementi di tale successione si dicono *parametri di Markov*. Allora la risposta a stato nullo diventa

$$y_k = \sum_{i=0}^{k} G_i u_{k-i}$$

Calcoliamo adesso la trasformata della successione G_k; per semplicità definiamo le due successioni

$$G_n^1 := \begin{cases} D & \text{se } n = 0 \\ 0 & \text{se } n > 0 \end{cases} \qquad G_n^2 := \begin{cases} 0 & \text{se } n = 0 \\ A^{n-1} & \text{se } n > 0 \end{cases}$$

allora chiaramente $G_n = G_n^1 + CG_n^2B$; quindi

$$\begin{aligned} \zeta[G_n](z) &= \zeta[G_n^1](z) + C\zeta[G_n^2](z)B \\ &= D + C\left[\sum_{i=1}^{\infty} A^{i-1}z^{-i}\right]B \\ &= D + Cz^{-1}\left[\sum_{i=0}^{\infty}(Az^{-1})^i\right]B \\ &= D + Cz^{-1}(\mathbb{I} - z^{-1}A)^{-1}B \\ &= D + C(z - A)^{-1}B \end{aligned}$$

che si dice *matrice di trasferimento* del sistema.

1.2 Stabilità di un sistema

Diamo la seguente definizione

Definizione 1.1. Dato un sistema DLTI con funzione di trasferimento $G(z)$, diciamo che il sistema è *bounded-input bounded-output (BIBO) stable* se per ogni segnale di input limitato u, il segnale di output y è limitato. In tal caso, diciamo che il sistema è stabile.

Un criterio per capire se un sistema sia stabile o meno ci viene proprio dallo studio analitico dell funzione di trasferimento. A tale scopo ricordiamo cosa sono i poli di una funzione: data una funzione olomorfa $f(z)$, z_0 si dice *polo* se è una singolarità isolata della funzione tale che $\lim_{z\to z_0} |f(z)| = \infty$. Allora si dimostra che

Teorema 1.2.1. *Dato un sistema DLTI con funzione di trasferimento $G(z)$, diciamo che il sistema è stabile se e solo se i poli di $G(z)$ sono tutti interni al cerchio unitario.*

Abbiamo visto che nel caso dei sistemi *state-space* la funzione di trasfermento è

$$G(z) = D + C(z - A)^{-1}B$$

Ricordiamo una formula per il calcolo della matric inversa: se $cof(A)$ è la matrice dei complementi algebrici di A, allora $A^{-1} = \frac{1}{det(A)} cof(A)$. Allora

$$G(z) = D + C\frac{cof(z - A)}{det(z - A)}B$$

Quindi le singolarità isolate (tra le quali si trovano i poli) di $G(z)$ sono tutti e soli gli autovalori di A.

1.3 Analisi di stabilità

Consideriamo un generico sistema

$$x_{k+1} = f(x_k)$$

Si dice *orbita* di uno stato $\hat{x}$ l'insieme degli stati

$$\mathcal{O}_{\hat{x}} = \{x : \exists k \in \mathbb{Z} \text{ t.c. } x_0 = \hat{x}, x_k = x\}$$

ossia tutti i punti in cui posso arrivare a partire da x_0 se evolvo il sistema per un tempo k. Un punto x_0 si dice *equilibrio* se $\mathcal{O}_{x_0} = \{x_0\}$.

Definizione 1.2. Sia x_e un equilibrio del sistema. Tale equilibrio si dice che é

- *stabile* se: $\forall \epsilon > 0 \; \exists \delta > 0 : \quad \|x_0 - x_e\| < \delta \Longrightarrow \mathcal{O}_{x_0} \subset B_\epsilon(x_e)$
- *convergente* se: $\exists \delta > 0 : \quad \|x_0 - x_e\| < \delta \Longrightarrow \lim_{k\to\infty} x_k = x_e$
- *asintoticamente stabile* se é stabile e convergente

Nel caso particolare in cui il sistema sia lineare, ovverosia abbia forma

$$x_{k+1} = Ax_k$$

osserviamo che l'origine é sempre punto di equilibrio. Si dimostra che vale inoltre la seguente proposizione:

Proposizione 1.3.1. *In un sistema lineare autonomo (ossia in cui la matrice A non dipende dalla variabile temporale)*

- *se il sistema ha uno stato di equilibrio convergente, esso é necessariamente lo stato 0 e non possono esserci altri equilibri*
- *se l'origine é punto di equilibrio convergente, allora é anche stabile*
- *l'equilibrio nell'origine é stabile se e solo se tutti gli autovalori di A hanno modulo inferiore o uguale a 1 e gli autovalori a modulo unitario sono radici semplici del polinomio minimo*
- *l'equilibrio nell'origine é convergente se e solo se tutti gli autovalori di A hanno modulo inferiore a 1*

1.3.1 Il metodo diretto di Lyapunov

Il metodo diretto di Lyapunov, o secondo metodo di Lyapunov, è un metodo che consente di capire le proprietà di stabilità di un equilibrio x_0 senza avere a disposizione l'espressione esplicita delle traiettorie del sistema. Dalla proposizione 1.3.1 abbiamo visto che se il sistema ammette equilibrio convergente, esso è per forza l'origine, pertanto si suppone di solito che x_0 non sia generico ma che $x_0 = 0$. Consideriamo un aperto W contenente l'origine su cui sia definita una funzione $V : W \to \mathbb{R}$ (detta *funzione di Lyapunov*) continua e che sia invariante rispetto alla funzione f (ossia $f(W) \subseteq W$). Allora in W possiamo definire la funzione

$$\begin{aligned} \Delta V := W &\to \mathbb{R} \\ x &\to V(f(x)) - V(x) \end{aligned}$$

Quindi la funzione ΔV corrisponde alla differenza di valori che assume la funzione V in due punti successivi di una traiettoria. In realtà quindi non è stressamente necessario richiedere che W sia invariante per f, ma è sufficiente che il pezzo di traiettoria che stiamo considerando non esca da W. Vogliamo ora fornire un criterio per determinare la stabilità o meno dell'equilibrio $x_0 = 0$. Diamo dunque la seguente definizione.

Definizione 1.3. Sia W un aperto in $\mathbb{R}^n$ contenente l'origine su cui è definita una funzione reale continua V. Tale funzione si dice:

- *semidefinita positiva* se $V(0) = 0$ ed esiste un intorno dell'origine $W' \subseteq W$ tale che $V(x) \geq 0 \ \forall x \in W'$
- *definita positiva* se è semidefinita positiva ed esiste un intorno dell'origine $W' \subseteq W$ tale che $V(x) > 0 \ \forall x \in W', x \neq 0$

in modo del tutto analogo si definisce una funzione *semidefinita negativa* e *definita negativa*

Si può dimostrare che vale il seguente criterio di stabilità.

Teorema 1.3.2 (Criterio di stabilità di Lyapunov)**.** *Sia $x_0 = 0$ punto di equilibrio del sistema discreto $x_{k+1} = f(x_k)$ con f continua e sia $V : W \to \mathbb{R}$ continua su W intorno dell'origine e ivi definita positiva. Se la funzione ΔV è semidefinita negativa, allora l'origine è equilibrio stabile. Se ΔV é definita negativa, allora l'origine é equilibrio asintoticamente stabile*

Il caso lineare: l'equazione di Lyapunov

L'equazione di Lyapunov é un'equazione di tipo matriciale che fornisce una tecnica per costruire funzioni di Lyapunov di tipo quadratico che possono essere utilizzate per determinare la stabilità dell'origine nel caso di un sistema lineare.

Teorema 1.3.3. *Una condizione necessaria e sufficiente per la stabilità asintotica del sistema lineare discreto $x_{k+1} = Ax_k$ é che per ogni matrice Q definita positiva l'equazione matriciale lineare*

$$Q = P - A^T P A$$

ammetta una soluzione P simmetrica definita positiva

Quindi, più debolmente, l'esistenza di una qualunque matrice P simmetrica definita positiva, tale che $P - A^T PA$ sia anch'essa definita positiva, garantisce l'asintotica stabilità di A. Infatti definita $V(x) = x^T Px$ essa é una candidata funzione di Lyapunov, in quanto é una funzione certamente definita positiva. In tal caso la funzione ΔV assume forma

$$\Delta V(x) = V(Ax) - V(x) = x^T A^T PAx - x^T Px = x^T (A^T PA - P)x$$

che é certamente definita negativa per come é stata scelta P; é quindi garantita l'assoluta stabilitá.
Per vedere la dimostrazione dettagliata dei teoremi visti si rimanda a [3].

1.4 Sistemi algebricamente equivalenti

Quando si scrivono le equazioni di un sistema lineare, si sceglie implicitamente una specifica base per trascrivere matrici e vettori. Ovviamente una scelta o un'altra della base comporta un cambiamento delle componenti delle matrici. In certe situazioni conviene però avere il sistema scritto in modo ""comodo" e per farlo siamo dunque interessati a cambiare la base in cui è scritto il problema. Vediamo dunque come effettuare il cambio di base.
Supponiamo che una coppia di successioni (u_k, y_k) siano rispettivamente sequenza degli input e sequenza degli output di un modello $\mathcal{M} = \{A, B, C, D\}$ tramite il vettore degli stati x_k. Allora, in realtá, saranno input e output anche di un'intera classe di modelli $\mathcal{M}_T = \{A_T, B_T, C_T, D_T\}$ tramite il vettore di stato $(x_T)_k$, definiti al variare della matrice di similitudine T nel seguente modo

$$A_T = T^{-1}AT, \qquad B_T = T^{-1}B, \qquad C_T = CT, \qquad D_T = D, \qquad (x_T)_k = T^{-1}x_k$$

Infatti

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ &= ATT^{-1}x_k + Bu_k \\ T^{-1}x_{k+1} &= T^{-1}ATT^{-1}x_k + T^{-1}Bu_k \\ (x_T)_{k+1} &= A_T(x_T)_k + B_T u_k \end{aligned} \qquad \begin{aligned} y_k &= Cx_k + Du_k \\ &= CTT^{-1}x_k + Du_k \\ &= C_T(x_T)_k + D_T u_k \end{aligned}$$

Per questo si dice che i sistemi sono algebricamente equivalenti

Risposta libera e serie formale associata

La prima equazione di un sistema DLTI *state-space*

$$x_{k+1} = Ax_k + Bu_k$$

si dice *mappa di aggiornamento dello stato* proprio perchè descrive come evolve il vettore di stato da un istante temporale discreto al successivo. Chiamiamo invece *risposta libera* la soluzione dell'equazione

$$x_{k+1} = Ax_k$$

corrisponde cioè all'evoluzione che ha lo stato in mancanza di sollecitazioni. Osserviamo che con una sostituzione ricorsiva della mappa di aggiornamento dello stato è possibile ottenere una scrittura esplicita dello stato al tempo k:

$$x_k = A^k x_0$$

Possiamo ora scrivere la serie formale a cui è associata:

$$X(z) := \sum_{k=0}^{\infty} x_k z^{-k} = x_0 \sum_{k=0}^{\infty} A^k z^{-k} = x_0(\mathbb{I} - Az^{-1})^{-1} = x_0 z(z-A)^{-1}$$

Quello che ci chiediamo è come influisce sull'espressione della serie formale il prendere un sistema algebricamente equivalente a quello considerato. Consideriamo dunque il sistema algebricamente equivalente

$$(x_T)_{k+1} = T^{-1}AT(x_T)_k$$

Poiché chiaramente $(T^{-1}AT)^k = T^{-1}A^kT$, la serie formale associata allo stato di questo sistema sarà

$$X_T(z) = x_0 T^{-1} z(z-A)^{-1} T$$

In pratica quindi, indipendentemente dalla base scelta per la rappresentazione del sistema, l'evoluzione della risposta libera é fortemente legata alle proprietà della matrice

$$z(z-A)^{-1}$$

1.5 Analisi modale

Abbiamo visto che la serie formale associata alla successione degli stati in evoluzione libera assume forma

$$X(z) = M(z)x_0$$

dove $M(z) = z(z-A)^{-1}$. Quello che vogliamo mostrare è che trascivendo in modo opportuno la $M(z)$ è possibile ""decomporre" la serie formale in modo da poterla analizzare meglio in termini di convergenza. Per comprendere meglio cosa si intenda, facciamo l'analogia con le applicazioni lineari. Supponiamo di avere una matrice in $\mathbb{R}^2$ diagonalizzabile in $\mathbb{R}$ di autovalori $\lambda_1 > 1 > \lambda_2 > 0$ e di autovettori di modulo 1 v_1, v_2. Per definizione accade che $\|Av_1\| = \lambda_1\|v_1\| > \|v_1\|$, in pratica quindi vediamo che l'applicazione lineare A induce una dilatazione lungo in sottospazio $span\{v_1\}$; analogamente, poiché $\|Av_2\| = \lambda_2\|v_2\| < \|v_2\|$, vediamo che l'applicazione lineare A induce una contrazione lungo in sottospazio $span\{v_2\}$. Scrivendo quindi in forma diagonale la matrice A, riusciamo a vedere in quali direzioni essa dilati e in quali altre contragga.
Similmente vorremmo poter trascrivere la matrice $M(z)$ in forma ""comoda" (leggasi, in forma diagonale o di Jordan), in modo da poter decomporre la serie $X(z)$ in serie minori (i modi del sistema) di comportamento noto. La possibilità di fare ciò é intimamente legata con la diagonalizzabilità o meno dela matrice del sistema A.

A diagonalizzabile in $\mathbb{R}$

Supponiamo che A sia diagonale,

$$A = diag(\lambda_1, \ldots, \lambda_n)$$

. In tal caso

$$z(z-A)^{-1} = \sum_{k=0}^{\infty} A^k z^{-k} = \sum_{k=0}^{\infty} diag(\lambda_1^k, \dots, \lambda_n^k) z^{-k}$$
$$= diag\left(\sum_{k=0}^{\infty} \lambda_1^k z^{-k}, \dots, \sum_{k=0}^{\infty} \lambda_n^k z^{-k}\right)$$

Allora i modi del sistema sono le funzioni

$$\sum_{k=0}^{\infty} \lambda_1^k z^{-k}, \dots, \sum_{k=0}^{\infty} \lambda_n^k z^{-k}$$

Osserviamo che tali serie sono

- convergenti per $|\lambda_i| < 1$
- limitate ma non convergenti per $|\lambda_i| = 1$
- divergenti per $|\lambda_i| > 1$

A diagonalizzabile in $\mathbb{C}$

Supponiamo che A sia diagonalizzabile in $\mathbb{C}$, con autovalori

$$\lambda_1(\cos\theta_1 \pm i\sin\theta_1), \dots, \lambda_h(\cos\theta_h \pm i\sin\theta_h), \lambda_{2h+1}, \dots, \lambda_n$$

(con $\lambda_1, \dots, \lambda_n > 0$) e autovettori

$$v_1 \pm iw_1, \dots, v_h \pm iw_h, u_{2h+1}, \dots, u_n$$

Allora la matrice A scritta rispetto alla base $v_1, w_1, \dots, v_h, w_h, u_{2h+1}, \dots, u_n$ è

$$A = diag\left(\lambda_1 \left[\begin{smallmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{smallmatrix}\right], \dots, \lambda_h \left[\begin{smallmatrix} \cos\theta_h & \sin\theta_h \\ -\sin\theta_h & \cos\theta_h \end{smallmatrix}\right], \lambda_{2h+1}, \dots, \lambda_n\right)$$

quindi

$$\begin{aligned} z(z-A)^{-1} &= \sum_{k=0}^{\infty} A^k z^{-k} \\ &= \sum_{k=0}^{\infty} diag\left(\lambda_1 \left[\begin{smallmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{smallmatrix}\right], \dots, \lambda_h \left[\begin{smallmatrix} \cos\theta_h & \sin\theta_h \\ -\sin\theta_h & \cos\theta_h \end{smallmatrix}\right], \lambda_{2h+1}, \dots, \lambda_n\right) z^{-k} \\ &= diag\Bigg(\left[\begin{smallmatrix} \sum_{k=0}^{\infty} \lambda_1^k \cos(k\theta_1) z^{-k} & \sum_{k=0}^{\infty} \lambda_1^k \sin(k\theta_1) z^{-k} \\ -\sum_{k=0}^{\infty} \lambda_1^k \sin(k\theta_1) z^{-k} & \sum_{k=0}^{\infty} \lambda_1^k \cos(k\theta_1) z^{-k} \end{smallmatrix}\right], \dots, \\ &\qquad \left[\begin{smallmatrix} \sum_{k=0}^{\infty} \lambda_h^k \cos(k\theta_h) z^{-k} & \sum_{k=0}^{\infty} \lambda_h^k \sin(k\theta_h) z^{-k} \\ -\sum_{k=0}^{\infty} \lambda_h^k \sin(k\theta_h) z^{-k} & \sum_{k=0}^{\infty} \lambda_h^k \cos(k\theta_h) z^{-k} \end{smallmatrix}\right], \\ &\qquad \sum_{k=0}^{\infty} \lambda_{2h+1}^k z^{-k}, \dots, \sum_{k=0}^{\infty} \lambda_n^k z^{-k}\Bigg) \end{aligned}$$

quindi i modi del sistema sono

$$\sum_{k=0}^{\infty} \lambda_1^k \cos(k\theta_1) z^{-k}, \dots, \sum_{k=0}^{\infty} \lambda_h^k \cos(k\theta_h) z^{-k}$$

$$\sum_{k=0}^{\infty} \lambda_1^k \sin(k\theta_1) z^{-k}, \ldots, \sum_{k=0}^{\infty} \lambda_h^k \sin(k\theta_1) z^{-k},$$

$$\sum_{k=0}^{\infty} \lambda_{2h+1}^k z^{-k}, \ldots, \sum_{k=0}^{\infty} \lambda_n^k z^{-k}$$

Lo studio della convergenza dei modi relativi agli autovalori $\lambda_{2h+1}, \ldots, \lambda_n$ è analogo a quello fatto nel caso diagonalizzabile reale; invece per le serie della forma $\sum_{k=0}^{\infty} \lambda_i^k \cos(k\theta_i) z^{-k}, \sum_{k=0}^{\infty} \lambda_i^k \sin(k\theta_i) z^{-k}$ vale che esse sono

- oscillanti convergenti per $|\lambda_i| < 1$
- limitate ma non convergenti per $|\lambda_i| = 1$
- oscillanti divergenti per $|\lambda_i| > 1$

A triangolarizzabile in forma di Jordan in $\mathbb{R}$

Supponiamo che A ammetta forma di Jordan e che in tale forma essa abbia blocchi diagonali $J_1, \ldots, J_r$; siano inoltre n_h, λ_h rispettivamente la dimensione del blocco J_h e l'autovalore relativo al blocco J_h.
Allora $z(z-A)^{-1}$ sarà anch'essa a blocchi $B_1, \ldots, B_r$ con

$$B_h = \begin{cases} b_{ij}^h = \sum_{k=0}^{\infty} \lambda_h^{k-(j-i)} \binom{k}{j-i} z^{-k} & \text{se } i \leq j \\ b_{ij}^h = 0 & \text{se } i > j \end{cases}$$

quindi i modi relativi al blocco J_h sono

$$\sum_{k=0}^{\infty} \lambda_j^{k-l} \binom{k}{l} z^{-k} \qquad \text{con } l = 0, \ldots, n_h - 1$$

Lo studio della convergenza dei modi ottenuti ponendo $l = 0$ è analogo a quello fatto nel caso diagonalizzabile reale; invece per le altre serie vale che

- convergenti per $|\lambda_i| < 1$
- divergenti per $|\lambda_i| \geq 1$

1.5.1 Carattere di convergenza dei modi

Riassumendo quindi,si può quindi formulare il seguente

Teorema 1.5.1. *I modi del sistema $x_{k+1} = Ax_k$ sono*

- *convergenti se e solo se tutti gli autovalori di A hanno modulo strettamente minore di 1;*
- *limitati se e solo se tutti gli autovalori di A hanno modulo minore o uguale a 1 e quelli di modulo unitario sono radici semplici del polinomio minimo.*

L'analogia che abbiamo fatto tra le informazioni che traiamo dalla diagonalizzazione di un'applicazione lineare e le informazioni che traiamo dai modi non é casuale. Cosideriamo infatti un sistema $x_{k+1} = Ax_k$, per semplicità supponiamo A diagonalizzabile. Supponiamo di aver già scritto il sistema usando

l'oppurtuna base di autovettori $\{v_1, \dots, v_n\}$ perchè A assuma forma diagonale e che in tale base lo stato iniziale abbia coordinate $x_0 = (x_1^0, \dots, x_n^0) = \sum_{i=1}^n x_i^0 v_i$. Allora

$$\begin{aligned} x_k = A^k x_0 &= diag(\lambda_1^k, \dots, \lambda_n^k) x_0 \\ &= x_1^0 \lambda_1^k v_1 + \cdots + x_n^0 \lambda_n^k v_n \end{aligned}$$

D'altra parte abbiamo visto che

$$\begin{aligned} X(z) &= diag\left(\sum_{k=0}^{\infty} \lambda_1^k z^{-k}, \dots, \sum_{k=0}^{\infty} \lambda_n^k z^{-k}\right) x_0 \\ &= x_1^0 \left(\sum_{k=0}^{\infty} \lambda_1^k z^{-k}\right) v_1 + \cdots + x_n^0 \left(\sum_{k=0}^{\infty} \lambda_n^k z^{-k}\right) v_n \end{aligned}$$

Quindi gli autovalori della matrice A che inducono contrazione/dilatazione sul relativo autospazio corrispondo a modi convergenti/divergenti.

1.6 Raggiungibilità e osservabilità

Come abbiamo visto, l'analisi modale ha a che fare con l'evoluzione libera del sistema. Tuttavia è interessante anche capire in quale misura le variabili di ingresso influenzino la dinamica del sistema. Per fare ciò introduciamo il concetto di raggiungibilità.
Quando si parla di raggiungibilità si considera un fissato stato iniziale e si cerca di determinare l'insieme degli stati nei quali il sistema può essere portato applicando opportuni ingressi. In generale quando si parla di ""stato fissato" si tende sempre a considerare lo stato 0. Cominciamo dando la definizione di raggiungibilità.

Definizione 1.4. Si consideri un sistema DLTI. Se lo stato iniziale $x_0 = 0$ può essere trasferito a un qualsiasi altro stato ξ in n passi (cioè $x_n = \xi$) agendo solo sugli input $u_0, \dots, u_n$, allora si dice che la coppia $\{A, B\}$ è *raggiungibile*.

Cerchiamo di capire come questa proprietà sia esprimibile in termini matematici. Consideriamo la mappa di aggiornamento dello stato, e supponiamo che lo stato iniziale sia nullo; con una sostituzione ricorsivasi ottiene che

$$x_k = \sum_{i=0}^{k-1} A^{k-1-i} B u_i = \begin{bmatrix} B & AB & A^2B & \dots & A^{k-1}B \end{bmatrix} \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \cdots \\ u_0 \end{bmatrix} \tag{1.1}$$

al variare del vettore contenente gli input io posso dunque scegliere dove arrivare in k passi. Detta dunque

$$\mathcal{R}_k = \begin{bmatrix} B & AB & A^2B & \dots & A^{k-1}B \end{bmatrix}$$

è chiaro che l'insieme degli stati raggiungibili dallo stato 0 sia dato da $Im\mathcal{R}_k$.
Vogliamo ora ricavare una condizione matematica che ci dica se il sistema è raggiungibile; seguendo il ragionamento appena fatto, un sistema di ordine n è raggungibile se $Im\mathcal{R}_\infty = \mathbb{R}^n$ o, equivalentemente, $rk(\mathcal{R}_\infty) = n$. Dal teorema di Hamilton-Cayley però $rk(\mathcal{R}_\infty) = rk(\mathcal{R}_n)$.
In conclusione, vale il seguente teorema.

Teorema 1.6.1. *Una condizione necessaria e sufficiente perchè un sistema sia raggiungibile è che la seguente matrice, detta* matrice di raggiungibilità*, abbia rango massimo*

$$\mathcal{R} = [B, AB, A^2B, \dots, A^{n-1}B]$$

Il concetto di osservabilità si riferisce invece alla possibilità di determinare lo stato di un sistema a partire dalla conoscenza di ingressi e uscite. Per questo una nozione chiave per definire un sistema osservabile e ricostruibile è quella di stati indistinguibili.

Definizione 1.5. Si considerino due stati iniziali x_0^1, x_0^2 e denotiamo con $(y_k^1)_{k\in\mathbb{N}}$, $(y_k^2)_{k\in\mathbb{N}}$ le successioni di output ottenute facendo evolvere il sistema da tali stati iniziali. Tali stati iniziali si dicono *indistinguibili nel futuro* per $\bar{k}$ passi se, a prescindere dalla scelta degli ingressi vale $y_k^1 = y_k^2 \quad \forall\, 0 \le k \le \bar{k}$.
Se non ci sono stati indistinguibili nel futuro per nessun numero di passi diciamo che la coppia $\{A, C\}$ è *osservabile*

Traduciamo la definizione in termini matematici. Dalle equazioni del sistema otteniamo che

$$y_k^l = CA^k x_0^l + Du_k + \sum_{i=1}^{k} CA^{i-1}Bu_{k-i} \qquad \text{con } l = 1, 2$$

quindi dire che i due stati iniziali sono indistinguibili per $\bar{k}$ passi significa dire che

$$CA^k x_0^1 = CA^k x_0^2 \quad \forall\, 0 \le k \le \bar{k} \qquad \text{cioè} \qquad x_0^1 - x_0^2 \in ker CA^k \quad \forall\, 0 \le k \le \bar{k}$$

Detta dunque

$$\mathcal{O}_k = \begin{bmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{k-1} \end{bmatrix}$$

due stati sono indistinguibili in k passi se $x_0^1 - x_0^2 \in ker\mathcal{O}_k$. Per assicurarci che allora che in un sistema di ordine n non esistano tati indistinguibili per un qualunque numero di passi vorremmo che $ker\mathcal{O}_\infty = span\{0\}$, ossia che $rk\mathcal{O}_\infty = n$. Dal teorema di Hamilton-Cayley però $rk(\mathcal{O}_\infty) = rk(\mathcal{O}_{n-1})$.
In conclusione,

Teorema 1.6.2. *Una condizione necessaria e sufficiente perchè un sistema sia osservabile è che la seguente matrice, detta* matrice di osservabilità*, abbia rango massimo*

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{n-1} \end{bmatrix}$$

1.7 Realizzazione minima

Nella sezione 1.1 abbiamo parlato dei coefficienti di Markov G_k e abbiamo osservato che essi determinano completamente la relazione ingresso-uscita, ossia

$$y_k = \sum_{i=0}^{k} G_i u_{k-i}$$

Di conseguenza se due distinte quaterne di matrici $\mathcal{M}_1 = \{A_1, B_1, C_1, D_1\}$ e $\mathcal{M}_2 = \{A_2, B_2, C_2, D_2\}$ generano gli stessi coefficienti di Markov, significa che entrambe le quaterne sono associate a modelli *state-space* che descrivono la stessa relazione ingresso-uscita.

Definizione 1.6. Fissata una sequenza di matrici $(G_k)_{k\in\mathbb{Z}}$, una qualunque quaterna $\mathcal{M} = \{A, B, C, D\}$ che abbia come coefficienti di Markov la sequenza scelta si dice *realizzazione* della sequenza.

Abbiamo già parlato di sistemi algebricamente equivalenti nella sezione 4.3.1 e mostrato come due sistemi di questo tipo descrivano la stessa relazione ingresso-uscita; si tratta dunque di un esempio di due realizzazioni dello stesso sistema. Ció si può anche verificare calcolando esplicitamente i coefficienti di Markov; definiamo dunque i sistemi algebricamente equivalenti $\mathcal{M} = \{A, B, C, D\}$ e $\mathcal{M}_T = \{A_T, B_T, C_T, D_T\}$ con coefficienti di Markov rispettivamente $(G_k)_{k\in\mathbb{Z}}$ e $(G_k^T)_{k\in\mathbb{Z}}$ [2] e

$$A_T = T^{-1}AT, \qquad B_T = T^{-1}B, \qquad C_T = CT, \qquad D_T = D$$

Verifichiamo che i $G_k = G_k^T$:

$$G_0^T = D_T = D = G_0$$

$$G_k^T = C_T A_T^{k-1} B_T = (CT)(T^{-1}AT)^{k-1}(T^{-1}B) = (CT)T^{-1}A^{k-1}T(T^{-1}B) = CA^{k-1}B = G_k$$

É confermato quindi che sistemi algebricamente equivalenti sono realizzazione degli stessi coefficienti di Markov.

I sistemi algebricamente equivalenti hanno però la particolarità di avere tutti lo stesso ordine. Tuttavia non tutte le realizzazioni degli stessi coefficienti di Markov hanno lo stesso ordine. Consideriamo ad esempio le quaterne di matrici

$$A_1 = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \qquad B_1 = \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \qquad C_1 = \mathbb{I}_2 \qquad D_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$A_2 = \lambda \qquad B_2 = b \qquad C_2 = \begin{pmatrix} b_0/b \\ b_1/b \end{pmatrix} \qquad D_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Calcoliamo i coefficienti di Markov di entrambi i sistemi $(G_k^1)_{k\in\mathbb{Z}}$ e $(G_k^2)_{k\in\mathbb{Z}}$: chiaramente $G_0^1 = G_0^2$; ora

$$G_k^1 = \mathbb{I}_2 \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}^{k-1} \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \lambda^{k-1} \begin{pmatrix} b_0 \\ b_1 \end{pmatrix}$$

[2] In questo momento, la notazione ""$\bullet^T$" non sta a indicare la trasposizione di matrice ma denota i coefficienti di Markov associati alla quaterna $\mathcal{M}_T = \{A_T, B_T, C_T, D_T\}$ trasformata tramite la matrice T

$$G_k^2 = \begin{pmatrix} b_0/b \\ b_1/b \end{pmatrix} \lambda^{k-1} b = \lambda^{k-1} \begin{pmatrix} b_0 \\ b_1 \end{pmatrix}$$

Quindi le due quaterne $\mathcal{M}_1 = \{A_1, B_1, C_1, D_1\}$ e $\mathcal{M}_2 = \{A_2, B_2, C_2, D_2\}$ sono realizzazioni degli stessi coefficienti di Markov, benchè il primo sistema sia di ordine 2 e il secondo di ordine 1.

Definizione 1.7. Tra tutte le realizzazioni di una stessa successione di coefficienti di Markov, quelle di ordine minimo si dicono *realizzazioni minime.*

Vale inoltre il seguente teorema

Teorema 1.7.1. *Una realizzazione è minima se è solo se è raggiungibile ed osservabile*

Dimostrazione. .
[realizz. minima ⇒ raggiungibilità e osservabilità](traccia)
Procediamo per assurdo: supponiamo che il sistema non sia raggiungibile e dimostriamo che allora esiste un'altra realizzazione di ordine inferiore, ottenendo un assurdo. La dimostrazione é articolata nei seguenti passaggi:

1. Supponiamo di avere un sistema $\mathcal{M} = \{A, B, C, D\}$ che assuma una delle seguenti due forme

$$A = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} \qquad B = \begin{pmatrix} B_1 \\ 0 \end{pmatrix} \qquad C = \begin{pmatrix} C_1 & C_2 \end{pmatrix} \tag{1.2}$$

$$A = \begin{pmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{pmatrix} \qquad B = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} \qquad C = \begin{pmatrix} C_1 & 0 \end{pmatrix} \tag{1.3}$$

   Allora il sistema $\tilde{\mathcal{M}} = \{A_{11}, B_1, C_1, D\}$ é algebricamente equivalente pur avendo ordine ridotto; infatti nel primo caso

$$\begin{aligned} CA^kB &= \begin{pmatrix} C_1 & C_2 \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}^k \begin{pmatrix} B_1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} C_1 & C_2 \end{pmatrix} \begin{pmatrix} A_{11}^k & * \\ 0 & A_{22}^k \end{pmatrix} \begin{pmatrix} B_1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} C_1 & C_2 \end{pmatrix} \begin{pmatrix} A_{11}^k B_1 \\ 0 \end{pmatrix} \\ &= C_1 A_{11}^k B_1 \end{aligned}$$

   mentre nel secondo

$$\begin{aligned} CA^kB &= \begin{pmatrix} C_1 & 0 \end{pmatrix} \begin{pmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{pmatrix}^k \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} \\ &= \begin{pmatrix} C_1 & 0 \end{pmatrix} \begin{pmatrix} A_{11}^k & 0 \\ * & A_{22}^k \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} \\ &= \begin{pmatrix} C_1 A_{11}^k & 0 \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} \\ &= C_1 A_{11}^k B_1 \end{aligned}$$

2. é sufficiente dunque mostrare che

- per ogni sistema $\mathcal{M} = \{A, B, C, D\}$ non raggiungibile é possibile trovare una trasformazione T che lo renda algebricamente equivalente a un sistema della forma (1.2)
- per ogni sistema $\mathcal{M} = \{A, B, C, D\}$ non osservabile é possibile trovare una trasformazione T che lo renda algebricamente equivalente a un sistema della forma (1.3)

[realizz. minima $\Leftarrow$ raggiungibilità e osservabilità]
Anche qua procediamo per assurdo: supponiamo che esistano due sistemi $\mathcal{M}_1 = \{A_1, B_1, C_1, D\}$ e $\mathcal{M}_2 = \{A_2, B_2, C_2, D\}$ realizzazioni degli stessi coefficienti di Markov ma con $A_1 \in \mathbb{R}^{n\times n}$ e $A_2 \in \mathbb{R}^{r\times r}$ con $r < n$; supponiamo per assurdo che $\mathcal{M}_1$ sia raggiungibile e osservabile e vediamo che otteniamo una contraddizione. Se i due sistemi sono realizzazioni degli stessi coefficienti di Markov, dette

$$\mathcal{O}_1 = \begin{bmatrix} C_1 \\ C_1A_1 \\ C_1A_1^2 \\ \dots \\ C_1A_1^{n-1} \end{bmatrix} \qquad \mathcal{O}_2 = \begin{bmatrix} C_2 \\ C_2A_2 \\ C_2A_2^2 \\ \dots \\ C_2A_2^{n-1} \end{bmatrix} \qquad \begin{aligned} \mathcal{R}_1 &= [B_1, A_1B_1, A_1^2B_1, \dots, A_1^{n-1}B_1] \\ \mathcal{R}_2 &= [B_2, A_2B_2, A_2^2B_2, \dots, A_2^{n-1}B_2] \end{aligned}$$

deve valere

$$\mathcal{O}_1\mathcal{R}_1 = \mathcal{O}_2\mathcal{R}_2$$

Le matrici $\mathcal{O}_2$ e $\mathcal{R}_2$ hanno rispettivamente r colonne ed r righe, pertanto hanno al più rango r e quindi $\text{rk}(\mathcal{O}_2\mathcal{R}_2) \leq r$; abbiamo però supposto che il primo sistema sia raggiungibile e osservabile, per cui $\mathcal{O}_1$ e $\mathcal{R}_1$ devono avere rango massimo (cioè pari a n) e quindi $\text{rk}(\mathcal{O}_1\mathcal{R}_1) = n$. [3] Assurdo. □

[3] Ricordiamo che,in generale, $\text{rk}(AB) \leq \min\{\text{rk}(A), \text{rk}(B)\}$; tuttavia se $A \in \mathbb{R}^{m\times n}$ e $B \in \mathbb{R}^{n\times p}$, allora $\text{rk}(A) = n \Rightarrow \text{rk}(AB) = \text{rk}(B)$ e $\text{rk}(B) = n \Rightarrow \text{rk}(AB) = \text{rk}(A)$

Capitolo 2

Stima dello stato in sistemi DLTI *state-space*

Come già detto, uno dei punti di forza della rappresentazione *state-space* consiste nella presenza del vettore di stato, il quale rappresenta la descrizione interna del sistema; le sue componenti rappresentato spesso delle vere e proprie quantità fisiche, la cui conoscenza può essere di grande interesse in ambito applicativo. Quando si ha a che fare con un sistema *state-space*, accade che i vettori di ingresso e di uscita siano misurabili, tuttavia non lo sia il vettore di stato; uno dei problemi che ci si trova dunque ad affrontare è la stima di tali variabili. Una delle tecniche che permette la stima degli stati è il *filtro di Kalman*.
L'approccio usuale per la stima degli stati richiede la conoscenza delle matrici del sistema. Vedremo però che una delle particolarita dei metodi *subspace* sta proprio nel fatto di invertire questa logica, ossia stimare prima gli stati e poi usare tale stima per la ricostruzione delle matrici del sistema. Per questo in un secondo momento dovremo riformulare il filtro di Kalman in modo tale che esso sia indipendente dalle matrici di sistema e permetta la stima degli stati avendo noti solo gli input e gli output. Per il momento ci occuperemo solo della trattazione classica del filtro.

Il filtro di Kalman è un algoritmo ricorsivo ed è strutturato essenzialmente in due fasi. Denotiamo con la scrittura $x_{k|j}$ la stima dello stato x_k note le misurazioni degli output $y_1, \dots, y_j$. Si supponga di avere a disposizione le stime $x_{1|1}, \dots, x_{k|k}$:

- la prima fase consta in una prima stima dello stato $(k+1)$-esimo note solo le misurazioni $y_1, \dots, y_k$, che denotiamo con $x_{k+1|k}$; chiameremo questa fase *fase di predizione*

- la seconda fase consiste in una correzione dello stato appena stimato che sfrutta l'output y_{k+1}, calcolando così $x_{k+1|k+1}$; chiameremo questa fase *fase di filtraggio*

La trattazione che faremo in questo capitolo sarà piuttosto riassuntiva. Per una deduzione completa e dettagliata del filtro di Kalman si rimanda a [2].

2.1 Processi stocastici

Un processo stocastico può essere visto come la versione probabilistica di un sistema dinamico. Si tratta infatti di una collezione di variabili aleatorie $\{x(t,\omega) : \Omega \to \Omega, t \in T\}$ indicizzate dal parametro t su uno spazio campione Ω. In un certo senso, fissando t possiamo ottenere una ""fotografia" momentanea del fenomeno, fissando ω otteniamo la traiettoria passante per lo stato ω.
Un esempio classico di processo stocastico è la passeggiata aleatoria. Supponiamo di essere sulla retta dei numeri reali e di poterci muovere un'unità avanti o un'unita indietro ad ogni passo con una qualche probabilità. In generale $x(t,\omega)$ individua la posizione in cui troviamo sulla retta dopo t passi sapendo che siamo partiti dal numero ω.

Diamo una definizione più rigorosa da un punto di vista matematico.
Tanto per cominciare occorre definire un primo spazio misurabile, vale a dire una coppia $(\Omega, \mathcal{F})$ in cui Ω è un insieme, $\mathcal{F}$ è una σ-algebra su Ω. Ricordiamo che una σ-algebra su un insieme Ω è una famiglia di sottoinsiemi di Ω chiuso per complementarietà e unioni numerabili; gli elementi di $\mathcal{F}$ si dicono insiemi misurabili. Poi occorre dotare tale spazio misurabile con una misura di probabilità P. La terna $(\Omega, \mathcal{F}, P)$ si dice appunto spazio di probabilità. Dobbiamo inoltre definire uno spazio topologico E e denotiamo con $\mathcal{E}$ la σ-algebra dei boreliani (ossia la più piccola σ-algebra contenente tutti gli aperti di E). Una variabile aleatoria è allora una funzione $Y : \Omega \to E$ tale che per ogni insieme della σ-algebra $\mathcal{E}$, la sua antiimmagine è un insieme della σ-algebra $\mathcal{F}$, ossia è una funzione misurabile.
Definiamo poi T insieme di parametri che possiamo pensare come insieme dei tempi; tale insieme può essere sia continuo che discreto, ma per semplicità lo penseremo discreto. Allora un *processo stocastico* sullo spazio di probabilità $(\Omega, \mathcal{F}, P)$ a valori in E è una funzione $X : \Omega \times T \to E$ tale che la funzione a t fissato $\omega \to X(\omega, t)$ sia misurabile.

Esistono due modi per descrivere in maniera completa un processo stocastico: il primo è conoscere le distribuzioni di probabilità congiunta, il secondo è conoscere i momenti di ordine k-esimo per ogni k.
Le funzioni di *densità di probabilità congiunta* sono delle funzioni

$$p_{t_1,\dots,t_n}(x_1,\dots,x_n) \qquad \text{tali che}$$

$$p_{t_1,\dots,t_n}(x_1,\dots,x_n)\,dx_1\dots dx_n = P(x_1 \le X_{t_1} \le x_1+dx_1,\dots,x_n \le X_{t_n} \le x_n+dx_n)$$

Tali funzioni sono chiaramente infinite e quindi avere una conoscenza completa del processo tramite tali funzioni non è sempre una strada praticabile.
Si definiscono invece i *momenti di ordine k*. Fissato t, X_t è una variabile aleatoria di cui possiamo calcolare valor medio, varianza, etc... che saranno chiaramente funzioni di t. Si dice momento di ordine k

$$M_X(t_1,\dots,t_k) := \mathbb{E}[X_{t_1} \cdot X_{t_2} \dots X_{t_k}]$$

Anche questi momenti sono infiniti ma quelli che nella pratica vengono utilizzari maggiormente sono i momenti di ordine 1 e 2:

$$\mu_X(t) := \mathbb{E}[X_t]$$

$$\Lambda_{XX}(t,s) := \mathbb{E}[(X_t - \mu_t)(X_s - \mu_s)]$$

vale a dire *valor medio* e *autocovarianza* del processo. Si osservi che $\sigma_X^2(t) = \Lambda_{XX}(t,t)$, dove con $\sigma_X^2(t)$ indichiamo la varianza. Se un processo ha varianze finite si dice *processo del secondo ordine*.
Un'altra quantità importante la funzione di *autocorrelazione* definita come

$$R_{XX}(t,s) := \mathbb{E}[X_t X_s]$$

Esempio 3. Un esempio di processo in cui basta calcolare i momenti del primo e del secondo ordine per avere una buona informazione su di esso è il *processo gaussiano*. Siano infatti $\mu_X(t)$ la media del processo e denotiamo, per brevità, con $\sigma_{ij} := \Lambda_{XX}(t_i, t_j)$ la covarianza. Sia allora $\Sigma = (\sigma_{ij})$ la matrice di covarianza. Allora vale

$$p_{t_1,\dots,t_k}(x_1,\dots,x_k) = \frac{1}{[(2\pi)^k \det(\Sigma)]^{1/2}} \exp\left(-\frac{1}{2}\sum_{i,j=1}^{k} \Sigma_{ij}^{-1}(x_i - \mu_X(t_i))(x_j - \mu_X(t_j))\right)$$

Quindi i momenti di primo e secondo ordine forniscono l'informazione necessaria alla rappresentazione di tutte le densità di probabilità congiunta.

2.1.1 Processi stocastici stazionari

Si consideri un processo stocastico $(X_t)_{t\in\mathbb{Z}}$ le cui proprietà statistiche non cambiano nel tempo; questo concetto può essere espresso in termini di densità di probabilità, ossia

$$p_{t_1,\dots,t_k}(x_1,\dots,x_k) = p_{t_1+l,\dots,t_k+l}(x_1,\dots,x_k)$$

ossia la densità di probabilità congiunta non varia per traslazioni temporali. Si parla allora di *processo stocastico fortemente stazionario*. Da ciò si ricava che, se il processo ha momento di ordine k finito, allora anch'esso è invariante per traslazioni temporali; in particolare, un processo del secondo ordine avrà

$$\mu_X(t) = \mu_X(0) \qquad \Lambda_{XX}(t,s) = \Lambda_{XX}(t-s,0)$$

Supponiamo di avere un processo stocastico del secondo ordine. Se la media è costante e la covarianza dipende solo dalla differenza temporale, allora il processo si dice *debolmente stazionario*. Chiaramente quindi, se un processo è fortemente stazionario è anche debolmente stazionario, ma non è detto che valga il viceversa. Tuttavia si noti che se un processo Gaussiano è debolmente stazionario allora è anche fortemente stazionario.
Che il processo sia stazionario in senso forte o in senso debole, la autocovarianza verrá semplicemente denotata $\Lambda_{XX}(t-s)$. Per i processi stazionari si può assumere sempre che siano a media nulla; questo perchè, visto che la media è costante, possiamo effetuare la seguente sostituzione

$$X_t \quad \longrightarrow \quad X_t - \mu_X$$

ed ottenere così un processo a media nulla. È palese che in tal caso autocovarianza e autocorrelazione del processo coincidono

$$\Lambda_{XX}(l) = R_{XX}(l)$$

2.1.2 Processi ergodici

Sia dato un processo stocastico $X : \Omega \times T \to E$. Fissato $\omega = \bar{\omega}$ la funzione $t \to X(\bar{\omega}, t)$ si dice *realizzazione* del processo.
L'idea è quella di cercare di capire se possiamo dedurre le caratteristiche dell'intero processo dall'osservazione di un'unica realizzazione. In tal caso si parla di *processo ergodico*. Ci occuperemo di questa questione solo nel caso di processi stazionari.

Precedentemente abbiamo definito valor medio, autocovarianza e autocorrelazione del processo. Tali funzioni si dicono anche *medie di insieme*, corrispondono a delle medie (pesate con le probabilità) effettuate a tempo fissato: fissati due tempi t, s posso calcolare $\mu_X(t), \Lambda_{XX}(t,s), R_{XX}(t,s)$.
Vogliamo adesso invece definire quelle che si dicono *medie temporali*, cioè fissato ω (e quindi una realizzazione)

$$\mu_X(\omega) := \lim_{N\to\infty} \frac{1}{2N+1} \sum_{k=-N}^{N} X_\omega(k) \qquad \text{(media temporale)}$$

$$R_{XX}(\omega, l) := \lim_{N\to\infty} \frac{1}{2N+1} \sum_{k=-N}^{N} X_\omega(k) X_\omega(k+l) \quad \text{(autocorrelazione temporale)}$$

In generale queste quantità sono concettualmente molto diverse da quelle definite in precedenza in quanto le medie temporali riguardano una singola realizzazione mentre le medie d'insieme riguardano il processo nel suo complesso.
Un processo si dice ergodico quando le medie temporali coincidono con le medie d'insieme: di conseguenza, le informazioni che noi di solito traiamo da quantità relative all'inter processo possiamo ugualmente trarle da quantità relative a una singola realizzazione.

2.2 Distribuzione gaussiana multivariata

Siano $x \in \mathbb{R}^n, y \in \mathbb{R}^p$ due vettori randomici tali che la loro distribuzione congiunta sia gaussiana. Ciò significa che, denotiati i vettori valor medio e le matrici di covarianza nel seguente modo

$$\mu_x = \mathbb{E}\{x\} \qquad \mu_y = \mathbb{E}\{y\}$$

$$\Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} = \begin{bmatrix} cov\{x,x\} & cov\{x,y\} \\ cov\{y,x\} & cov\{y,y\} \end{bmatrix}$$

la distribuzione di probabilità congiunta delle due variabili è

$$p(x,y) = \frac{1}{[(2\pi)^{(n+p)} \det(\Sigma)]^{1/2}} \exp\left(-\frac{1}{2} \begin{bmatrix} x-\mu_x \\ y-\mu_y \end{bmatrix}^T \Sigma^{-1} \begin{bmatrix} x-\mu_x \\ y-\mu_y \end{bmatrix} \right)$$

Si osservi dunque che, dato un processo stocastico gaussiano, due generiche variabili aleatorie X_t,X_s del processo hanno questa proprietà.
Vediamo dunque due lemmi riguardanti queste variabili aleatorie: il primo dá una formula esplicita per il calcolo della distribuzione condizionata di x dato y; il secondo lega tale distribuzione alla possibilità di calcolare una stima del vettore x che dipenda da y minimizzando la varianza dell'errore di stima.

Lemma 2.2.1. *Se x e y hanno distribuzione congiunta gaussiana, allora la distribuzione condizionata di x dato y è anch'essa gaussiana e ha media*

$$\mathbb{E}[x|y] = \mu_x + \Sigma_{xy}\Sigma_{yy}^{+}(y - m_y)$$

Lemma 2.2.2. *Definiti x e y vettori con distribuzione congiunta gaussiana, definiamo $\hat{x}(y)$ come la funzione $f(y)$ y-misurabile tale che*

$$\hat{x}(y) := argmin_f \mathbb{E}[\|x - f(y)\|^2]$$

Vale che

$$\hat{x}(y) = \mathbb{E}[x|y]$$

pertanto, in particolare, è lineare della y.

Piú avanti daremo una connotazione geometrica alla stima a minima varianza di quest'ultimo lemma; per fare ciò dobbiamo parlare di spazi di Hilbert.

2.3 Spazi di Hilbert

Ricordiamo che uno spazio di Hilbert è una coppia $(\mathcal{H}, \langle\cdot,\cdot\rangle)$ dove $\mathcal{H}$ è uno spazio vettoriale reale o complesso e $\langle\cdot,\cdot\rangle$ è un prodotto scalare definito su esso che che induce una norma $\|\cdot\| := \sqrt{\langle\cdot,\cdot\rangle}$ e quindi una distanza $\mathbf{d}$ tale per cui lo spazio metrico $(\mathcal{H}, \mathbf{d})$ risulta completo.
Come è noto, gli spazi di Hilbert generalizzano il concetto di spazio euclideo e permottono quindi di parlare di distanze, angoli e ortogonalità. Si pensi ad esempio ad $\mathbb{R}^n$ dotato di prodotto scalare usuale e si supponga di voler risolvere il seguente problema: dato W sottospazio vettoriale di $\mathbb{R}^n$ e dato un generico elemento di $v \in \mathbb{R}^n$, vorremmo trovare l'elemento di W ""più vicino" all'elemento v, ossia che lo approssima meglio; in termini matematici

$$w_v := \text{argmin}_W \|v - w\|$$

Il Teorema di Proiezione (si veda [**?**]) dice che tale vettore w_v si ottiene proiettando il vettore v ortogonalmente sul sottospazio W.
Trattare altri spazi di Hilbert ci permetterà quindi di definire opportuni concetti di norma e ortogonalità per calcolare approssimazioni a cui siamo interessati, sfruttando le proiezioni ortogonali. In un generico spazio di Hilbert diremo che due vettori $\xi, \eta \in \mathcal{H}$ si dicono ortogonali se $\langle\xi, \eta\rangle = 0$.

Spazio di Hilbert dei vettori randomici a varianza finita $L_2(\Omega)$

Definiamo il seguente insieme

$$\mathcal{H} = \left\{ x \in \mathbb{R}^n \middle| \mathbb{E}[\|x\|^2] := \sum_{i=1}^{n} \mathbb{E}[x_i^2] < \infty \right\}$$

Si dimostra che tale spazio è uno spazio vettoriale. Si può inoltre vedere che, se dotato del seguente prodotto scalare

$$\langle x, y\rangle := \mathbb{E}[x^T y] = tr(\mathbb{E}[xy^T])$$

e di conseguenza della norma $\|x\| := \sqrt{\langle x, x\rangle}$, esso è uno spazio di Hilbert.

Spazi di Hilbert generati da processi stocastici stazionari

Supponiamo che $(Y_t)_{t\in\mathbb{Z}}$ sia un processo stocastico del secondo ordine a media nulla. Consideriamo ora lo spazio generato da tutte le combinazioni lineari finite di variabili aleatorie Y_t cioè

$$\mathcal{H} = \left\{ \xi := \sum_{k=k_1}^{k_2} a_k Y_k,\ a_k \in \mathbb{R} \right\}$$

Si dimostra che tale spazio vettoriale è uno spazio di Hilbert se dotato del seguente prodotto scalare: siano $\xi, \eta \in \mathcal{H}$, diciamo che il loro prodotto scalare è dato da

$$\langle \xi, \eta \rangle := \mathbb{E}[\xi\eta]$$

Più precisamente: due generici elementi di questo spazio avranno la seguente forma:

$$\xi := \sum_{i=i_1}^{i_2} a_i Y_i \qquad \eta := \sum_{j=j_1}^{j_2} a_j Y_j$$

Allora il loro prodotto scalare sarà

$$\begin{aligned}
\langle \xi, \eta \rangle &= \mathbb{E}\left[\left(\sum_{i=i_1}^{i_2} a_i Y_i \right) \left(\sum_{j=j_1}^{j_2} a_j Y_j \right) \right] \\
&= \sum_{i,j} a_i b_j \mathbb{E}[Y_i Y_j] \\
&= \sum_{i,j} a_i b_j \Lambda_{YY}(i,j) \\
&= \sum_{i,j} a_i b_j \Lambda_{YY}(i-j)
\end{aligned}$$

L'ultima riga è ovviamente vera solo nel caso in cui il processo sia stazionario. Tale spazio di Hilbert è più sinteticamente denotato

$$\mathcal{H} = \overline{\text{span}}\{Y_k | -\infty < k < \infty\}$$

Tale spazio è un sottospazio del precedente spazio di Hilbert.

Spazio di Hilbert generato dai vettori randomici $y_1, \ldots, y_k$ a varianza finita

Si considerino $y_1, \ldots, y_k \in \mathbb{R}^p$ vettori randomici a varianza finita e definiamo lo spazio

$$\mathcal{Y} = \left\{ a + \sum_{i=1}^{k} A_i y_i \middle| a \in \mathbb{R}^n, A_i \in \mathbb{R}^{n\times p} \right\}$$

Poichè i generatori sono vettori a varianza finita, tale spazio vettoriale è un sottospazio dello spazio precedentemente definito $L_2(\Omega)$, purchè dotato dello stesso prodotto scalare e quindi della stessa norma.
È utile inoltre osservare che

Lemma 2.3.1. *Preso un generico vettore $x \in \mathcal{H}$, esso è ortogonale a tutto lo spazio $\mathcal{Y}$, se e solo se verifica $\mathbb{E}[x] = 0$ ed è ortogonale a tutti i generatori di $\mathcal{Y}$.*

Esempio 4. Facciamo un esempio di calcolo della proiezione ortogonale in questo spazio.
Supponiamo di avere due vettori x, y con distribzione di probabilità congiunta gaussiana. Vogliamo calcolare la proiezione ortogonale di x sullo spazio

$$\mathcal{Y} = \{b + Ay | b \in \mathbb{R}^n, A \in \mathbb{R}^{n \times p}\}$$

dobbiamo cioè trovare $\bar{b}$ e $\bar{A}$ tali che $x - (\bar{b} + \bar{A}y) \perp y$. Imponiamo quindi che

$$\begin{aligned} \mathbb{E}[x - \bar{b} - \bar{A}y] = 0 \qquad &\implies \qquad \bar{b} = \mu_x - \bar{A}\mu_y \\ \mathbb{E}[(x - \bar{b} - \bar{A}y)y^T] = 0 \qquad &\implies \qquad \mathbb{E}[(x - \mu_x - \bar{A}(y - \mu_y))y^T] = 0 \end{aligned}$$

quindi $\mathbb{E}[(x-\mu_x-\bar{A}(y-\mu_y))(y-\mu_y)^T] = 0$; si ottiene allora che $\Sigma_{xy} - \bar{A}\Sigma_{yy} = 0$, da cui ricaviamo $\bar{A}$. In conclusione quindi la proiezione di x su $\mathcal{Y}$ é

$$\hat{x} = \mu_x + \Sigma_{xy}\Sigma_{yy}^{+}(y - \mu_y)$$

Quindi, in questo spazio di Hilbert, la proiezione di x su $\mathcal{Y}$ (che denoteremo con il simbolo $\hat{\mathbb{E}}[x|\mathcal{Y}]$) corrisponde alla media $\mathbb{E}[x|y]$ della distribuzione condizionata di x dato y ed è anche la stima di x dato y che minimizza la varianza di errore di stima.

Abbiamo dunque dato una connotazione geometrica alla stima di minima varianza per una coppia vettori con distribuzione congiunta gaussiana, descrivendola in termini di proiezioni ortogonali. Si dimostra che ciò é generalizzabile al caso di più vettori.
Dati dei vettori randomici gaussiani $y_0, \dots, y_k \in \mathbb{R}^p$, definiamo $\mathcal{G} := \sigma\{y_0, \dots, y_k\}$ la σ-algebra generata da tali vettori, ossia la più piccola σ-algebra che rende misurabile l'insieme $\{y_0, \dots, y_k\}$.Possiamo pensare a $\mathcal{G}$ come all'informazione complessiva contenuta nel set di vettori.
Consideriamo dunque il sottospazio di Hilbert appena definito

$$\mathcal{Y} = \left\{ a + \sum_{i=1}^{k} A_i y_i \middle| a \in \mathbb{R}^n, A_i \in \mathbb{R}^{n \times p} \right\}$$

si può stavolta dimostrare che

$$\mathbb{E}[x|\mathcal{G}] = \hat{\mathbb{E}}[x|\mathcal{Y}]$$

ossia la proiezione di x su $\mathcal{Y}$ coincide con il valore medio di x rispetto alla σ-algebra $\mathcal{G}$. [1]

2.4 Stima ottimale dello stato per processi gaussiani

Consideriamo un generico sistema *state-space* puramente stocastico:

$$\begin{cases} x_{k+1} = Ax_k + w_k \\ y_k = C_k + v_k \end{cases}$$

[1] Ricordiamo che, data una σ-algebra $\mathcal{G} \subset \mathcal{F}$, e una variabile aleatoria $X : \Omega \to \mathbb{R}$ integrabile si denota con $\mathbb{E}[X|\mathcal{G}]$ una qualunque variabile aleatoria $\mathcal{G}$-misurabile integrabile tale che $\int_A Y\, dP = \int_A X\, dP \ \forall A \in \mathcal{G}$

$$\mathbb{E}\left[\begin{pmatrix} w_t \\ v_t \end{pmatrix}\begin{pmatrix} w_s \\ v_s \end{pmatrix}^T\right] = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}\delta_{ts}$$

Lo stato iniziale x_{t_0} é gaussiano con media $\mu_x(0)$ e con matrice di covarianza $\Pi(0) := \mathbb{E}[(x_0 - \mu_x(0))(x_0 - \mu_x(0))^T]$, ed é scorrelato da $w_k, v_k \, \forall k$, i quali sono rumori gaussiani a media nulla. Supponiamo di voler ricavare una stima $x_{t+m|t}$ dello stato x_{t+m} assumendo di conoscere solo gli output fino al tempo k, $y_0, \ldots, y_k$, tale che sia minimizzato il funzionale

$$J = \mathbb{E}[\|x_{t+m} - x_{t+m|t}\|^2]$$

Tale problema si dice problema di *predizione* se $m > 0$, problema di *filtrazione* se $m = 0$ e problema di *smoothing* se $m < 0$.

L'idea é quella di sfruttare quanto visto nella sezione precedente in merito ai processi gaussiani. È necessario dunque studiare prima le proprietà dei processi $(x_k)_{k\in\mathbb{Z}}$ e $(y_k)_{k\in\mathbb{Z}}$.

Lemma 2.4.1. *I processi $(x_k)_{k\in\mathbb{Z}}$ e $(y_k)_{k\in\mathbb{Z}}$ sono gaussiani.*

Dimostrazione. Assumento $t_0 = 0$ per semplicità, sostituendo ricorsivamente la prima equazione del modello *state-space* si ottiene che

$$x_k = A^k x_0 + \sum_{j=1}^{k} A^{j-1} w_{k-j}$$

È quindi combinazione lineare di vettori gaussiani, e quindi tutto il processo è gaussiano.
Ovviamente quindi, dalla seconda equazione del modello, ricaviamo che anche il processo degli output è gaussiano. □

Pertanto, trattandosi di processi gaussiani, possiamo concludere che

Lemma 2.4.2. *La stima di minima varianza $x_{t+m|t}$ è espressa in termini di valore medio rispetto alla σ-algebra $\mathcal{F}_t$*

$$x_{t+m|t} := \mathbb{E}[x_{t+m}|\mathcal{F}_t]$$

Inoltre, definito lo spazio di Hilbert

$$\mathcal{Y}_t = \left\{ c + \sum_{i=1}^{t} A_i y_t \middle| c \in \mathbb{R}^n, A_i \in \mathbb{R}^{n\times p} \right\}$$

tale stima è data dalla proiezione dello stato x_{t+m} su $\mathcal{Y}_t$

$$x_{t+m|t} = \hat{\mathbb{E}}[x_{t+m}|\mathcal{Y}_t]$$

2.5 Il Filtro di Kalman per sistemi *state-space* stocastici

Si consideri il seguente processo stocastico

$$\begin{cases} e_0 := y_0 - \mu_y(0) \\ e_k := y_k - \mathbb{E}[y_k|\mathcal{F}_{k-1}] \end{cases}$$

Ricordiamo che lo stesso processo puó essere espresso nel seguente modo

$$\begin{cases} e_0 := y_0 - \mu_y(0) \\ e_k := y_k - \hat{\mathbb{E}}[y_k|\mathcal{Y}_{k-1}] \end{cases}$$

dove il simbolo $\hat{\mathbb{E}}$ denota l'operazione di proiezione ortogonale nello spazio di Hilbert in cui stiamo lavorando.

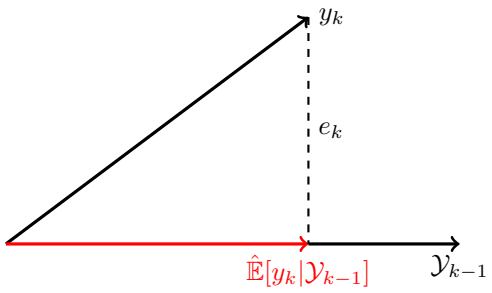


Figura 2.1: Rappresentazione geometrica di e_k

Viene da sé dunque che valga la seguente decomposizione ortogonale

$$\mathcal{Y}_k = \mathcal{Y}_{k-1} \oplus \text{span}\{e_k\}$$

Per tale processo, vale il seguente lemma

Lemma 2.5.1. *Sia $x_{k|k-1}$ la stima dello stato x_k noti gli output $y_0, \dots, y_{k-1}$ e indichiamo nel seguente modo la matrice di covarianza dell'errore di stima*

$$P_{k|k-1} := \mathbb{E}[(x_k - x_{k|k-1})(x_k - x_{k|k-1})^T]$$

Allora il processo $e = (e_k)_{k\in\mathbb{Z}}$ sopra definito ha de seguenti proprietà:

- *Per ogni indice k le seguenti σ-algebre coincidono:*

  $$\sigma\{y_0, \dots, y_k\} = \sigma\{e_0, \dots, e_k\}$$

  In tal caso il processo e si dice innovation process *del processo y*

- *il processo e ha media nulla*

- *la matrice di covarianza del processo e è*

  $$\mathbb{E}[e_t e_s^T] = [CP_{t|t-1}C^T + R]\delta_{ts}$$

Poiche abbiamo detto che possiamo pensare alla σ-algebra $\sigma\{y_0, \dots, y_k\}$ come l'informazione complessiva contenuta nei primi k output, dire che e é un *innovation process* di y equivale essenzialmente a dire che tale processo contiene lo stesso tipo di informazione che é presente nel processo originario.

Vogliamo ora utilizzare questo processo per ricavare le equazioni di predizione e filtraggio del filtro di Kalman.

Equazioni di predizione

Cominciamo calcolando le equazioni ricorsive per la predizione a un passo dello stato $x_{k|k-1}$ e per la matrice di covarianza dell'errore $P_{k|k-1}$

- Osserviamo che possiamo riscrivere e_k nel seguente modo

$$\begin{aligned} e_k &= y_k - \hat{\mathbb{E}}[y_k|\mathcal{Y}_{k-1}] \\ &= y_k - \hat{\mathbb{E}}[(Cx_k + v_k)|\mathcal{Y}_{k-1}] \\ &= y_k - C\hat{\mathbb{E}}[x_k|\mathcal{Y}_{k-1}] \\ &= y_k - Cx_{k|k-1} \qquad (2.1) \\ &= (Cx_k + v_k) - Cx_{k|k-1} \\ &= C(x_k - x_{k|k-1}) + v_k \qquad (2.2) \end{aligned}$$

- Poichè $x_{k|k-1}$ è la proiezione ortogonale di x_k su $\mathcal{Y}_{k-1}$, significa che x_k può essere scritto con la seguente decomposizione ortogonale:

$$x_k = x_{k|k-1} \oplus (x_k - x_{k|k-1}) \qquad (2.3)$$

Pertanto

$$\begin{aligned} \hat{\mathbb{E}}[x_k(x_k - x_{k|k-1})^T] &= \hat{\mathbb{E}}[x_{k|k-1}(x_k - x_{k|k-1})^T] + \hat{\mathbb{E}}[(x_k - x_{k|k-1})(x_k - x_{k|k-1})^T] \\ &= \hat{\mathbb{E}}[x_k - x_{k|k-1})(x_k - x_{k|k-1})^T] \\ &= P_{k|k-1} \qquad (2.4) \end{aligned}$$

- Vediamo che

$$\begin{aligned} \mathbb{E}[x_{k+1}e_k^T] =& \mathbb{E}[(Ax_k + w_k)e_k^T] \\ \overset{(2.2)}{=}& \mathbb{E}[(Ax_k + w_k)(C(x_k - x_{k|k-1}) + v_k)^T] \\ =& A\mathbb{E}[x_k(x_k - x_{k|k-1})^T]C^T + A\mathbb{E}[x_k v_k^T] + \mathbb{E}[w_k(x_k - x_{k|k-1})^T]C^T + \mathbb{E}[w_k v_k^T] \\ =& A\mathbb{E}[x_k(x_k - x_{k|k-1})^T]C^T + \mathbb{E}[w_k v_k^T] \\ \overset{(2.4)}{=}& AP_{k|k-1}C^T + S \qquad (2.5) \end{aligned}$$

- Vogliamo adesso determinare la matrice K_k tale che

$$\hat{\mathbb{E}}[x_{k+1}|e_k] = K_k e_k$$

poiché vogliamo eseguire una proiezione ortogonale dobbiamo imporre che $(x_{k+1} - K_k e_k) \perp e_k$, cioé

$$\hat{\mathbb{E}}[(x_{k+1} - K_k e_k)|e_k] = 0$$

Ma

$$\begin{aligned} \mathbb{E}[(x_{k+1} - K_k e_k)e_k^T] &= \mathbb{E}[x_{k+1}e_k^T] - K_k\mathbb{E}[e_k e_k^T] \\ &\overset{(2.5)}{=} (AP_{k|k-1}C^T + S) - K_k\mathbb{E}[e_k e_k^T] \\ &= (AP_{k|k-1}C^T + S) - K_k(CP_{k|k-1}C^T + R) \end{aligned}$$

quindi

$$K_k = (AP_{k|k-1}C^T + S)(CP_{k|k-1}C^T + R)^{-1}$$

- infine vediamo che

$$\begin{aligned}\hat{\mathbb{E}}[x_{k+1}|\mathcal{Y}_{k-1}] &= \hat{\mathbb{E}}[Ax_k + w_k|\mathcal{Y}_{k-1}] \\ &= A\hat{\mathbb{E}}[x_k|\mathcal{Y}_{k-1}] \\ &= Ax_{k|k-1} \end{aligned} \tag{2.6}$$

Siamo ora pronti a dedurre le equazioni ricorsive. Osserviamo che poichè $\mathcal{Y}_k = \mathcal{Y}_{k-1} \oplus \text{span}\{e_k\}$ allora

$$\begin{aligned} x_{k+1|k} &= \hat{\mathbb{E}}[x_{k+1}|\mathcal{Y}_k] \\ &= \hat{\mathbb{E}}[x_{k+1}|\mathcal{Y}_{k-1}] + \hat{\mathbb{E}}[x_{k+1}|e_k] \\ &\overset{(2.6)}{=} Ax_{k|k-1} + \hat{\mathbb{E}}[x_{k+1}|e_k] \\ &= Ax_{k|k-1} + K_k e_k \\ &\overset{(2.1)}{=} Ax_{k|k-1} + K_k(y_k - Cx_{k|k-1}) \end{aligned}$$

con $K_k = (AP_{k|k-1}C^T + S)(CP_{k|k-1}C^T + R)^{-1}$.
Di conseguenza l'errore di predizione $x_k - x_{k|k-1}$ ammette anche lui una scrittura ricorsiva, precisamente (posto $\varepsilon^p_k := x_k - x_{k|k-1}$):

$$\begin{aligned} \varepsilon^p_{k+1} &= x_{k+1} - x_{k+1|k} \\ &= (Ax_k + w_k) - (Ax_{k|k-1} + K_k(y_k - Cx_{k|k-1})) \\ &= A\varepsilon^p_k + w_k - K_k(y_k - Cx_{k|k-1}) \\ &= A\varepsilon^p_k + w_k - K_k(Cx_k + v_k - Cx_{k|k-1}) \\ &= (A - K_kC)\varepsilon^p_k + w_k - K_kv_k \end{aligned}$$

quindi, ricordando che $(AP_{k|k-1}C^T - S) = K_k(CP_{k|k-1}C^T + R)$, la relazione ricorsiva che lega le matrici di covarianza dell'errore è

$$\begin{aligned} P_{k+1|k} &= \mathbb{E}[\varepsilon^p_{k+1}(\varepsilon^p_{k+1})^T] = \mathbb{E}[((A - K_kC)\varepsilon^p_k + w_k - K_kv_k)((A - K_kC)\varepsilon^p_k + w_k - K_kv_k)^T] \\ &= (A - K_kC)\mathbb{E}[\varepsilon^p_k(\varepsilon^p_k)^T](A - K_kC)^T + (\mathbb{I} \ - K_k)\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix}\begin{pmatrix} \mathbb{I} \\ -K_k^T \end{pmatrix} \\ &= (A - K_kC)P_{k|k-1}(A - K_kC)^T + Q + K_kRK_k^T - SK_k^T - K_kS^T \\ &= AP_{k|k-1}A^T + Q + K_k(CP_{k|k-1}C^T + R)K_k^T - (AP_{k|k-1}C^T - S)K_k^T - K_k(AP_{k|k-1}C^T - S)^T \\ &= AP_{k|k-1}A^T + Q - K_k(CP_{k|k-1}C^T + R)K_k^T \end{aligned}$$

Riassumendo:

Equazioni di predizione
Dato un generico sistema *state-space* puramente stocastico:

$$\begin{cases} x_{k+1} = Ax_k + w_k \\ y_k = C_k + v_k \end{cases}$$

$$\mathbb{E}\left[\begin{pmatrix} w_t \\ v_t \end{pmatrix}\begin{pmatrix} w_s \\ v_s \end{pmatrix}^T\right] = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}\delta_{ts}$$

le equazioni ricorsive che permettono di approssimare lo stato x_{k+1} noti gli output $y_0, \dots, y_k$ sono

$$\begin{aligned} x_{k+1|k} &= Ax_{k|k-1} + K_k(y_k - Cx_{k|k-1}) \\ P_{k+1|k} &= AP_{k|k-1}A^T + Q - K_k(CP_{k|k-1}C^T + R)K_k^T \end{aligned}$$

con $K_k = (AP_{k|k-1}C^T + S)(CP_{k|k-1}C^T + R)^{-1}$

Equazioni di filtraggio

Calcoliamo le equazioni ricorsive per il filtraggio $x_{k|k}$ e per la matrice di covarianza dell'errore $P_{k|k}$

- $$\begin{aligned} \mathbb{E}[x_k e_k^T] &\overset{(2.2)}{=} \mathbb{E}[x_k[(x_k + x_{k|k-1})^T C^T - v_k^T]] \\ &= \mathbb{E}[x_k(x_k + x_{k|k-1})^T]C^T \\ &\overset{(2.4)}{=} P_{k|k-1}C^T \end{aligned} \tag{2.7}$$

- Determiniamo la matrice K_f tale che

  $$\hat{\mathbb{E}}[x_k|e_k] = K_f e_k$$

  Dobbiamo imporre che $(x_k - K_f e_k) \perp e_k$, cioè

  $$\hat{\mathbb{E}}[(x_k - K_f e_k)|e_k] = 0$$

  Ma

  $$\begin{aligned} \hat{\mathbb{E}}[(x_k - K_f e_k)|e_k] &= \mathbb{E}[(x_k - K_f e_k)e_k^T] \\ &= \mathbb{E}[x_k e_k^T] - K_f\mathbb{E}[e_k e_k^T] \\ &\overset{(2.7)}{=} P_{k|k-1}C^T - K_f\mathbb{E}[e_k e_k^T] \\ &= P_{k|k-1}C^T - K_f(CP_{k|k-1}C^T + R) \end{aligned}$$

  dove l'ultima uguaglianza è vera per il lemma (2.5.1). Allora

  $$\hat{\mathbb{E}}[x_k|e_k] = P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}e_k \tag{2.8}$$

- Poi

  $$\begin{aligned} \mathbb{E}[(x_k - x_{k|k-1})e_k^T] &\overset{(2.2)}{=} \mathbb{E}[(x_k - x_{k|k-1})[(x_k - x_{k|k-1})^T C^T - v_k^T]] \\ &= \mathbb{E}[(x_k - x_{k|k-1})(x_k - x_{k|k-1})^T]C^T \\ &= P_{t|t-1}C^T \end{aligned} \tag{2.9}$$

Ricaviamo dunque le equazioni. Poichè $\mathcal{Y}_k = \mathcal{Y}_{k-1} \oplus \text{span}\{e_k\}$ allora

$$\begin{aligned} x_{k|k} &= \hat{\mathbb{E}}[x_k|\mathcal{Y}_k] \\ &= \hat{\mathbb{E}}[x_k|\mathcal{Y}_{k-1}] + \hat{\mathbb{E}}[x_k|e_k] \\ &= x_{k|k-1} + \hat{\mathbb{E}}[x_k|e_k] \\ &\overset{(2.8)}{=} x_{k|k-1} + P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}e_k \\ &\overset{(2.1)}{=} x_{k|k-1} + P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}(y_k - Cx_{k|k-1}) \end{aligned}$$

Da ciò ricaviamo che $x_{k|k-1} - x_{k|k} = -K_f e_k$, con $K_f = P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}$. Detto l'errore $\varepsilon_k^f := x_k - x_{k|k}$ abbiamo che

$$\varepsilon_k^f = x_k - x_{k|k} = (x_k - x_{k|k-1}) + (x_{k|k-1} - x_{k|k}) = \varepsilon_k^p - K_f e_k$$

quindi

$$\begin{aligned} P_{t|t} &= \mathbb{E}[\varepsilon_k^f(\varepsilon_k^f)^T] \\ &= \mathbb{E}[(\varepsilon_k^p - K_f e_k)(\varepsilon_k^p - K_f e_k)^T] \\ &= \mathbb{E}[\varepsilon_k^p(\varepsilon_k^p)^T] - K_f\mathbb{E}[e_k(\varepsilon_k^p)^T] - \mathbb{E}[\varepsilon_k^p e_k^T]K_f^T + K_f\mathbb{E}[e_k e_k^T]K_f^T \\ &\overset{(2.9)}{=} P_{t|t-1} - K_f C P_{t|t-1} - P_{t|t-1}C^T K_f^T + K_f(CP_{k|k-1}C^T + R)K_f^T \end{aligned}$$

Sostituendo K_f possiamo ottenere:

Equazioni di filtraggio
Dato un generico sistema *state-space* puramente stocastico:

$$\begin{cases} x_{k+1} = Ax_k + w_k \\ y_k = C_k + v_k \end{cases}$$

$$\mathbb{E}\left[\begin{pmatrix} w_t \\ v_t \end{pmatrix}\begin{pmatrix} w_s \\ v_s \end{pmatrix}^T\right] = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}\delta_{ts}$$

le equazioni ricorsive che permettono di correggere $x_{k|k-1}$ noti $y_0, \ldots, y_k$ sono

$$\begin{aligned} x_{k|k} &= x_{k|k-1} + P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}(y_k - Cx_{k|k-1}) \\ P_{k|k} &= P_{k|k-1} + P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}CP_{k|k-1} \end{aligned}$$

2.6 Estensione del Filtro di Kalman a sistemi *state-space* generici

Nel caso di sistemi *state-space* generici, quindi della forma

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + w_k \\ y_k = C_k x_k + Du_k + v_k \end{cases}$$

$$\mathbb{E}\left[\begin{pmatrix} w_t \\ v_t \end{pmatrix}\begin{pmatrix} w_s \\ v_s \end{pmatrix}^T\right] = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}\delta_{ts}$$

Poichè non possiamo assicurare che il processo x_k sia ancora gaussiano per u_k generico, assumeremo che u_k abbia forma

$$u_k = L_k y_k = L_k(Cx_k + v_k)$$

in modo da avere garanzia che il processo sia ancora gaussiano.

L'estensione del filtro di Kalman al caso deterministico si basa su questa idea: vogliamo scomporre il vettore di stato e il vettore di output in due componenti facendi parte una di un sistema completamente deterministico e una si un sistema completamente stocastico. Grossomodo, allo stato sistema puramente stocastico possiamo applicare il filtro di Kalman della sezione precedente mentre la componente deterministica verrà stimata con un calcolo banale dedotto dalle equazioni del sistema.
Decomponiamo dunque i vettori di stato e di output in parte deterministica e parte stocastica

$$x_k = x_k^d + x_k^s \qquad\qquad y_k = y_k^d + y_k^s$$

le quasi soddisfano rispettivamente un sottosistema puramente deterministico e un sottosistema puramente stocastico

$$\begin{cases} x_{k+1}^d = Ax_k^d + Bu_k \\ y_k^d = Cx_k^d + Du_k \\ x_0^d = 0 \end{cases} \qquad\qquad \begin{cases} x_{k+1}^s = Ax_k^s + w_k \\ y_k^s = Cx_k^s + v_k \\ x_0^s = x_0 \end{cases}$$

Osserviamo che sostituendo in modo ricorsivo l'equazione relatica a x_k^d vediamo che

$$x_k^d = \sum_{i=0}^{k+1} A^i B u_{k-1-i} \tag{2.10}$$

quindi x_k^d é combinazione lineare delle u_k che a loro volta sono lineari nelle y_k. Questo significa che proiettare ortogonalmente x_k^d in $\mathcal{Y}_k$ lo lascia immutato, quindi

$$x_{k|k-1} = \hat{\mathbb{E}}[x_k|\mathcal{Y}_{k-1}] = \hat{\mathbb{E}}[x_k^s|\mathcal{Y}_{k-1}] + x_k^d$$
$$x_{k|k} = \hat{\mathbb{E}}[x_k|\mathcal{Y}_k] = \hat{\mathbb{E}}[x_k^s|\mathcal{Y}_k] + x_k^d$$

Vediamo quindi che per calcolare la stima complessiva dello stato è sufficiente capire come eseguire le proiezioni $x_{k|j}^s := \hat{\mathbb{E}}[x_k^s|\mathcal{Y}_j]$, (in quando x_k^d può essere dedotto dalla formula (2.10)), per le quali possiamo appoggiarci al filtro di Kalman già dedotto.
Il problema di eserguire le proiezioni $\hat{\mathbb{E}}[x_k^s|\mathcal{Y}_j]$ sta nel fatto che gli y_k non sono l'output del sistema puramente stocastico, pertanto a priori non possiamo applicare il filtro di Kalman così com'è. Vale però il seguente lemma.

Lemma 2.6.1. *Sia $\mathcal{F}_k^s$ la σ-algebra generata dagli output del sistema puramente stocastico $y_0^s, \dots, y_k^s$. Vale*

$$\mathcal{F}_k^s = \mathcal{F}_k$$

Dimostrazione. Osserviamo che

$$y_k = y_k^s + y_k^d = y_k^s + (Cx_k^d + Du_k) \overset{(2.10)}{=} y_k^s + C\sum_{i=0}^{k+1} A^i B u_{k-1-i} + Du_k$$

Dunque:

- per $k = 0$, $y_0^s = y_0$ e quindi $\mathcal{F}_0^s = \mathcal{F}_0$
- per $k = 1$, $y_1 = y_1^s + CBu_0$.
  - u_0 è $\mathcal{F}_0$-misurabile, quindi per il punto precedente è $\mathcal{F}_0^s$-misurabile, e dunque è anche $\mathcal{F}_1^s$-misurabile; pertanto y_1 è somma di due oggetti $\mathcal{F}_1^s$-misurabili e quindi lo è a sua volta: questo prova che $\mathcal{F}_1 \subseteq \mathcal{F}_1^s$
  - in modo del tutto analogo si dimostra l'inclusione $\mathcal{F}_1^s \subseteq \mathcal{F}_1$
- procedendo per induzione, possiamo dimostrareche il lemma è valido per tutti i k

□

Di conseguenza $x_{k|j}^s := \hat{\mathbb{E}}[x_k^s | \mathcal{Y}_j^s]$.
Osserviamo inoltre che l'errore di predizione $x_k - x_{k|k-1}$ coincide con l'errore di predizione della componente stocastica dello stato $x_k^s - x_{k|k-1}^s$, pertanto non sarà necessario ricalcolare le equazioni relative alle matrici $P_{t|t-1}, P_{t|t}$ Possiamo ora ricavare le equazioni del filtro di Kalman generico.

Equazioni di predizione e filtraggio degli stati

Vediamo preliminarmente che

$$\begin{aligned} y_k^s - Cx_{k|k-1}^s &= (y_k^s + y_k^d) - (y_k^d + Cx_{k|k-1}^s) \\ &= (y_k^s + y_k^d) - (Cx_k^d + Du_k + Cx_{k|k-1}^s) \\ &= (y_k^s + y_k^d) - (C(x_k^d + x_{k|k-1}^s) + Du_k) \\ &= y_k - (Cx_{k|k-1} + Du_k) \end{aligned}$$

quindi

$$\begin{aligned} x_{k+1|k} &= x_{k+1|k}^s + x_{k+1}^d \\ &= [Ax_{k|k-1}^s + K(y_k^s - Cx_{k|k-1}^s)] + (Ax_k^d + Bu_k) \\ &= A(x_{k|k-1}^s + x_k^d) + Bu_k + K(y_k^s - Cx_{k|k-1}^s) \\ &= Ax_{k|k-1} + Bu_k + K[y_k - (Cx_{k|k-1} + Du_k)] \end{aligned}$$

infine

$$\begin{aligned} x_{k|k} &= x_{k|k}^s + x_k^d \\ &= [x_{k|k-1}^s + P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}(y_k^s - Cx_{k|k-1}^s)] + x_k^d \\ &= (x_{k|k-1}^s + x_k^d) + P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}(y_k^s - Cx_{k|k-1}^s) \\ &= x_{k|k-1} + P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}[y_k - (Cx_{k|k-1} + Du_k)] \end{aligned}$$

In conclusione:

Filtro di Kalman per un sistema state-space *generico*
Dato un generico sistema *state-space*:

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + w_k \\ y_k = C_k x_k + Du_k + v_k \end{cases}$$

$$\mathbb{E}\left[\begin{pmatrix} w_t \\ v_t \end{pmatrix} \begin{pmatrix} w_s \\ v_s \end{pmatrix}^T \right] = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta_{ts}$$

le equazioni del filtro di Kalman di predizione e filtraggio sono

$$\begin{cases} x_{k+1|k} = Ax_{k|k-1} + Bu_k + K[y_k - (Cx_{k|k-1} + Du_k)] \\ P_{k+1|k} = AP_{k|k-1}A^T + Q - K(CP_{k|k-1}C^T + R)K^T \\ \\ x_{k|k} = x_{k|k-1} + P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}[y_k - (Cx_{k|k-1} + Du_k)] \\ P_{k|k} = P_{k|k-1} + P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}CP_{k|k-1} \end{cases}$$

con $K = (AP_{k|k-1}C^T + S)(CP_{k|k-1}C^T + R)^{-1}$

La formulazione del filtro che abbiamo proposto è la stessa riportata in [2]. Tuttavia non è l'unica formulazione possibile: in alcune situazioni si preferisce stimare non la matrice di covarianza dell'errore $P_{k|j} := \mathbb{E}[(x_k - x_{k|j})(x_k - x_{k|j})^T]$, bensì la matrice di covarianza dello stato stimato $\Pi_{k|j} := \mathbb{E}[x_{k|j}x_{k|j}^T]$. Le equazioni del filtro risultano quindi diverse, come accade in [1]. Questa seconda formulazione del filtro di Kalman sarà utilizzata nel Capitolo 3; per vedere il passaggio da una formulazione all'altra si veda l'Appendice A.

Capitolo 3

Metodi *subspace*

3.1 Generalità sui metodi *subspace*

I metodi *subspace* sono una famiglia di algoritmi che permettono, a partire dalla conoscenza degli input e degli output, di identificare le matrici e i vettori di stato. Rispetto all'approccio classico dei metodi di identificazione hanno delle differenze sostanziali:

- La principale differenza sta nel ruolo centrale che assumono i vettori di stato. Nell' approccio classico si procede innanzitutto con l'identificazione delle matrici e in un momento successivo con la costruzione, attraverso esse, del filtro di Kalman per la stima degli stati. Viceversa i metodi *subspace* partono dalla stima dei vettori di stato e poi, grazie a tali stime, ricavano le matrici del sistema. Questa procedura è possibile poichè in realtà gli stati del filtro di Kalman possono essere ottenuti conoscendo solo gli input e gli output utilizzando tecniche di algebra lineare. Una volta noti gli stati vedremo che il calcolo delle matrici si riduce essere semplicemente un problema ai minimi quadrati lineari.

- Tali metodi si basano su un approccio di tipo geometrico che permetterà una trattazione unificata di tre situazioni differenti: il caso puramente deterministico, in cui i segnali w_k, v_k sono nulli; il caso stocastico, nel quale sono gli input u_k a essere nulli; infine, il caso combinato, in cui invece compaiono tutti i termini descritti. Si vedrà infatti che quest'ultimo caso non è altro che la generalizzazione dei precedenti e che i risultati teorici ricavati per i primi due casi altro non sono che i risultati del caso combinato applicati alla specifica situazione.

- Sono inoltre algoritmi che, in quanto non iterativi, non hanno problemi di convergenza e sono piuttosto veloci. Inoltre non abbiamo il problema di trovare un numero minimo di parametri per la costruzione del modello in quanto l'unico vero parametro da scegliere è l'ordine del modello, che può essere identificato con l'analisi dei valori singolari di una data matrice. D'altra parte però la questione dell'ordine del modello non é da sottovalutare: questi algoritmi sono concepiti per stimare la realizzazione minima a partire da una relazione ingresso-uscita. Questo significa che nel caso

in cui gli si chieda di stimare realizzazioni non minime, possono insorgere problemi.

In questo capitolo si vogliono illustrare in maniera concisa i punti chiave della teoria dei metodi *subspace* e gli algoritmi che da essa si possono costruire. Per approfondimenti si rimanda a [1].

3.2 Proiezioni ortogonali e oblique

3.2.1 Proiezione di dati determininistici

Abbiamo detto che uno dei punti di forza degli algortimi *subspace* consiste nel loro approccio geometrico. Qui di seguito è quindi illustrato il principale strumento geometrico che verrà utilizzato.
Le righe di una generica matrice $A \in \mathbb{R}^{p\times j}$ possono ovviamente essere considerate come la base di un sottospazio vettoriale di $\mathbb{R}^j$, che viene detto *spazio delle righe* di A e lo denotiamo con $R(A)$. Diamo ora la seguente

Definizione 3.1. Date tre matrici $A \in \mathbb{R}^{p\times j}, B \in \mathbb{R}^{q\times j}, C \in \mathbb{R}^{r\times j}$, denotiamo con

$$A/B \in \mathbb{R}^{p\times j}$$

$$A/_C B \in \mathbb{R}^{p\times j}$$

rispettivamente la matrice che ha per righe i vettori ottenuti proiettando ortogonalmente le righe di A su $R(B)$ e la matrice che ha per righe i vettori ottenuti proiettando le righe di A su $R(B)$ parallelamente al sottospazio $R(C)$.

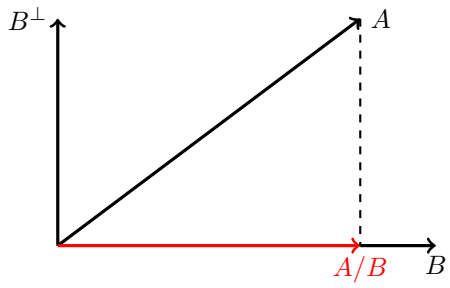


Figura 3.1: proiezione ortogonale

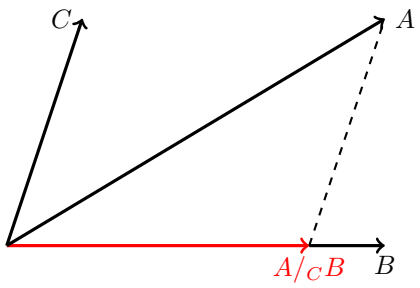


Figura 3.2: proiezione obliqua

La matrice A/B può essere scritta in modo esplicito in due modi equivalenti. Scriviamo le matrici $A, B, A/B$ per righe:

$$A = \begin{bmatrix} a_1 \\ \dots \\ a_p \end{bmatrix} \in \mathbb{R}^{p\times j} \qquad B = \begin{bmatrix} b_1 \\ \dots \\ b_q \end{bmatrix} \in \mathbb{R}^{q\times j} \qquad A/B = \begin{bmatrix} p_1 \\ \dots \\ p_q \end{bmatrix} \in \mathbb{R}^{q\times j}$$

La generica riga p_i corrisponde dunque alla proiezione ortogonale della riga a_i sullo spazio $R(B)$. Come già accennato nella sezione(2.3), ciò è equivalente a dire che

$$p_i := \text{argmin}_{y\in R(B)} \|a_i - y\|$$

Poichè vogliamo che $p_i \in R(B)$, stiamo in pratica cercando dei coefficienti x_{ij} tali che $p_i = \sum_{j=1}^{q} x_{ij} b_j$; definito dunque il vettore $x_i = (x_{i1}, \ldots, x_{iq})$ possiamo scrivere

$$p_i = x_i B \qquad \text{con } x_i = \text{argmin}_x \|a_i - xB\|$$

Quindi una prima scrittura della matrice A/B è

$$A/B = XB \qquad \text{con } X = \text{argmin}_M \|A - MB\|$$

Un secondo modo é quello di scrivere esplicitamente la matrice X. Siccome si vuole che la proiezione avvenga ortogonalmente, cioé che $(a_i - p_i) \in R(B)^\perp$, imponiamo che $(a_i - p_i) b_j^T = 0 \, \forall i, j$. In forma matriciale dunque

$$(A - XB)B^T = 0$$

$$XBB^T = AB^T$$

$$X = AB^T(BB^T)^+$$

da cui ricaviamo che

$$A/B = A\Pi_B \qquad \text{con } \Pi_B := B^T(BB^T)^+ B$$

Mostriamo ora come, passando per la proiezione ortogonale di A su $R(B)$, possiamo ricavare l'espressione algebrica della proiezione obliqua parallela a $R(C)$. Le righe della matrice A possono essere scritte come combinazione lineare delle righe di B e C e dei vettori che costituiscono il complemento ortogonale di $R(B) \oplus R(C)$; equivalentemente, esistono delle matrici X, Y, Z tali che

$$A = XB + YC + Z \begin{pmatrix} B \\ C \end{pmatrix}^\perp$$

È chiaro allora che $A/_C B = XB$. Ora basta osservare che

$$\begin{aligned} XB + YC &= A/\left(\begin{smallmatrix} B \\ C \end{smallmatrix}\right) \\ &= A \begin{pmatrix} B \\ C \end{pmatrix}^T \left[\begin{pmatrix} B \\ C \end{pmatrix} \begin{pmatrix} B \\ C \end{pmatrix}^T \right]^+ \begin{pmatrix} B \\ C \end{pmatrix} \\ &= A \begin{pmatrix} B \\ C \end{pmatrix}^T \left[\begin{pmatrix} B \\ C \end{pmatrix} \begin{pmatrix} B \\ C \end{pmatrix}^T \right]^+_{[:,0:q]} B + A \begin{pmatrix} B \\ C \end{pmatrix}^T \left[\begin{pmatrix} B \\ C \end{pmatrix} \begin{pmatrix} B \\ C \end{pmatrix}^T \right]^+_{[:,q+1:q+r]} C \end{aligned}$$

Quindi

$$A/_C B = A \begin{pmatrix} B \\ C \end{pmatrix}^T \left[\begin{pmatrix} B \\ C \end{pmatrix} \begin{pmatrix} B \\ C \end{pmatrix}^T \right]^+_{[:,0:q]} B$$

In alternativa, si può dimostrare che un altro metodo per il calcolo delle proiezioni oblique é dato dalla seguente formula:

$$\begin{aligned} A/_B C &= (A/B^\perp)(C/B^\perp)^+ C \\ &= (A - A/B)(C - C/B)^+ C \end{aligned}$$

3.2.2 Proiezione di dati stocastici

Abbiamo parlato di ortogonalità tra due vettori. Come già abbiamo accennato (si rimanda alla sezione 2.3), per parlare di ortogonalità occorrerebbe definire un ""ambiente di lavoro", avremmo dovuto specificare cioè specificare lo spazio di Hilbert in cui stiamo lavorando. Ovviamente quando abbiamo parlato di proiezioni ortogonali nella sezione precedente abbiamo dato per scontato che lo spazio di Hilbert considerato fosse $\mathbb{R}^j$ dotato del prodotto scalare usuale $\langle v, w\rangle = \sum_{i=1}^{j} v_i w_i$. Abbiamo però visto che nel caso avessimo a che fare con vettori stocastici ha senso lavorare in un ambiente diverso e usare quindi un altro concetto di ortogonalità.
Lo spazio in cui lavoriamo é dunque lo spazio di Hilbert dei vettori randomici a varianza finita

$$\mathcal{H} = \left\{ v \in \mathbb{R}^n \middle| \mathbb{E}[\|v\|^2] := \sum_{i=1}^{n} \mathbb{E}[v_i^2] < \infty \right\}$$

$$\langle x, y\rangle := \mathbb{E}[x^T y]$$

Siamo interessati a calcolare A/B e $A/_C B$ sfruttando questa definizione di proiezione ortogonale. Diamo la seguente definizione

Definizione 3.2. Si dice matrice di covarianza tra le matrici $A \in \mathbb{R}^{p\times j}, B \in \mathbb{R}^{q\times j}$ la matrice

$$\Phi_{[A,B]} := (\phi_{ij})_{i\in\{1\dots p\}, j\in\{1\dots q\}}$$

con

$$\phi_{ij} = \mathbb{E}[a_i b_j^T]$$

Il procedimento algebrico con cui si ricavano le formule é identico al caso deterministico. Il risultato é

$$A/B = \Phi_{[A,B]}\Phi^{+}_{[B,B]}B$$

$$A/_B C = \begin{pmatrix}\Phi_{[A,C]} & \Phi_{[A,B]}\end{pmatrix}\left[\begin{pmatrix}\Phi_{[C,C]} & \Phi_{[C,B]}\\ \Phi_{[B,C]} & \Phi_{[B,B]}\end{pmatrix}^{+}\right]_{[:,0:r]} C$$

Nello studio dei metodi *subspace* vedremo che essi funzionano molto bene nel caso in cui le nostre sequenze di input,output e stati siano infinite. In tal caso le matrici di cui faremo le proiezioni saranno (almeno nell'ambito delle deduzioni teoriche) con dimensione j infinita. Occorre quindi preoccuparsi di come trattare questo caso. In generale assumeremo che i dati che abbiamo siano ergodici, pertanto vale che

$$\Phi_{[A,B]} := \lim_{j\to\infty} \frac{1}{j} AB^T$$

Pertanto, nel caso di j finito ha senso utilizzare l'approssimazione

$$\Phi_{[A,B]} \approx \frac{1}{j} AB^T$$

Si verifica che con questa approssimazione le definizioni date in contesto deterministico e in contesto stocastico in realtà coincidono.

3.3 Teoria dei metodi *subspace* per sistemi puramente deterministici

Il caso puramente deterministico è più di interesse accademico che pratico in quanto nella realtà la maggiorparte delle misurazioni sono disturbate da un rumore. Tuttavia permette di dare un primo sguardo agli algoritmi *subspace* rendendo di più facile comprensione il più realistico caso combinato.

Problema di identificazione *subspace* deterministico
Dati $u_0, \dots, u_{s-1} \in \mathbb{R}^m$ misurazioni di input e $y_0, \dots, y_{s-1} \in \mathbb{R}^l$ output generat dal modello deterministico (ignoto) di ordine n

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned}$$

vogliamo determinare

- l'ordine del sistema n
- le matrici $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{l\times n}, D \in \mathbb{R}^{l\times m}$ (a meno di trasformazioni per similitudine

3.3.1 Notazione

Allo scopo di trascrivere il modello state space in forma matriciale definiamo le seguenti matrici

Matrici di Hankel: definiamo la matrice di Hankel degli input come

$$U_{0|2i-1} = \begin{bmatrix} u_0 & u_1 & u_2 & \cdots & u_{j-1} \\ u_1 & u_2 & u_3 & \cdots & u_j \\ \vdots & \vdots & \vdots & & \vdots \\ u_{2i-1} & u_{2i} & u_{2i+1} & \cdots & u_{2i+j-2} \end{bmatrix} \in \mathbb{R}^{2mi\times j}$$

è cioè ottenuta incolonnando i vettori di input, dove i è un numero scelto in modo da essere sicuri che sia maggiore dell'ordine n, mentre j è scelto tipicamente in modo tale che $2i + j - 2 = s - 1$ cosicché la matrice contenga tutte le misurazioni di input a disposizione. Similmente definiamo le matrici

$$U_p := U_{0|i-1} \qquad U_f := U_{i|2i-1} \qquad U_{p^+} := U_{0|i} \qquad U_{f^-} := U_{i+1|2i-1}$$

dove la lettera p sta per ""passato" e la lettera f per ""futuro". In modo analogo definiamo $Y_{0|2i-1} \in \mathbb{R}^{li\times j}, Y_p, Y_f, Y_{p^+}, Y_{f^-}$. Siano inoltre

$$W_p := \begin{pmatrix} U_p \\ Y_p \end{pmatrix} \qquad W_{p^+} := \begin{pmatrix} U_{p^+} \\ Y_{p^+} \end{pmatrix}$$

Definiamo infine le matrici che hanno per colonne i vettori di stato

$$X_k = \begin{bmatrix} x_k & x_{k+1} & \cdots & x_{k+j-1} \end{bmatrix} \in \mathbb{R}^{n\times j}$$

$$X_p = X_0 \qquad X_f = X_i$$

Matrici collegate al sistema: Definiamo le matrici di osservabilità e raggiungibilità estese

$$\Gamma_i = \begin{pmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{i-1} \end{pmatrix} \in \mathbb{R}^{li \times n} \qquad \Delta_i^d = \begin{pmatrix} A^{i-1}B & A^{i-2}B & \dots & AB & B \end{pmatrix} \in \mathbb{R}^{n \times mi}$$

dove l'aggettivo estese si riferisce al fatto che $i > n$. Assumiamo che la coppia $\{A, C\}$ sia osservabile, vale a dire che lo stato iniziale del sistema può essere determinato in tempo finito noti solo gli output; ciò si traduce nell'imporre la condizione algebrica $rk(\Gamma_i) = n$. Assumiamo inoltre che la coppia $\{A, B\}$ sia raggiungibile, vale a dire che si può fare in modo che lo stato iniziale del sistema assuma un qualunque valore agendo solo sugli input. Infine definiamo la matrice triangolare inferiore a blocchi di Toepliz

$$H_i = \begin{pmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{i-2}B & CA^{i-3}B & CA^{i-4}B & \dots & D \end{pmatrix} \in \mathbb{R}^{li \times mi}$$

3.3.2 Equazioni matriciali

Definite queste matrici possiamo ora riformulare il modello state space in forma matriciale nel seguente modo

$$\begin{aligned} X_f &= A^i X_p + \Delta_i^d U_p \\ Y_p &= \Gamma_i X_p + H_i U_p \\ Y_f &= \Gamma_i X_f + H_i U_f \end{aligned}$$

3.3.3 Teorema di identificazione deterministica

Teorema 3.3.1. *Supponiamo che*

- *gli input u_k siano persistentemente eccitanti, ossia la matrice di covarianza $R_{uu} = \Phi_{[U_{0|2i-1}, U_{0|2i-1}]}$ sia a rango pieno.*
- $R(U_f) \cap R(X_p) = \emptyset$
- *le matrici dei pesi scelte $W_1 \in \mathbb{R}^{li \times li}, W_2 \in \mathbb{R}^{j \times j}$ siano tali che*
  - *W_1 sia a rango pieno*
  - *$rk(W_p) = rk(W_p W_2)$*

Definiamo la matrice

$$O_i := Y_f /_{U_f} W_p$$

e la decomposizione a valori singolari

$$W_1 O_i W_2 = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} = U_1 S_1 V_1^T$$

Allora

1) $O_i = \Gamma_i X_f$

2) l'ordine del sistema n *é pari al numero di valori singolari non nulli in* S_1

3) $\Gamma_i = W_1^{-1} U_1 S_1^{1/2} T$*, dove* T *è una trasformazione arbitraria non singolare*

4) la parte di X_f *che sta in* $C(W_2)$ *si calcola:* $X_f W_2 = T^{-1} S_1^{1/2} V_1^T$

5) $X_f = \Gamma_i^+ O_i$

Sono necessari alcuni commenti:

- Una definizione più intuitiva di persistentemente eccitante è la seguente: in pratica si richiede che gli input sollecitino tutti i modi del sistema, in modo tale che, sottoponendo tale input al sistema, si ottenga una dinamica più ricca di informazioni possibile.

- La seconda ipotesi corrisponde a richiedere che gli input futuri non siano influenzati dallo stato del sistema corrente, cioè il sistema considerato sia a catena aperta. Viceversa in un sistema a catena chiusa vi è un controllo automatico che ad ogni istante temporale, in base allo stato corrente, agisce sugli input da sottoporre al sistema negli istanti successivi per far sì che gli output si mantengano vicini a degli output di riferimento che occorre rispettare.

- Nella terza ipotesi vengono nominate delle matrici di pesi. La scelta di tali matrici è a discrezione dell'utente in quanto la scelta ottimale è ancora oggetto di studio. Tale formulazione permette però di unificare sotto una stessa teoria algoritmi vari interpretandoli come applicazioni di questo teorema una volta fatta una scelta specifica delle matrici dei pesi. Le ipotesi sul rango sono invece solo funzionali alla costruzione delle relazioni algebriche.

- Le prime due tesi del teorema di potrebbero riassumere nelle seguenti tre relazioni:

$$\begin{aligned} R(O_i) &= R(X_f) \\ C(O_i) &= C(\Gamma_i) \\ rk(O_i) &= n \end{aligned}$$

  da cui deriva il nome di metodi *subspace*: tali metodi permettono di ricavare le matrici del sistema a partire dai sottospazi ottenuti proiettando le matrici contenenti i dati.

- Nella terza tesi del teorema viene introdotta una matrice T che ha la particolarità di essere completamente arbitraria, fatta eccezione per la richiesta di non singolarità. Il motivo dell'introduzione di tale matrice è il seguente: nella dimostrazione del teorema, una volta dimostrata la prima tesi attraverso il solo utilizzo delle equazioni matriciali del modello, si osserva che

$$(W_1 \Gamma_i)(X_f W_2) = W_1 O_i W_2 = U_1 S_1 V_1^T = (U_1 S_1^{1/2} T)(T^{-1} S_1^{1/2} V_1^T)$$

per una qualunque scelta della matrice T; tuttavia, in teoria, perché i prodotti siano uguali membro a membro occorrerebbe scegliere una specifica matrice $T(W_1, W_2)$; una analisi più approfondita però permette di verificare che scegliere la matrice T in modo differente corrisponde semplicemente a trascrivere le matrici di sistema in una base diversa, senza però cambiarne il contenuto in termini di informazioni. Pertanto se non si è interessati a trascrivere il modello in una specifica base, la scelta di T è completamente arbitraria. In molti casi si sceglie dunque $T = \mathbb{I}$.

Un algoritmo

Per mostrare come può essere utilizzato il teorema ai fini del calcolo delle matrici di sistema ripostiamo qui un esempio di algoritmo che ne sfrutta i risultati. Osserviamo che il modello state-space può essere riscritto nella seguente forma matriciale

$$\begin{pmatrix} X_{i+1} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} X_i \\ U_{i|i} \end{pmatrix}$$

Poiché $U_{i|i}, Y_{i|i}$ sono noti, se si riescono a calcolare X_i, X_{i+1} il calcolo di A, B, C, D diventa un problema ai minimi quadrati.

1. calcoliamo innanzitutto l'ordine del sistema:
   - calcoliamo $O_i := Y_f /_{U_f} W_p$, che è calcolabile poiché utilizza solo le sequenze di input e di output
   - decomponiamo ai valori singolari $W_1 O_i W_2 = U_1 S_1 V_1^T$
   - poniamo $n := rk(S_1)$
2. calcoliamo X_i:
   - calcoliamo la matrice di osservabilità estesa come indicato al punto 3) del teorema: $\Gamma_i := W_1^{-1} U_1 S_1^{1/2}$
   - sfruttando il punto 5) del teorema adesso: $X_i := \Gamma_i^+ O_i$
3. calcoliamo X_{i+1}; un corollario del teorema ci permette di calcolare questa matrice seguendo una procedura molto simile al calcolo di X_i:
   - definiamo la matrice $O_{i-1} := Y_{f^-} /_{U_{f^-}} W_{p^+}$
   - calcoliamo la matrice Γ_{i-1} osserando che è ottenibile semplicemente privando Γ_i delle ultime l righe (denoteremo in seguito questa operazione di rimozione di righe con $\underline{\Gamma_i}$)
   - dal corollario discende che $X_{i+1} := \Gamma_{i-1}^+ O_{i-1}$

A questo punto possiamo agevolmente calcolare le matrici del sistema. La tecnica dei minimi quadrati applicata al sistema sopra scritto non è l'unica via che si può seguire ma esistono anche altre tecniche che, ad esempio, garantiscono la stabilità di A ma possono introdurre errore di bias. Sta alla scelta dell'utente capire quale tecnica è la più adatta.

3.4 Teoria dei metodi *subspace* per sistemi generici

Problema di identificazione *subspace* deterministico
Dati $u_0, \dots, u_{s-1} \in \mathbb{R}^m$ misurazioni di input e $y_0, \dots, y_{s-1} \in \mathbb{R}^l$ output generati dal modello deterministico (ignoto) di ordine n

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + Du_k + v_k \end{aligned}$$

con w_k, v_k sequenze a media nulla di tipo rumore bianco con matrice di covarianza

$$\mathbb{E}\Big[\begin{pmatrix} w_p \\ v_p \end{pmatrix}\begin{pmatrix} w_p \\ v_p \end{pmatrix}^T\Big] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix}\delta_{pq}$$

vogliamo determinare

- l'ordine del sistema n
- le matrici $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{l\times n}, D \in \mathbb{R}^{l\times m}$ (a meno di trasformazioni per similitudine
- le matrici $Q \in \mathbb{R}^{n\times n}, S \in \mathbb{R}^{n\times l}, R \in \mathbb{R}^{l\times l}$ tali che siano uguali la seconda statistica d'ordine della componente stocastica dell'output dato e la seconda statistica d'ordine dell'output del sottosistema stocastico.

Ricordiamo cosa si intende per sottosistema stocastico.
Possiamo decomporre i vettori di stato e di output in parte deterministica e parte stocastica

$$x_k = x_k^d + x_k^s \qquad\qquad y_k = y_k^d + y_k^s$$

le quasi soddisfano rispettivamente un sottosistema puramente deterministico e un sottosistema puramente stocastico

$$\begin{cases} x_{k+1}^d = Ax_k^d + Bu_k^d \\ y_k^d = Cx_k^d + Du_k^d \end{cases} \qquad\qquad \begin{cases} x_{k+1}^s = Ax_k^s + w_k \\ y_k^s = Cx_k^s + v_k \end{cases}$$

Siccome noi siamo interessati allo studio del sistema nella sua totalità, richiederemo che le coppie $\{A, C\}$ e $\{A, [B, Q^{1/2}]\}$ siano rispettivamente osservabile e raggiungibile. Tuttavia a priori questo no significa che anche le coppie $\{A, B\}$ e $\{A, Q^{1/2}\}$ siano raggiungibili, il che comporta che le dinamiche degli input e degli output dei due sistemi posso anche essere disaccopiate.

3.4.1 Notazione

Abbiamo già definito le matrici $U_{0|2i-1}, Y_{0|2i-1}$; in modo analogo definiamo $Y^d_{0|2i-1}, Y^s_{0|2i-1}$ usando rispettivamente la componente deterministica e la componente stocastica dell'output. Conserviamo anche le definizioni di W_p, Y_f, U_f, W_{p^+}, Y_{f^-}, U_{f^-}. Definiamo poi la matrice degli stati, la matrice delle componenti deterministiche degli stati e la matrice delle componenti stocastiche degli

stati.

$$
\begin{aligned}
X_k &= \begin{bmatrix} x_k & x_{k+1} & \cdots & x_{k+j-1} \end{bmatrix} \in \mathbb{R}^{n\times j} \\
X_k^d &= \begin{bmatrix} x_k^d & x_{k+1}^d & \cdots & x_{k+j-1}^d \end{bmatrix} \in \mathbb{R}^{n\times j} \qquad X_p^d = X_0^d, \quad X_f^d = X_i^d \\
X_k^s &= \begin{bmatrix} x_k^s & x_{k+1}^s & \cdots & x_{k+j-1}^s \end{bmatrix} \in \mathbb{R}^{n\times j} \qquad X_p^s = X_0^s, \quad X_f^s = X_i^s
\end{aligned}
$$

Per il sottosistema deterministico valgono ancora le definizioni di $\Gamma_i, \Delta_i^d, H_i$. Definiamo inoltre le seguenti matrici di covarianza:

$$
\begin{aligned}
R^{uu} &:= \Phi_{[U_{0|2i-1}, U_{0|2i-1}]} = \begin{pmatrix} \Phi_{[U_p,U_p]} & \Phi_{[U_p,U_f]} \\ \Phi_{[U_f,U_p]} & \Phi_{[U_f,U_f]} \end{pmatrix} = \begin{pmatrix} R_p^{uu} & R_p^{uu} f \\ R_p^{uu} f^T & R_f^{uu} \end{pmatrix} \\
S^{uu} &:= \Phi_{[X_p^d, U_{0|2i-1}]} = \begin{pmatrix} \Phi_{[X_p^d,U_p]} & \Phi_{[X_p^d,U_f]} \end{pmatrix} = \begin{pmatrix} S_p^{xu} & S_f^{xu} \end{pmatrix} \\
\Sigma^d &:= \Phi_{[X_p^d, X_p^d]}
\end{aligned}
$$

Per il sottosistema stocastico, definiamo

$$\Sigma^s := \Phi_{[X_p^s, X_p^s]}$$

poi, detta

$$\Lambda_i := \mathbb{E}[y_{k+i} y_k^T] = \Phi_{[Y_{i|i}, Y_{0|0}]}$$

definiamo le matrici di Toepliz a blocchi

$$
L_i = \begin{bmatrix} \Lambda_0 & \Lambda_{-1} & \Lambda_{-2} & \cdots & \Lambda_{-(i-1)} \\ \Lambda_1 & \Lambda_0 & \Lambda_{-1} & \cdots & \Lambda_{-(i-2)} \\ \Lambda_2 & \Lambda_1 & \Lambda_0 & \cdots & \Lambda_{-(i-3)} \\ \vdots & \vdots & \vdots & & \vdots \\ \Lambda_{i-1} & \Lambda_{i-2} & \Lambda_{i-3} & \cdots & \Lambda_0 \end{bmatrix} = \Phi_{[Y_p, Y_p]} = \Phi_{[Y_f, Y_f]} \in \mathbb{R}^{li\times li}
$$

$$
C_i = \begin{bmatrix} \Lambda_i & \Lambda_{i-1} & \cdots & \Lambda_2 & \Lambda_1 \\ \Lambda_{i+1} & \Lambda_i & \cdots & \Lambda_3 & \Lambda_2 \\ \Lambda_{i+2} & \Lambda_{i+1} & \cdots & \Lambda_4 & \Lambda_3 \\ \vdots & \vdots & & \vdots & \vdots \\ \Lambda_{2i-1} & \Lambda_{2i-2} & \cdots & \Lambda_{i+1} & \Lambda_i \end{bmatrix} = \Phi_{[Y_f, Y_p]} \in \mathbb{R}^{li\times li}
$$

Infine, detta

$$G := \mathbb{E}[x_{k+1}^s y_k^T]$$

definiamo la matrice di raggiungibilitá estesa per il sistema stocastico

$$\Delta_i^c = \begin{pmatrix} A^{i-1}G & A^{i-2}G & \ldots & AG & G \end{pmatrix} \in \mathbb{R}^{n\times li}$$

3.4.2 Una formulazione alternativa per il filtro di Kalman

Nel Capitolo 2 abbia dedotto per intero le equazioni del filtro di Kalman. Il filtro di Kalman permette , dato un sistema *state space* puramente stocastico o combinato, di esegire una stima dei vettori di stato, a patto di conoscere ingressi, uscite e matrici del sistema. Nel quadro dei metodi *subspace* le matrici non sono note ma si vorrebbe comunque avere a disposizione uno strumento per stimare gli stati. Grazie poi a tali stime sarà possibile calcolare le matrici del

sistema. Occorre quindi riformulare in filtro di Kalman.
Nel Capitolo 2 abbiamo visto una prima formulazione del filtro di Kalman nella sua forma usuale: le equazioni ricorsive, oltre a stimare gli stati, eseguono una stima anche della matrice di covarianza dell'errore di stima. Questa formulazione è quella descritta in [2]. Tuttavia per dedurre un filtro che stima gli stati in mancanza della conoscenza delle matrici è preferibile avere a disposizione delle equazioni ricorsive che non stimino la matrice di covarianza dell'errore di stima bensì la matrice di covarianza dello stato stimato, ed è per questo che riportiamo qui di seguito un'altra formulazione del filtro, la stessa che si trova in [1]. Per vedere l'equivalenza tra le due formulazioni si rimanda all'Appendice A.

Definizione 3.3. Siano dati

- una stima dello stato iniziale $\hat{x}_0$
- una stima iniziale della matrice di covarianza dello stato iniziale Π_0
- le misurazioni degli input e degli output $u_0, \dots, u_{k-1}, y_0, \dots, y_{k-1}$

Allora la predizione dello stato al tempo $k+1$ del filtro di Kalman (e della sua matrice di covarianza) si ottiene dalla seguente formula ricorsiva:

$$\begin{aligned} x_{k+1|k} &= Ax_{k|k-1} + Bu_k + K_k[y_k - Cx_{k|k-1} - Du_{k-1}] \\ \Pi_{k+1|k} &= A\Pi_{k|k-1}A^T + K_k(G - A\Pi_{k|k-1}C^T)^T \end{aligned}$$

con $K_k = (G - A\Pi_{k|k-1}C^T)(\Lambda_0 - C\Pi_{k|k-1}C^T)^{-1}$

In questo caso parliamo di *non-steady state Kalman filter*, in quanto la matrice $\Pi_{k+1|k}$ dipende effettivamente dall'indice k. Si parla invece di *steady state Kalman filter* quando per ogni k, $\Pi_{k+1|k} \equiv \Pi$ dove Π è la soluzione dell'equazione di Riccati $\Pi = A\Pi A^T + (G - A\Pi C^T)(\Lambda_0 - C\Pi C^T)^{-1}(G - A\Pi C^T)^T$. In tal caso si noti che anche le matrici K_k diventano indipendenti dall'indice k.
Si può dimostrare, sostituendo ricorsivamente la formula, che esiste un'espressione non ricorsiva per la stima dello stato, precisamente

$$x_{k|k-1} = \begin{pmatrix} A^k - \Omega_k\Gamma_k, & \Delta_k^d - \Omega_k H_k, & \Omega_k \end{pmatrix} \begin{pmatrix} \hat{x}_0 \\ u_0 \\ \dots \\ u_{k-1} \\ y_0 \\ \dots \\ y_{k-1} \end{pmatrix}$$

con $\Omega_k = (\Delta_k^c - A^k P_0 \Gamma_k^T)(L_k - \Gamma_k P_0 \Gamma_k^T)^{-1}$. L'importanza di questa osservazione sta nel fatto che abbiamo scritto lo stato al generico istante temporale k come combinazione lineare dei dati noti e della stima dello stato iniziale. Inoltre questa scrittura ci permette anche di ricavare un'espressione di tipo matriciale

$$\hat{X}_i = \begin{pmatrix} A^i - \Omega_i\Gamma_i, & \Delta_i^d - \Omega_i H_i, & \Omega_i \end{pmatrix} \begin{pmatrix} \hat{X}_0 \\ U_p \\ Y_p \end{pmatrix}$$

Questo tipo di scrittura ha due vantaggi fondamentali

- attraverso la scrittura matriciale ci permette di pensare a una struttura ""verticale" (si eda la figura) per il filtro di Kalman. Parleremo dunque di *banco* di filtri.

- inoltre, vedremo successivamente, tale scrittura non richiede la conoscenza esplicita delle matrici del sistema A, B, C, D, Q, R, S ma possono essere invece ricavate *direttamente dai dati*. Pertanto si ha la possibilità, similmente al caso deterministico, di stimare prima gli stati e in un secondo momento le matrici del sistema.

$$\hat{X}_0 = \begin{bmatrix} \hat{x}_{0,0} & \cdots & \hat{x}_{0,q} & \cdots & \hat{x}_{0,j-1} \end{bmatrix}$$

$$W_p = \begin{bmatrix} u_0 & & u_q & & u_{j-1} \\ \vdots & & \vdots & & \vdots \\ u_{i-1} & & u_{i+q-1} & & u_{i+j-2} \\ y_0 & & y_q & & y_{j-1} \\ \vdots & & \vdots & & \vdots \\ y_{i-1} & & y_{i+q-1} & & y_{i+j-2} \end{bmatrix}$$

$$\hat{X}_i = \begin{bmatrix} x_{i|i-1} & \cdots & x_{i+q|i+q-1} & \cdots & x_{i+j-1|i+j-2} \end{bmatrix}$$

Figura 3.3: Con la notazione $\hat{x}_{0,q}$ si intende una stima iniziale dello stato al tempo q grazie alla quale ricavare, attraverso il filtro di Kalman, la stima dello stato al tempo $i+q$; non è quindi da confondere con $x_{q|q-1}$, stima dello stato calcolato col filtro di Kalman a partire dallo stato iniziale stimato $\hat{x}_{0,q-i}$. La procedura può essere ripetuta per tutte le j colonne.

Per indicare la sequenza di Kalman stimata a partire da $\hat{X}_0, \Pi_0$ useremo la notazione

$$\hat{X}_{i[\hat{X}_0,\Pi_0]}$$

3.4.3 Equazioni matriciali

Analogamente al caso deterministico, riformuliamo il problema state space sottoforma di equazioni matriciali.

$$\begin{aligned} X_f^d &= A^i X_p^d + \Delta_i^d U_p \\ Y_p &= \Gamma_i X_p^d + H_i U_p + Y_p^s \\ Y_f &= \Gamma_i X_f^d + H_i U_f + Y_f^s \end{aligned}$$

3.4.4 Teorema di proiezione ortogonale

A differenza del caso puramente deterministico, in cui abbiamo un'unica strategia per l'identificazione delle matrici A, B, C, D ed essa si basa sul teorema di indentificazione deterministica, nel caso combinato abbiamo due tipologie di

algoritmi. La seconda che illustreremo non sarà altro che l'estensione di quella descritta nel caso deterministico ma soggetta ad alcune restrizioni; la prima invece si appoggia anche ad un teorema nuovo, riportato qui di seguito.

Teorema 3.4.1. *Supponiamo che*

- *Gli input u_k siano scorrelati dai segnali di rumore w_k, v_k*
- *Il numero di misurazioni j tenda ad infinito*
- *w_k, v_k non siano identicamente nulle*
- *gli input u_k siano persistentemente eccitanti, ossia la matrice di covarianza $R_{uu} = \Phi_{[U_{0|2i-1}, U_{0|2i-1}]}$ sia a rango pieno.*

Definiamo la matrice

$$Z_i := Y_f / \begin{pmatrix} W_p \\ U_f \end{pmatrix}$$

e la sequenza di stati stimati ottenuta dal filtro di Kalman

$$\bar{X}_i := \hat{X}_{i[\hat{X}_0, \Pi_0]} \qquad con \begin{cases} \hat{X}_0 := S^{xu}(R^{uu})^{-1} \begin{pmatrix} U_p \\ U_f \end{pmatrix} \\ \Pi_0 := S^{xu}(R^{uu})^{-1}(S^{xu})^T - \Sigma^d \end{cases}$$

Allora vale che

$$Z_i = \Gamma_i \bar{X}_i + H_i U_f$$

Si osservi che un altro modo per trascrivere lo stato iniziale e la matrice di covarianza dello stato iniziale per la sequenza di Kalman del teorema è

$$\begin{aligned} \hat{X}_0 &= X_p^d / U_{[0,2i-1]} \\ \Pi_0 &= -\Phi_{[X_p^d / U_{[0,2i-1]}^{\perp}, X_p^d / U_{[0,2i-1]}^{\perp}]} \end{aligned}$$

Ció permette di dare un'interpretazione a queste due matrici:

- Lo stato iniziale stimato è quindi la migliore approssimazione della componente deterministica dello stato iniziale vero nello spazio delle righe degli input. Il motivo per cui non contiene la parte stocastica X_p^s è perche essa non è stimabile. Si osservi inoltre che Z_i è combinazione lineare degli input e degli output passati e quindi perchè valga la decomposizione di Z_i illustrata nel teorema è ragionevole che occorra prendere un dato iniziale ottenuto da una proiezione sullo spazio delle righe degli input.
- In generale, per ogni sequenza di Kalman, si può dimostrare che, detta P_0 la matrice di covarianza dell'errore sulla stima del dato iniziale, vale che $P_0 = \Sigma^s - \Pi_0$; in questo spedifico caso significa che

$$P_0 = \Phi_{[(X_p^d + X_p^s) - X_p^d / U_{[0,2i-1]}, (X_p^d + X_p^s) - X_p^d / U_{[0,2i-1]}]}$$

3.4.5 Teorema di identificazione

Enunciamo adesso il teorema che estende il teorema di identificazione del caso deterministico.

Teorema 3.4.2. *Supponiamo che*

- *Gli input u_k siano scorrelati dai segnali di rumore w_k, v_k*
- *Il numero di misurazioni j tenda ad infinito*
- *w_k, v_k non siano identicamente nulle*
- *gli input u_k siano persistentemente eccitanti, ossia la matrice di covarianza $R_{uu} = \Phi_{[U_{0|2i-1}, U_{0|2i-1}]}$ sia a rango pieno.*
- *le matrici dei pesi scelte $W_1 \in \mathbb{R}^{li \times li}, W_2 \in \mathbb{R}^{j \times j}$ siano tali che*
  - *W_1 sia a rango pieno*
  - *$rk(W_p) = rk(W_p W_2)$*

Definiamo la matrice

$$O_i := Y_f /_{U_f} W_p$$

la decomposizione a valori singolari

$$W_1 O_i W_2 = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} = U_1 S_1 V_1^T$$

e la sequenza di stati stimati ottenuta dal filtro di Kalman

$$\tilde{X}_i := \hat{X}_{i[\hat{X}_0, \Pi_0]} \qquad con \begin{cases} \hat{X}_0 := X_p^d /_{U_f} U_p \\ \Pi_0 := S^{xu} (R^{uu})^{-1} (S^{xu})^T - \Sigma^d \end{cases}$$

Allora

1) $O_i = \Gamma_i \tilde{X}_f$

2) l'ordine del sistema n è pari al numero di valori singolari non nulli in S_1

3) $\Gamma_i = W_1^{-1} U_1 S_1^{1/2} T$*, dove T è una trasformazione arbitraria non singolare*

4) la parte di X_f che sta in $C(W_2)$ si calcola: $\tilde{X}_f W_2 = T^{-1} S_1^{1/2} V_1^T$

5) $\tilde{X}_f = \Gamma_i^+ O_i$

Nel caso combinato è facile intravedere il ruolo delle matrici dei pesi W_1, W_2. Tra le ipotesi del teorema si richiede che il numero di misurazioni j tenda ad infinito, tuttavia ovviamente nella pratica j è finito. In tal caso i valori singolari diel prodotto $W_1 O_i W_2$ sono tutti non nulli. Siccome in generale siamo interessati a ottenere un modello con un ordine non eccessivamente grande, quello che si fa' è scegliere un ordine n e lavorare con una matrice $\mathcal{R}$ di rango n invece che con la matrice O_i. Tale matrice $\mathcal{R}$ dovrà ovviamente essere ottenuta da O_i conservando la porzione di informazione in essa contenuta che riteniamo più importante. In termini matematici significa che

$$\mathcal{R} = argmin ||W_1 (O_i - \mathcal{R}) W_2||_F$$

Pertanto la scelta delle matrici dei pesi è fondamentale per selezionare quale parte dell'informazione di O_i intendiamo conservare poichè $\mathcal{R}$ dipenderà anche da esse.

3.4.6 I due teoremi a confronto

Osserviamo che

- la sequenza del filtro di Kalman utilizzata nel teorema di identificazione è diversa da quella utilizzata nel teorema di proiezione ortogonale. Infatti benchè utilizzino la stessa Π_0, usano come stati stimati iniziali

$$\bar{X}_0 := X_p^d / \binom{U_p}{U_f}$$
$$\tilde{X}_0 := X_p^d /_{U_f} U_p$$

- una analogia simile intercorre tra le matrici fondamentali dei due teoremi Z_i e O_i, in quanto

$$Z_i := Y_f / \binom{W_p}{U_f}$$
$$O_i := Y_f /_{U_f} W_p$$

  Il problema di identificazione del modello essenzialmente consiste nel trovare un modello che dagli input ci fornisca degli output che stimino il meglio possibile gli output sperimentali. Queste due matrici sono proprio due approssimazioni degli input futuri su due diversi spazi vettoriali (uno incluso nell'altro). Siano infatti L_p, L_u le due matrici che risolvono

$$\min_{L_p, L_u} ||Y_f - (L_p W_p + L_u U_f)||_F$$

  allora é chiaro che

$$Z_i = L_p W_p + L_u U_f, \qquad O_i = L_p W_p$$

3.4.7 Applicazione pratica dei teoremi: gli algoritmi

Per una generica sequenza di Kalman $\hat{X}_i$, indipendentemente dalla scelta dei dati iniziali, valgono le equazioni ricorsive del filtro di Kalman, che scritte il forma matriciale diventano

$$\hat{X}_{i+1} = A\hat{X}_i + BU_{i|i} + K_i(Y_{i|i} - C\hat{X}_i - DU_{i|i})$$

e banalmente vale che $Y_{i|i} = C\hat{X}_i + DU_{i|i} + (Y_{i|i} - C\hat{X}_i - DU_{i|i})$; allora le due equazioni matriciali possono essere radunate nel sistema

$$\begin{pmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} \hat{X}_i \\ U_{i|i} \end{pmatrix} + \begin{pmatrix} K_i \\ \mathbb{I} \end{pmatrix} (Y_{i|i} - C\hat{X}_i - DU_{i|i})$$

Allora, detti $\rho_w = K_i(Y_{i|i} - C\hat{X}_i - DU_{i|i})$ e $\rho_v = Y_{i|i} - C\hat{X}_i - DU_{i|i}$, il sistema diventa

$$\begin{pmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} \hat{X}_i \\ U_{i|i} \end{pmatrix} + \begin{pmatrix} \rho_w \\ \rho_v \end{pmatrix}$$

Come nel caso deterministico, l'idea di fondo è sostitutirvi dentro una qualche sequenza di Kalman e risolverlo poi ai minimi quadrati per ricavare le matrici del sistema. Nei due teoremi precedenti abbiamo introdotto due specifiche sequenze di Kalman ma ciascuno dei due teoremi presenta dei problemi:

- l'applicazione del teorema di identificazione permetterà un conto esplicito di $\bar{X}_i$ ma dall'altro lato creerà problemi di bias salvo che in specifiche situazioni
- d'altra parte il teorema di proiezione descrive solo una relazione di interdipendenza tra la matrice Z_i e $\bar{X}_i$, non un conto esplicito; inoltre il teorema di proiezione non permette il calcolo dell'ordine del sistema né della matrice Γ_i e deve quindi comunque appoggiarsi al teorema di identificazione.

Il calcolo dell'ordine e delle matrici Γ_i,Γ_{i-1} avvengono similmente al caso deterministico:

- calcoliamo $O_i := Y_f /_{U_f} W_p$
- decomponiamo ai valori singolari $W_1 O_i W_2 = U_1 S_1 V_1^T$
- poniamo $n := rk(S_1)$, $\Gamma_i := W_1^{-1} U_1 S_1^{1/2}$, $\Gamma_{i-1} := \underline{\Gamma_i}$

Inoltre, entrambi gli algoritmi stimano le matrici Q, R, S nel seguente modo

$$\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \approx \lim_{j\to\infty} \frac{1}{j} \Big[\begin{pmatrix} \rho_w \\ \rho_v \end{pmatrix} \begin{pmatrix} \rho_w \\ \rho_v \end{pmatrix}^T \Big]$$

Algoritmo 1: applicazione del teorema di proiezione

Abbiamo accennato che il teorema di proiezione non fornisce un'espressione esplicita per $\bar{X}_i, \bar{X}_{i+1}$; infatti definite

$$Z_i := Y_f / \begin{pmatrix} W_p \\ U_f \end{pmatrix} \qquad Z_{i+1} := Y_{f^-} / \begin{pmatrix} W_{p^+} \\ U_{f^-} \end{pmatrix}$$

dal teorema possiamo ricavare che

$$\bar{X}_i = \Gamma_i^+ (Z_i - H_i U_f) \qquad \bar{X}_{i+1} = \Gamma_{i-1}^+ (Z_{i+1} - H_{i-1} U_{f^-})$$

ma queste espressioni non sono ancora esplicite in quanto le matrici H_i,H_{i-1} sono ignote. Possiamo però sostituirle nel sistema matriciale soprastante per ricavare un nuovo sistema ai minimi quadrati:

$$\begin{pmatrix} \Gamma_{i-1}^+ Z_{i+1} \\ Y_{i|i} \end{pmatrix} = \left(\begin{array}{c|c} A & \mathcal{K} \\ C & \end{array} \right) \begin{pmatrix} \Gamma_i^+ Z_i \\ U_f \end{pmatrix} + \begin{pmatrix} \rho_w \\ \rho_v \end{pmatrix} \tag{3.1}$$

$$\text{con } \mathcal{K} = \begin{pmatrix} B & \Gamma_{i-1}^+ H_{i-1} \\ D & 0 \end{pmatrix} - \Gamma_i^+ H_i \begin{pmatrix} A \\ C \end{pmatrix}$$

Siamo ora pronti per la descrizione dell'algoritmo.

1. Si calcolano $Z_i := Y_f / \begin{pmatrix} W_p \\ U_f \end{pmatrix}$ $Z_{i+1} := Y_{f^-} / \begin{pmatrix} W_{p^+} \\ U_{f^-} \end{pmatrix}$ e le si sostituiscono nel sistema soprastante
2. Si dimostra che $R\begin{pmatrix} \rho_w \\ \rho_v \end{pmatrix} \perp R\begin{pmatrix} Z_i \\ U_f \end{pmatrix}$. Pertanto il calcolo di $A, C, \mathcal{K}$ può avvenire risolvendo il sistema soprastante ai minimi quadrati. [1]

[1] Infatti risolvere un sistema $B = XA + Y$ equivale a dire che $X = BA^+ - YA^+$; ma se $R(Y) \perp C(A^+)$ ovviamente $X = BA^+$; ricordando che $C(A^+) = C(A^H) = R(A)$, è chiaro che ci troviamo proprio in questa situazione.

3. Dividiamo $\mathcal{K}$ in blocchi nel seguente modo

$$\mathcal{K} = \begin{pmatrix} \mathcal{K}_{1|1} & \mathcal{K}_{1|2} & \dots & \mathcal{K}_{1|i} \\ \mathcal{K}_{2|1} & \mathcal{K}_{2|2} & \dots & \mathcal{K}_{2|i} \end{pmatrix}$$

Allora si dimostra che grazie alle matrici A, C posso costruire una matrice $\mathcal{N}$ tale che

$$\begin{pmatrix} \mathcal{K}_{1|1} \\ \mathcal{K}_{1|2} \\ \dots \\ \mathcal{K}_{1|i} \\ \mathcal{K}_{2|1} \\ \mathcal{K}_{2|2} \\ \dots \\ \mathcal{K}_{2|i} \end{pmatrix} = \mathcal{N} \begin{pmatrix} D \\ B \end{pmatrix}$$

Pertanto risolvendo questo secondo sistema ai minimi quadrati posso ricavare B, D.
Costruiamo dunque la matrice $\mathcal{N}$. Le matrici $\binom{A}{C}\Gamma_i^+$ e Γ_{i-1}^+ possono spezzate in blocchi nel seguente modo:

$$\begin{pmatrix} A \\ C \end{pmatrix} \Gamma_i^+ = \begin{pmatrix} \mathcal{L}_{1|1} & \mathcal{L}_{1|2} & \dots & \mathcal{L}_{1|i} \\ \mathcal{L}_{2|1} & \mathcal{L}_{2|2} & \dots & \mathcal{L}_{2|i} \end{pmatrix}$$
$$\Gamma_{i-1}^+ = \begin{pmatrix} \mathcal{M}_1 & \mathcal{M}_2 & \dots & \mathcal{M}_{i-1} \end{pmatrix}$$

allora $\mathcal{N}$ è

$$\mathcal{N} = \begin{pmatrix} -\mathcal{L}_{1|1} & \mathcal{M}_1 - \mathcal{L}_{1|2} & \dots & \mathcal{M}_{i-2} - \mathcal{L}_{1|i-1} & \mathcal{M}_{i-1} - \mathcal{L}_{1|i} \\ \mathcal{M}_1 - \mathcal{L}_{1|2} & \mathcal{M}_2 - \mathcal{L}_{1|3} & \dots & \mathcal{M}_{i-1} - \mathcal{L}_{1|i} & 0 \\ \mathcal{M}_2 - \mathcal{L}_{1|3} & \mathcal{M}_3 - \mathcal{L}_{1|4} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \mathcal{M}_{i-1} - \mathcal{L}_{1|i} & 0 & \dots & 0 & 0 \\ \mathbb{I}_l - \mathcal{L}_{2|1} & -\mathcal{L}_{2|2} & \dots & -\mathcal{L}_{2|i-1} & -\mathcal{L}_{2|i} \\ -\mathcal{L}_{2|2} & -\mathcal{L}_{2|3} & \dots & -\mathcal{L}_{2|i} & 0 \\ -\mathcal{L}_{2|3} & -\mathcal{L}_{2|4} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ -\mathcal{L}_{2|i} & 0 & \dots & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbb{I}_l & 0 \\ 0 & \Gamma_{i-1} \end{pmatrix}$$

Algoritmo 2: estensione dell'algoritmo deterministico

Noti n, Γ_i, Γ_{i-1}, grazie al teorema di identificazione (e a un suo corollario) possiamo stimare $\tilde{X}_i, \tilde{X}_{i+1}$

1. calcoliamo $\tilde{X}_i$ sfruttando il punto 5) del teorema: $\tilde{X}_i := \Gamma_i^+ O_i$
2. calcoliamo $\tilde{X}_{i+1}$: definita $O_{i-1} := Y_{f^-}/_{U_{f^-}} W_{p^+}$, poniamo $\tilde{X}_{i+1} := \Gamma_{i-1}^+ O_{i-1}$

Il vantaggio di questo algoritmo, rispetto al precedente, è che possiamo calcolare esplicitamente le due matrici degli stati. C'è tuttavia un problema: $\tilde{X}_i, \tilde{X}_{i+1}$ sono sequenze di Kalman che partono da due stati iniziali *diversi*, cioè

$$\tilde{X}_i \quad \text{è ottenuta a partire da} \quad X_p^d/_{U_f} U_p$$

$$\tilde{X}_{i+1} \quad \text{è ottenuta a partire da} \quad X_p^d/_{U_{f^-}} U_{p+}$$

Quindi in generale non possiamo usare il teorema di identificazione. Tuttavia si dimostra che in tre casi $\tilde{X}_i$ e $\bar{X}_i$ coincidono:

- quando i tende a infinito: in tal caso il *non-steady state Kalman filter* tende a un *steady state Kalman filter*, pertanto l'influenza delle condizioni inziali pressoche scompare
- nel caso in cui il sistema sia puramente deterministico
- nel caso in cui u_k sia un segnale di tipo rumore bianco: in tal caso X_p^d risulta scorrelato da U_p e U_f, di conseguenza i dati iniziali definiti nei due teoremi dai quali costruiamo le sequenze di Kalman sono matrici nulle e pertanto le sequenze coincidono.

Sotto una qualsiasi di queste ipotesi quindi tale algoritmo è applicabile. Nella pratica solo la terza è significativa, in quanto non si possono prendere un numero infinito di misurazioni, ma accade di frequente che l'ingresso non sia un segnale di tipo rumore bianco. In tal caso questo algoritmo restituisce una soluzione affetta da bias.

Un algoritmo robusto

Gli algoritmi presentati funzionano bene nel momento in cui sono applicati a sequeze di dati infinite. Nella pratica tuttavia questa condizione crolla. Ad esempio il primo algoritmo presentato, nel caso la matrice di Hankel degli input sia malcondizionata, non funziona come dovrebbe. Quello che si fa dunque è introdurre delle modifiche (si veda [1]). Alcune di esse hanno il difetto di aumentare notevolmente i costi computazionali ma, alla luce della accuratezza guadagnata, sono comunque più convenienti. Inoltre molte di queste modifiche non sono basate su osservazioni di tipo teorico, ma nella pratica funzionano meglio. Non è ancora chiaro quali di queste modifiche porti all'algoritmo migliore, tuttavia l'algoritmo presentato qui di seguito funziona bene nella situazione pratica di sequenze di dati finite.
Le alterazioni che apportiamo al primo algoritmo sono le seguenti:

- Non si è ancora determinata una scelta ottimale delle matrici dei pesi W_1 e W_2; in questo caso, sono scelte nel seguente modo:

$$W_1 := \mathbb{I}_{li} \qquad W_2 := \Pi_{U_f^\perp}$$

- tramite la risoluzione del sistema (3.1) abbiamo calcolato A e C; capita tuttavia che, benchè la matrice A del sistema da stimare sia stabile, la sua stima eseguita tramite (3.1) non lo sia. In questa situazione si può scegliere di utilizzare altre strategie per il calcolo della matrice A. Nel caso in cui però non si avessero informazioni sulla stabilità della matrice esatta la scelta di forzare la stabilità di A non è sempre la migliore, in quanto questo ci impedisce di distinguere i sistemi stabili dai sistemi instabili. Tuttavia si dimostra che nel caso in cui il modello abbia ordine basso e sia ottenuto da dati ""lineari" il sistema è sempre stabile e quindi in tali condizioni è sensato forzare la stabilità. Negli altri casi a seconda della situazione si può scegliere come procedere. Le strategie per forzare la stabilità sono riportate alla sezione 3.5.
- Scegliamo di utilizzare le matrici A e C per ricalcolare le matrici Γ_i e Γ_{i-1}. Questo perchè le matrici Γ_i e Γ_{i-1} originali, che sono ricavate dalla

decomposizione a valori singolari, sono solo un'approssimazione in quanto le sequenze di dati a nostra disposizione sono finite. Tuttavia tali matrici sono molto importanti per il successivo calcolo di B e D. È dunque bene ricalcolarle a partire dalle A e C ottenute per assicurarsi che B e D siano ""maggiormente compatibili" con A e C.

- Riformuliamo il calcolo di B e D. Per semplicitá di notazione rinominiamo

$$\mathcal{P} := \begin{pmatrix} \Gamma_{i-1}^{+} Z_{i+1} \\ Y_{i|i} \end{pmatrix} - \begin{pmatrix} A \\ C \end{pmatrix} \Gamma_i^{+} Z_i$$

e spezziamo in blocchi orizzontali di uguali dimensioni la sequente matrice

$$U_f = \begin{pmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \\ \vdots \\ \mathcal{Q}_i \end{pmatrix}$$

infine definiamo le seguenti matrici

$$\mathcal{N}_1 = \begin{pmatrix} -\mathcal{L}_{1|1} & \mathcal{M}_1 - \mathcal{L}_{1|2} & \dots & \mathcal{M}_{i-2} - \mathcal{L}_{1|i-1} & \mathcal{M}_{i-1} - \mathcal{L}_{1|i} \\ \mathbb{I}_l - \mathcal{L}_{2|1} & -\mathcal{L}_{2|2} & \dots & -\mathcal{L}_{2|i-1} & -\mathcal{L}_{2|i} \end{pmatrix} \begin{pmatrix} \mathbb{I}_l & 0 \\ 0 & \Gamma_{i-1} \end{pmatrix}$$

$$\mathcal{N}_2 = \begin{pmatrix} \mathcal{M}_1 - \mathcal{L}_{1|2} & \mathcal{M}_2 - \mathcal{L}_{1|3} & \dots & \mathcal{M}_{i-1} - \mathcal{L}_{1|i} & 0 \\ -\mathcal{L}_{2|2} & -\mathcal{L}_{2|3} & \dots & -\mathcal{L}_{2|i} & 0 \end{pmatrix} \begin{pmatrix} \mathbb{I}_l & 0 \\ 0 & \Gamma_{i-1} \end{pmatrix}$$

$$\dots$$

$$\mathcal{N}_i = \begin{pmatrix} \mathcal{M}_{i-1} - \mathcal{L}_{1|i} & 0 & \dots & 0 & 0 \\ -\mathcal{L}_{2|i} & 0 & \dots & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbb{I}_l & 0 \\ 0 & \Gamma_{i-1} \end{pmatrix}$$

allora chiaramente

$$B, D = argmin \Big\| \mathcal{P} - \sum_{k=1}^{i} \mathcal{N}_k \left(\begin{smallmatrix} D \\ B \end{smallmatrix}\right) \mathcal{Q}_k \Big\|_F$$

Ora possiamo determinare una scrittura esplicita di B e D. Date una generica matrice $X = (x_{ij}) = [x_1, \dots, x_n]$ (dove x_i sono le colonne della matrice e x_{ij} le entrate) e un'altra matrice generica Y, definiamo le seguenti operazioni

$$vec(X) := \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \qquad X \otimes Y := \begin{pmatrix} x_{11}Y & \dots & x_{1n}Y \\ \dots & \dots & \dots \\ x_{m1}Y & \dots & x_{mn}Y \end{pmatrix}$$

Si dimostra che, per tre generiche matrici X, Y, Z, vale
$vec(XZY) = (Y^T \otimes X) vec(Z)$; allora

$$B, D = argmin \Big\| vec(\mathcal{P}) - \Big[\sum_{k=1}^{i} \mathcal{Q}_k^T \otimes \mathcal{N}_k \Big] vec\left(\begin{smallmatrix} D \\ B \end{smallmatrix}\right) \Big\|_F$$

quindi

$$vec\left(\begin{smallmatrix} D \\ B \end{smallmatrix}\right) = \Big[\sum_{k=1}^{i} \mathcal{Q}_k^T \otimes \mathcal{N}_k \Big]^{+} vec(\mathcal{P})$$

Riassumendo, questo terzo algoritmo si articola nei seguenti punti:

Algoritmo robusto

1) Si calcolano le matrici

$$O_i := Y_f/_{U_f}W_p \qquad Z_i := Y_f/\binom{W_p}{U_f} \qquad Z_{i+1} := Y_{f^-}/\binom{W_{p^+}}{U_{f^-}}$$

2) Eseguiamo la decomposizione a valori singolari pesata

$$O_i\Pi_{U_f^\perp} = USV^T$$

3) Poniamo l'ordine $n := rk(S_1)$ e determiniamo e le matrici U_1 e S_1

4) Si calcolano le matrici

$$\Gamma_i := U_1 S_1^{1/2} \qquad \Gamma_{i-1} := \underline{\Gamma_i}$$

5.1) Si calcolano le matrici A, C e $\mathcal{K}$ risolvendo ai minimi quadrati il sistema lineare

$$\begin{pmatrix} \Gamma_{i-1}^+ Z_{i+1} \\ Y_{i|i} \end{pmatrix} = \left(\begin{array}{c|c} A & \mathcal{K} \\ C & \end{array}\right) \begin{pmatrix} \Gamma_i^+ Z_i \\ U_f \end{pmatrix}$$

5.2) Se la matrice A calcolata risulta instabile e vogliamo forzarne la stabilità, calcoliamo A e C sfruttando una delle tecniche illustrate alla sezione 3.5

6) Sfruttando le matrici A e C appena calcolate, ricalcoliamo Γ_i e Γ_{i-1}

7) Calcoliamo B e D risolvendo l'equazione

$$vec\binom{D}{B} = \Big[\sum_{k=1}^{i} \mathcal{Q}_k^T \otimes \mathcal{N}_k\Big]^+ vec(\mathcal{P})$$

3.5 Il problema della stabilità

Come abbiamo già detto, può capitare che, benchè la matrice A del sistema *state-space* da stimare sia stabile, la sua stima attraverso i metodi *subspace* risulti instabile. In [5] e [6] sono riportate delle tecniche alternative che forzano la stabilità della matrice A stimata.

3.5.1 Primo metodo: una modifica nell'algoritmo

La prima tecnica di calcolo di A che ne garantisce la stabilità si basa sull'utilizzo della matrice di osservabilità estesa.

$$\Gamma = \begin{pmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^n \\ \dots \end{pmatrix}$$

Detta infatti $\overline{\Gamma}$ la matrice ottenuta rimuovendo da Γ il primo blocco è chiaro che $\overline{\Gamma} = \Gamma A$, e quindi

$$A = \Gamma^{+}\overline{\Gamma}$$

Ovviamente, trattandosi di matrici con infinite componenti, Γ e $\overline{\Gamma}$ non sono matrici trattabili in pratica. D'altra parte i metodi subspace illustrati ci forniscono una stima della matrice ottenuta prendendo i primi i blocchi della matrice di osservabilità

$$\Gamma_i = \begin{pmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{i-1} \end{pmatrix}$$

Quindi, detta analogamente $\overline{\Gamma}_i$ la matrice ottenuta rimuovendo da Γ_i il primo blocco, possiamo scegliere di approssimare la matrice A usando la seguente formula.

$$A = \Gamma_i^{+} \begin{pmatrix} \overline{\Gamma}_i \\ 0 \end{pmatrix}$$

(mentre per calcolare C basta prendere le prime l righe della matrice Γ_i).
Il fatto che questo metodo garantisca la stabilità di A discende dal seguente teorema:

Teorema 3.5.1. *Siano $M, \overline{M}_0 \in \mathbb{R}^{Np \times n}$ due matrici a blocchi della seguente forma*

$$M = \begin{pmatrix} M_1 \\ M_2 \\ \dots \\ M_{N-1} \\ M_N \end{pmatrix} \qquad \overline{M}_0 = \begin{pmatrix} M_2 \\ M_3 \\ \dots \\ M_N \\ 0 \end{pmatrix}$$

Allora la matrice A definita nel seguente modo

$$A = M^{+}\overline{M}_0$$

é stabile, ossia tutti i suoi autovalori hanno modulo minore o uguale a 1

Dimostrazione. Supponiamo di avere a disposizione una fattorizzazione QR della matrice M, $M = QR$; allora possiamo riscrivere entrambe le matrici nel

seguente modo:

$$M = QR = \begin{pmatrix} Q_1 \\ Q_2 \\ \dots \\ Q_{N-1} \\ Q_N \end{pmatrix} R, \qquad \overline{M}_0 = \overline{Q}_0 R = \begin{pmatrix} Q_2 \\ Q_3 \\ \dots \\ Q_N \\ 0 \end{pmatrix} R$$

Innanzitutto proviamo a stimare la norma $\|Q^T\overline{Q}_0\|$, ossia il massimo valore singolare della matrice $Q^T\overline{Q}_0$. Ricordiamo che, data una generica matrice P, i suoi valori singolari sono dati dalle radici quadrate degli autovalori della matrice P^TP. Cerchiamo dunque di capire come é fatta la matrice $(Q^T\overline{Q}_0)^T(Q^T\overline{Q}_0)$ e di determinarne i valori singolari.

$$(Q^T\overline{Q}_0)^T(Q^T\overline{Q}_0) = \overline{Q}_0^T\overline{Q}_0 = \sum_{k=2}^{N} Q_k^TQ_k = \mathbb{I} - Q_1^TQ_1$$

Tutti gi autovalori di $Q_1^TQ_1$ sono uguali a 0 oppure a 1, di conseguenza lo saranno anche quelli di $\mathbb{I} - Q_1^TQ_1$. Pertanto

$$\|Q^T\overline{Q}_0\| = 1$$

Osserviamo ora che la matrice A si ottiene dal conto

$$A = R^+Q^T\overline{Q}_0R$$

Siano ora λ e w autovalore e autovettore di A; abbiamo dunque che

$$\lambda Rw = RAw = RR^+Q^T\overline{Q}_0Rw$$

quindi

$$|\lambda|\|Rw\| = \|RR^+Q^T\overline{Q}_0Rw\| \leq \|RR^+\|\|Q^T\overline{Q}_0\|\|Rw\| = \|Rw\|$$

quindi

$$|\lambda| \leq 1$$

□

3.5.2 Secondo metodo: ottimizzazione vincolata

Per evitare ambiguitá di notazione, da questo momento utilizzeremo la notazione ""$M \succ 0$" per indicare una matrice definita positiva; analogamente per indicare una matrice semidefinita positiva utilizzeremo la notazione ""$M \succeq 0$".
Riportiamo dunque un lemma:

Lemma 3.5.2. *[7]*
Data una generica matrice A si dice g-inversa una qualunque matrice A^{-1_g} tale che $AA^{-1_g}A = A$. [2]
Sia M la seguente matrice Hermitiana

$$M := \begin{pmatrix} A & B \\ B^H & C \end{pmatrix} \succeq 0$$

[2]Si noti che se A è invertibile, A^{-1} é una g-inversa e che anche la pseudoinversa di Moore-Penrose è una g-inversa

Denotiamo con il simbolo M/A il complemento di Schur generalizzato $M/A = D - CA^{-1_g}B$. Allora vale che

- *$M \succeq 0$ se e solo se $A \succeq 0$,$M/A \succeq 0$, $B = AA^{-1_g}B$*
- *$M \succeq 0$ se e solo se $C \succeq 0$,$M/C \succeq 0$, $B^H = CC^{-1_g}B^H$*

Noi saremo in particolare interssati al fatto che

$$M \succeq 0 \Longrightarrow A \succeq 0, M/C \succeq 0$$

Calcolo di A dati gli stati

Il calcolo delle matrici del sistema avviene essenzialmente risolvendo ai minimi quadrati il sistema lineare

$$\begin{pmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} \hat{X}_i \\ U_{i|i} \end{pmatrix}$$

Più in generale si può scegliere di introdurre dei pesi nel sistema e che quindi il problema consista nel minimizzare il seguente funzionale

$$\begin{aligned} J(A,B,C,D) :&= \left\| \begin{pmatrix} L_x & 0 \\ 0 & L_y \end{pmatrix} \left[\begin{pmatrix} X_{i+1} \\ Y_{i|i} \end{pmatrix} - \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} X_i \\ U_{i|i} \end{pmatrix} \right] R_x \right\|_F^2 \\ &= \left\| L_x \left[X_{i+1} - \begin{pmatrix} A & B \end{pmatrix} \begin{pmatrix} X_i \\ U_{i|i} \end{pmatrix} \right] R_x \right\|_F^2 + \left\| L_y \left[Y_{i|i} - \begin{pmatrix} C & D \end{pmatrix} \begin{pmatrix} X_i \\ U_{i|i} \end{pmatrix} \right] R_x \right\|_F^2 \\ &= J_1(A,B) + J_2(C,D) \end{aligned}$$

I due funzionali sono indipendenti e possiamo quindi minimizzarli separatamente.
Per imporre la positiva definizione della matrice A vorremmo utilizzare la condizione fornitaci dal criterio di Lyapunov (si veda alla sezione 1.3.1). Tali condizioni tuttavia fanno sì che gli insiemi in cui possiamo scegliere P e A siano aperti, ma per garantire l'esistenza di un ottimo in un problema di ottimizzazione convessa occorre lavorare con insiemi chiusi. Quindi, preso $\delta > 0$, useremo le condizioni date dall'equazione di Lyapunov

$$P - \delta\mathbb{I} \succeq 0 \qquad (P - \delta\mathbb{I}) - APA^T \succeq 0$$

Dal lemma 3.5.2 tuttavia sappiamo che una condizione sufficiente affinchè ciò valga è

$$\begin{pmatrix} P - \delta\mathbb{I} & AP \\ PA^T & P \end{pmatrix} \succeq 0$$

Poniamo ora che R_x abbia la seguente forma

$$R_x = \begin{pmatrix} X_i \\ U_{i|i} \end{pmatrix}^{-1_R} \begin{pmatrix} PR_{x_1} & 0 \\ 0 & R_{x_2} \end{pmatrix}$$

e definiamo le matrici X_1, X_2 nel seguente modo

$$[X_1, X_2] := X_{i+1} \begin{pmatrix} X_i \\ U_{i|i} \end{pmatrix}^{-1_R}$$

Allora il funzionale $J_1(A,B)$ assume la seguente forma

$$\begin{aligned} J_1(A,B) &= \left\| L_x \left[(X_1, X_2) - (A,B) \right] \begin{pmatrix} PR_{x_1} & 0 \\ 0 & R_{x_2} \end{pmatrix} \right\|_F^2 \\ &= \|L_x(X_1 - A)PR_{x_1}\|_F^2 + \|L_x(X_2 - B)R_{x_2}\|_F^2 \\ &= J_{11}(A) + J_{12}(B) \end{aligned}$$

Il funzionale $J_{12}(B)$ viene minimizzato ponendo $B = X_2$; resta quindi da minimizzare solo il funzionale $J_{11}(A)$. Definiamo la matrice $Q := AP$, allora il problema diventa

$$\begin{aligned} \min \quad & J_{11}(A) := \|L_x(X_1 P - Q)R_{x_1}\|_F^2 \\ \text{s.t.} \quad & \begin{pmatrix} P - \delta\mathbb{I} & Q \\ Q^T & P \end{pmatrix} \succeq 0 \end{aligned}$$

Il problema scritto in questa forma é un problema di ottimizzazione quadratico, vogliamo dunque riscriverlo perché diventi un problema di ottimizzazione lineare. Definiamo i seguenti matrici e vettori

$$Z_1 := J_{11}(A) \qquad Z_2 := L_x(X_1 P - Q)R_{x_1} \qquad Z_3 := \begin{pmatrix} P - \delta\mathbb{I} & Q \\ Q^T & P \end{pmatrix}$$

$$z_i := vec(Z_i) \qquad z = [z_1, z_2^T, z_3^T] \qquad c_x = [1, 0, \ldots, 0]^T$$

Allora il problema diventa

$$\begin{aligned} \min \quad & c_x^T z \\ \text{s.t.} \quad & L_x \begin{pmatrix} -\mathbb{I} & X_1 \end{pmatrix} Z_3 \begin{pmatrix} 0 \\ R_{x_1} \end{pmatrix} = Z_2 \\ & \begin{pmatrix} 0 & \mathbb{I} \end{pmatrix} Z_3 \begin{pmatrix} 0 \\ \mathbb{I} \end{pmatrix} - \delta\mathbb{I} = \begin{pmatrix} \mathbb{I} & 0 \end{pmatrix} Z_2 \begin{pmatrix} \mathbb{I} \\ 0 \end{pmatrix} \\ & Z_1 \geq \|Z_2\|_F \\ & Z_3 \succeq 0 \end{aligned}$$

Calcolo di A data la matrice di osservabilità

Supponiamo invece di aver a disposizione una matrice di osservabilità Γ_i. Allora è chiaro che valga

$$\underline{\Gamma}_i = \overline{\Gamma}_i A$$

Quindi un modo ragionevole per calcolare A é minimizzare il funzionale

$$J_\Gamma(A) = \|L_\Gamma(\underline{\Gamma}_i - \overline{\Gamma}_i A)R_\Gamma\|_F^2$$

Anche stavolta occorre imporre la condizione di stabilità asintotica di A; perciò imponiamo che la matrice R_Γ abbia forma

$$R_\Gamma = PR_{\Gamma,1}$$

Quindi, definita come prima la matrice $Q := AP$, il problema diventa

$$\begin{aligned} \min \quad & J_\Gamma := \|L_\Gamma(\underline{\Gamma}_i P - \overline{\Gamma}_i Q)R_{\Gamma,1}\|_F^2 \\ \text{s.t.} \quad & \begin{pmatrix} P - \delta\mathbb{I} & Q \\ Q^T & P \end{pmatrix} \succeq 0 \end{aligned}$$

Definiamo dunque i seguenti matrici e vettori

$$Z_1 := J_\Gamma \qquad Z_2 := L_\Gamma(\underline{\Gamma}_i P - \overline{\Gamma}_i Q)R_{\Gamma,1} \qquad Z_3 := \begin{pmatrix} P - \delta\mathbb{I} & Q \\ Q^T & P \end{pmatrix}$$

$$z_i := vec(Z_i) \qquad z = [z_1, z_2^T, z_3^T] \qquad c_\Gamma = [1, 0, \ldots, 0]^T$$

Allora il problema diventa

$$\begin{aligned} \min \quad & c_\Gamma^T z \\ \text{s.t.} \quad & L_\Gamma \begin{pmatrix} -\underline{\Gamma}_i & \overline{\Gamma}_i \end{pmatrix} Z_3 \begin{pmatrix} 0 \\ R_{\Gamma,1} \end{pmatrix} = Z_2 \\ & \begin{pmatrix} 0 & \mathbb{I} \end{pmatrix} Z_3 \begin{pmatrix} 0 \\ \mathbb{I} \end{pmatrix} - \delta\mathbb{I} = \begin{pmatrix} \mathbb{I} & 0 \end{pmatrix} Z_2 \begin{pmatrix} \mathbb{I} \\ 0 \end{pmatrix} \\ & Z_1 \geq \|Z_2\|_F \\ & Z_3 \succeq 0 \end{aligned}$$

Capitolo 4

Analisi numerica dei metodi *subspace*

risovere In questo capitolo si intende analizzare più nel dettaglio il funzionamento dei metodi *subspace* usandoli per stimare le matrici A, B, C, D di modelli di ordine 1 e 2 il più possibile semplificati (gran parte delle analisi saranno fatte, ad esempio, prendendo come ingresso il campione unitario). Particolare attenzione sarà rivolta all'analisi degli indici di condizionamento delle matrici principali su cui si basa il metodo e sull'influenza che matrici malcondizionate possono avere sul risultato finale.

Le matrici che considereremo sono le matrici di Hankel dei dati di input e di output

$$U_{0|2i-1} = \begin{bmatrix} u_0 & u_1 & u_2 & \cdots & u_{j-1} \\ u_1 & u_2 & u_3 & \cdots & u_j \\ \vdots & \vdots & \vdots & & \vdots \\ u_{2i-1} & u_{2i} & u_{2i+1} & \cdots & u_{2i+j-2} \end{bmatrix}$$
$$Y_{0|2i-1} = \begin{bmatrix} y_0 & y_1 & y_2 & \cdots & y_{j-1} \\ y_1 & y_2 & y_3 & \cdots & y_j \\ \vdots & \vdots & \vdots & & \vdots \\ y_{2i-1} & y_{2i} & y_{2i+1} & \cdots & y_{2i+j-2} \end{bmatrix} \tag{4.1}$$

(dove j sarà fissato pari a $N - 2i + 1$, dove N corrisponde al numero di istanti temporali discreti per cui osserviamo il fenomeno), e la matrice del sistema lineare che risolto ai minimi quadrati fornisce la stima delle matrici A e C.

Occorre a tal proposito fare una precisazione. Da un punto di vista teorico, il calcolo delle matrici del sistema A, B, C, D avviene risolvendo ai minimi quadrati il sistema

$$\begin{pmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} \hat{X}_i \\ U_{i|i} \end{pmatrix}$$

La matrice del sistema di cui dovremmo dunque analizzare il condizionamento è $\binom{\hat{X}_i}{U_{i|i}}$.Tuttavia, se usiamo l'algoritmo robusto, il sistema che risolviamo in

pratica è

$$\begin{pmatrix} \Gamma_{i-1}^{+} Z_{i+1} \\ Y_{i|i} \end{pmatrix} = \left(\begin{array}{c|c} A & \\ C & \mathcal{K} \end{array}\right) \begin{pmatrix} \Gamma_i^{+} Z_i \\ U_f \end{pmatrix} + \begin{pmatrix} \rho_w \\ \rho_v \end{pmatrix}$$

Tuttavia nel codice proposto in [1] viene utilizzata ancora un'altra formulazione: si può vedere infatti che tutte le espressioni dell'algoritmo si semplificano notevolmente utilizzando lo strumento della decomposizione RQ (per uniformarci alla notazione in [1] parleremo però di *decoposizione* RQ^T, indicando quindi con il simbolo ""Q^T" l'usuale Q; analogamente parleremo di *decomposizione* QR^T intendendo la classica decomposizione QR). Riportiamo qui un breve cenno a questa formulazione, che sarà trattata nel dettagio alla sezione B.2: decomponendo in tal modo la matrice

$$\frac{1}{\sqrt{j}} \begin{pmatrix} U_{0|2i-1} \\ Y_{0|2i-1} \end{pmatrix} = RQ^T$$

si riesce a esprimere tutti i calcoli in termini solo di sottomatrici della matrice R con notevoli vantaggi in termini di costi computazionali e costi di memoria; in particolare, il sistema soprastante assume la seguente forma

$$\begin{pmatrix} \Gamma_{i-1}^{+} R_{[6:6][1:5]} \\ R_{[5:5][1:5]} \end{pmatrix} = \left(\begin{array}{c|c} A & \\ C & \mathcal{K} \end{array}\right) \begin{pmatrix} \Gamma_i^{+} [R_{[5:6][1:4]}, \mathbb{O}_{li \times l}] \\ R_{[2:3][1:5]} \end{pmatrix} \tag{4.2}$$

dove $R_{[6:6][1:5]}, R_{[5:5][1:5]}, R_{[5:6][1:4]}$ e $R_{[2:3][1:5]}$ sono appunto sottomatrici di R. Quindi nella pratica il sistema a cui noi stiamo facendo riferimento quando parliamo di malcondizionamento è quest'ultimo.

4.1 Sistemi di ordine 1 con ingresso il campione unitario

Si consideri un sistema di ordine 1

$$\begin{cases} x_{k+1} = \lambda x_k + u_k \\ y_k = x_k \end{cases}$$

in questo caso le matrici A, B, C, D sono semplicemente degli scalari, precisamente

$$A = \lambda, \qquad B = 1, \qquad C = 1, \qquad D = 0$$

Vogliamo testare le proprietà dei metodi subspace e la loro capacità di stimare bene le matrici A, B, C, D al variare del numero λ. Come già osservato, il sistema non ci restituirà esattamente le matrici del sistema ma ci restituirà la matrici di un qualunque sistema algebricamente equivalente a quello da stimare. Tuttavia la situazione in cui ci poniamo è quella in cui $C = 1$; pertanto se l'algoritmo ci restituisce il sistema identificato $\{A_s, B_s, C_s, D_s\}$é chiaro che la matrice di cambio di base T per portare il sistema in base fisica è ovviamente $T = C_s^{-1}$.

Il primo tentativo che si è fatto è stato provare a utilizzare come ingresso il campione unitario, ossia

$$u_k = \begin{cases} 1, & \text{se } k = 0 \\ 0, & \forall\, k = 1, \dots, (N-1) \end{cases}$$

Facendo variare l'autovalore λ da nell'intervallo $[0.1, 2]$ (considerando quindi situazioni sia di stabilità che di instabilità), la stima della matrice A risulta sempre estremamente precisa, nonostante le matrici di Hankel degli input e degli output e la matrice del sistema (4.2) siano singolari. Ciò che invece emerge è che la matrice stimata e riportata in base fisica B_f risulta sempre nulla.

4.1.1 L'origine della singolarità delle matrici di Hankel

La singolarità di queste matrici è dovuta al fatto che l'ingresso scelto è prevalentemente nullo. Ricordiamo infatti che la matrice di Hankel degli input è

$$U_{0|2i-1} = \begin{bmatrix} u_0 & u_1 & u_2 & \cdots & u_{j-1} \\ u_1 & u_2 & u_3 & \cdots & u_j \\ \vdots & \vdots & \vdots & & \vdots \\ u_{2i-1} & u_{2i} & u_{2i+1} & \cdots & u_{2i+j-2} \end{bmatrix}$$

nel nostro caso quindi è chiaro che l'unica entrata non nulla della matrice è u_0. Se $u_k \in \mathbb{R}$, ovviamente una situazione analoga si ottiene per ingressi che hanno $u_k = 0 \quad \forall k > \bar{k}$ con $\bar{k} < \min\{j-1, 2i-1\}$: in tal caso ci sarà un numero di colonne e di righe completamente nulle tale da impedire che la matrice sia di rango massimo.
Consideriamo ora la matrice di Hankel degli output. Sappiamo che sostituendo ricorsivamente le equazioni di un generico sistema *state-space* possiamo ottenere l'espressione del k-esimo output, cioè

$$y_k = CA^k x_0 + Du_k + \sum_{j=0}^{k-1} CA^j B u_{k-1-j}$$

che nel nostro specifico caso, in cui l'ingresso è il campione unitario, con la scelta delle matrici A, B, C, D che abbiamo fatto, diventa

$$y_k = \lambda^{k-1}(\lambda x_0 + 1)$$

Definiti dunque i seguenti vettori linearmente indipendenti

$$v := (0, 1, \lambda, \lambda^2, \dots, \lambda^{2i-2})^T$$
$$w := (1, \lambda, \lambda^2, \lambda^3 \dots, \lambda^{2i-1})^T$$

si verifica facilmente che le colonne della matrice di Hankel degli output hanno forma

$$Y_{0|2i-1} = [x_0 v + w, (\lambda x_0 + 1)v, \lambda(\lambda x_0 + 1)v, \lambda^2(\lambda x_0 + 1)v, \dots, \lambda^{j-2}(\lambda x_0 + 1)v]$$

Pertanto tale matrice ha necessariamente rango 2.
Per quanto riguarda la matrice del sistema (4.2), nel caso generale in cui

$u_0, \dots, u_{N-1} \in \mathbb{R}^m$ e $y_0, \dots, y_{N-1} \in \mathbb{R}^l$, essa ha le seguenti dimensioni

$$\underbrace{\begin{pmatrix} \Gamma_i^+ [R_{[5:6][1:4]}, \mathbb{O}_{li \times l}] \\ R_{[2:3][1:5]} \end{pmatrix}}_{2mi+l(i+1)} \begin{matrix} n \\ mi \end{matrix}$$

siccome la matrice $R_{[2:3][1:5]}$ è essenzialmente la matrice 'R' della decomposizione RQ^T della parte inferiore della matrice $U_{0|2i-1}$ (per i dettagli si rimanda nuovamente alla sezione B.2), essa è nulla, pertanto è palese la ragione della singolarità di questa seconda matrice.

4.1.2 La ragione del calcolo esatto di λ

Cerchiamo di capire perchè nonostante questa situazione di singolarità la stima della matrice A avvenga il modo perfetto.

L'espressione esplicita dell'equazione per il calcolo di A

Cominciamo con l'osservare che, visto che la matrice $R_{[2:3][1:5]}$ è nulla, il sistema (4.2) è del tutto equivalente al sistema

$$\begin{pmatrix} \Gamma_{i-1}^+ R_{[6:6][1:5]} \\ R_{[5:5][1:5]} \end{pmatrix} = \begin{pmatrix} A \\ C \end{pmatrix} \Gamma_i^+ [R_{[5:6][1:4]}, \mathbb{O}_{li \times l}]$$

Pertanto lo scalare λ, che coincide con A, si ottiene risolvendo il seguente sistema nell'indeterminata $\boldsymbol{x}$

$$\Gamma_{i-1}^+ R_{[6:6][1:5]} = \boldsymbol{x} \cdot \Gamma_i^+ [R_{[5:6][1:4]}, \mathbb{O}_{li \times l}]$$

Proviamo dunque a capire come sono fatte $R_{[6:6][1:5]}$ e $R_{[5:6][1:4]}$. Tali matrici sono sottomatrici della matrice R della seguente decomposizione QR^T

$$\frac{1}{\sqrt{j}} (U_{0|2i-1}^T, Y_{0|2i-1}^T) = QR^T$$

Per capire che forma assume tale matrice R occorre ricordare come funziona l'algoritmo di Householder (per cui rimandiamo all'Appendice C). Nel nostro caso la matrice di cui dobbiamo calcolare la decomposizione QR^T é

$$\frac{1}{\sqrt{j}} (U_{0|2i-1}^T, Y_{0|2i-1}^T) = \begin{pmatrix} 1 & 0 & \dots & 0 & y_0 & y_1 & \dots & y_{2i-2} & y_{2i-1} \\ 0 & 0 & \dots & 0 & y_1 & y_2 & \dots & y_{2i-1} & y_{2i} \\ & & & & \dots & & & & \\ 0 & 0 & \dots & 0 & y_{j-1} & y_j & \dots & y_{2i+j-3} & y_{2i+j-2} \end{pmatrix}$$

Essendo le prime $2i$ colonne della matrice in questione sulle entrate subdiagonali sono già nulle, le matrici $Q_1, \dots, Q_{2i}$ sono matrici identiche. La prima matrice Q_k non identica sará dunque Q_{2i+1}; per definizione, la sua sottomatrice P_{2i+1} è la simmetria rispetto a un opportuno iperpiano che porta il vettore $(y_{2i}, \dots, y_{j-1})$ sul sottospazio span(e_1). Ci troviamo dunque nella seguente situazione

$$\underbrace{\begin{pmatrix} \mathbb{I}_{2i} & 0 \\ 0 & P_{2i+1} \end{pmatrix}}_{Q_{2i+1}^T} \underbrace{\begin{pmatrix} 1\ 0\ \dots\ 0 & * \\ 0\ 0\ \dots\ 0 & \\ 0\ 0\ \dots\ 0 & \\ \mathbb{O} & \mathcal{Y} \end{pmatrix}}_{\frac{1}{\sqrt{j}}(U_{0|2i-1}^T, Y_{0|2i-1}^T)} = \begin{pmatrix} 1\ 0\ \dots\ 0 & * \\ 0\ 0\ \dots\ 0 & \\ 0\ 0\ \dots\ 0 & \\ \mathbb{O} & P_{2i+1}\mathcal{Y} \end{pmatrix}$$

La matrice P_{2i+1} ruota tutte le colonne della matrice $\mathcal{Y}$; tali colonne però sono tutte multipli del vettore $(1, \lambda, \lambda^2, \dots, \lambda^{j-2i-3})^T$, pertanto se la prima colonna viene ridotta a un vettore che ha come unica entrata non nulla la prima componente, così accadrà a tutte le colonne di $\mathcal{Y}$. Quindi la matrice Q_{2i+1} è sufficiente a portare la matrice $\frac{1}{\sqrt{j}}(U^T_{0|2i-1}, Y^T_{0|2i-1})$ nella forma triangolare superiore R^T; tale matrice, per sua struttura, non toccherà le prime le prime $2i$ righe e le prime $2i$ colonne di tale matrice e annullerà le ultime $j-2i-1$ righe:

$$R^T = \frac{1}{\sqrt{j}} \left(\begin{array}{c|ccc|ccc} 1 & 0 & \dots & 0 & y_0 & \dots & y_{2i-1} \\ \hline 0 & & & & \vdots & & \vdots \\ \vdots & & \mathbb{O} & & \vdots & & \vdots \\ 0 & & & & y_{2i-1} & \dots & y_{4i-2} \\ \hline & & & & & v & \\ \cline{5-7} & & \mathbb{O} & & & \mathbb{O} & \end{array} \right)$$

$$\text{con } v = \lambda^{2i-1}(\lambda x_0 + 1)\sqrt{\frac{\lambda^{2(j-2i)} - 1}{\lambda^2 - 1}} \cdot (1, \lambda, \dots, \lambda^{2i-1})$$

dove v contiene semplicemente i moduli delle colonne di $\mathcal{Y}$.
Quindi le matrici $R_{[5:6][1:4]}$,$R_{[6:6][1:5]}$ sono cosí fatte:

$$R = \frac{1}{\sqrt{j}} \left(\begin{array}{c|ccc|c|c} 1 & 0 & \dots & 0 & & \\ \hline 0 & & & & & \\ \vdots & & \mathbb{O} & & & \mathbb{O} \\ 0 & & & & & \\ \hline y_0 & \dots & \dots & y_{2i-1} & & \\ \vdots & & & \vdots & & \mathbb{O} \\ y_{i-1} & \dots & \dots & y_{3i-2} & & \\ y_i & \dots & \dots & y_{3i-1} & v & \mathbb{O} \\ y_{i+1} & \dots & \dots & y_{3i} & & \\ \vdots & & & \vdots & \mathbb{O} & \mathbb{O} \\ y_{2i-1} & \dots & \dots & y_{4i-2} & & \end{array} \right)$$

$R_{[5:6][1:4]}$

$R_{[6:6][1:5]}$

$$R_{[5:6][1:4]} = \begin{pmatrix} \lambda^{i-1}(\lambda x_0+1) & \dots & \lambda^{3i-2}(\lambda x_0+1) & \lambda^{2i-1}(\lambda x_0+1)\sqrt{\alpha}\lambda^i & \\ \vdots & & \vdots & \vdots & \mathbb{O}_{i\times(i-1)} \\ \lambda^{2i-2}(\lambda x_0+1) & \dots & \lambda^{4i-3}(\lambda x_0+1) & \lambda^{2i-1}(\lambda x_0+1)\sqrt{\alpha}\lambda^{2i-1} & \end{pmatrix}$$

$$= (\lambda x_0+1)\lambda^{i-1}\begin{pmatrix} 1 & \dots & \lambda^{2i-1} & \lambda^{2i}\sqrt{\alpha} & \\ \vdots & & \vdots & \vdots & \mathbb{O}_{i\times(i-1)} \\ \lambda^{i-1} & \dots & \lambda^{3i-2} & \lambda^{3i-1}\sqrt{\alpha} & \end{pmatrix}$$

$$R_{[6:6][1:5]} = \begin{pmatrix} \lambda^{i}(\lambda x_0+1) & \dots & \lambda^{3i-1}(\lambda x_0+1) & \lambda^{2i-1}(\lambda x_0+1)\sqrt{\alpha}\lambda^{i+1} & \\ \vdots & & \vdots & \vdots & \mathbb{O}_{(i-1)\times i} \\ \lambda^{2i-2}(\lambda x_0+1) & \dots & \lambda^{4i-3}(\lambda x_0+1) & \lambda^{2i-1}(\lambda x_0+1)\sqrt{\alpha}\lambda^{2i-1} & \end{pmatrix}$$

$$= (\lambda x_0+1)\lambda^{i}\begin{pmatrix} 1 & \dots & \lambda^{2i-1} & \lambda^{2i}\sqrt{\alpha} & \\ \vdots & & \vdots & \vdots & \mathbb{O}_{(i-1)\times i} \\ \lambda^{i-2} & \dots & \lambda^{3i-3} & \lambda^{3i}\sqrt{\alpha} & \end{pmatrix}$$

dove per brevitá abbiamo denotato $\alpha = \frac{\lambda^{2(j-2i)}-1}{\lambda^2-1}$. Pertanto il sistema

$$\Gamma^+_{i-1}R_{[6:6][1:5]} = \boldsymbol{x}\cdot\Gamma^+_i[R_{[5:6][1:4]}, \mathbb{O}_{li\times l}]$$

puó essere riscritto nel seguente modo, semplificando i coefficienti e le colonne nulle ad ambi i membri

$$\Gamma^+_{i-1}\lambda\begin{pmatrix} 1 & \dots & \lambda^{2i-1} & \lambda^{2i}\sqrt{\alpha} \\ \vdots & & \vdots & \vdots \\ \lambda^{i-2} & \dots & \lambda^{3i-3} & \lambda^{3i}\sqrt{\alpha} \end{pmatrix} = \boldsymbol{x}\cdot\Gamma^+_i\begin{pmatrix} 1 & \dots & \lambda^{2i-1} & \lambda^{2i}\sqrt{\alpha} \\ \vdots & & \vdots & \vdots \\ \lambda^{i-1} & \dots & \lambda^{3i-2} & \lambda^{3i-1}\sqrt{\alpha} \end{pmatrix}$$

quindi

$$\Gamma^+_{i-1}\lambda\begin{pmatrix} 1 \\ \vdots \\ \lambda^{i-2} \end{pmatrix}\left(1,\dots,\lambda^{2i-1},\lambda^{2i}\sqrt{\alpha}\right) = \boldsymbol{x}\cdot\Gamma^+_i\begin{pmatrix} 1 \\ \vdots \\ \lambda^{i-1} \end{pmatrix}\left(1,\dots,\lambda^{2i-1},\lambda^{2i}\sqrt{\alpha}\right)$$

$$\left[\Gamma^+_{i-1}\lambda\begin{pmatrix} 1 \\ \vdots \\ \lambda^{i-2} \end{pmatrix} - \boldsymbol{x}\cdot\Gamma^+_i\begin{pmatrix} 1 \\ \vdots \\ \lambda^{i-1} \end{pmatrix}\right]\left(1,\dots,\lambda^{2i-1},\lambda^{2i}\sqrt{\alpha}\right) = 0$$

quindi basterebbe risolvere l'equazione scalare

$$\Gamma^+_{i-1}\lambda\begin{pmatrix} 1 \\ \vdots \\ \lambda^{i-2} \end{pmatrix} - \boldsymbol{x}\cdot\Gamma^+_i\begin{pmatrix} 1 \\ \vdots \\ \lambda^{i-1} \end{pmatrix} = 0$$

Se denotiamo in questo modo le entrate del vettore $\Gamma_i = (g_0,\dots,g_{i-1})^T$, osserviamo che $\Gamma_i = (\Gamma^T_{i-1}, g_{i-1})^T$; possiamo ora agevolmente calcolare le pseudo-inverse di Γ_i,Γ_{i-1} grazie al seguente lemma:

Lemma 4.1.1. *Dato un vettore $v \in \mathbb{R}^n$, la sua pseudo-inversa é data da*

$$v^+ = \frac{v^T}{\|v\|^2}$$

Dimostrazione. Per una generica matrice A, ricordiamo che si definisce *pseudo-inversa di A* l'unica matrice che verifica le seguenti 4 proprietà:

1. A è regolare, cioè $AA^+A = A$
2. A^+ è regolare, cioè $A^+AA^+ = A^+$
3. $(AA^+)^H = AA^+$
4. $(A^+A)^H = A^+A$

Nel caso in cui A sia semplicemente un vettore v, è immediato vedere che $\frac{v^T}{\|v\|^2}$ verifica tutte e 4 le proprietà ed è pertanto la sua pseudo-inversa. □

Quindi le pseudo-inverse di Γ_i,Γ_{i-1} sono

$$\Gamma_i^+ = \frac{1}{\|\Gamma_{i-1}\|^2 + g_{i-1}^2}(\Gamma_{i-1}^T, g_{i-1})$$
$$\Gamma_{i-1}^+ = \frac{1}{\|\Gamma_{i-1}\|^2}\Gamma_{i-1}^T$$

detto dunque $z := (1, \ldots, \lambda^{i-2})^T$, l'equazione diventa

$$\frac{\lambda}{\|\Gamma_{i-1}\|^2}\Gamma_{i-1}^T z - \boldsymbol{x}\frac{1}{\|\Gamma_{i-1}\|^2 + g_{i-1}^2}(\Gamma_{i-1}^T z + g_{i-1}\lambda^{i-1}) = 0$$

$$\boldsymbol{x} = \lambda\frac{\|\Gamma_{i-1}\|^2 + g_{i-1}^2}{\|\Gamma_{i-1}\|^2}\frac{\Gamma_{i-1}^T z}{\Gamma_{i-1}^T z + g_{i-1}\lambda^{i-1}} \tag{4.3}$$

Si osservi che quando le matrici Γ_{i-1}, Γ_i sono stimate bene dal metodo *subspace* (e quindi risulta $\Gamma_{i-1} := (1, \ldots, \lambda^{i-2})^T$ e $\Gamma_i := (1, \ldots, \lambda^{i-1})^T$) la soluzione viene esattamente $\boldsymbol{x} = \lambda$.

Quello che si evince da questa analisi è che la singolarità delle matrici di Hankel possa essere del tutto irrilevante per via della possibilitá di ricavare decomposizioni a rango pieno di tali matrici singolari. La questione sembra quindi ridursi al problema di calcolare in modo esatto la matrice Γ_i. Cerchiamo dunque di capire se in questo specifico caso tale stima avvenga correttamente e perchè.

Il calcolo di Γ_i

Ricordiamo come avviene la stima della matrice Γ_i in generale: definita la matrice

$$O_i := Y_f/_{U_f}W_p$$

con decomposizione a valori singolari pesata

$$O_i\Pi_{U_f^\perp} = USV^T = \begin{bmatrix}U_1, U_2\end{bmatrix}\begin{pmatrix}S_1 & 0\\ 0 & 0\end{pmatrix}\begin{pmatrix}V_1^T\\ V_2^T\end{pmatrix} \tag{4.4}$$

si pone

$$\Gamma_i := U_1 S_1^{1/2}$$

Nell'Appendice B abbiamo riformulato anche il calcolo di O_i in termini di sottomatrici di R. Precisamente:

$$O_i = R_{fp} R_{pp}^{+} R_{[1,4][1:6]}$$

con

$$R_{fp} := \left[R_{[5:6][1:3]} - \operatorname{argmin}_X \| X R_{[2:3][1:3]} - R_{[5:6][1:3]} \| R_{[2:3][1:3]}, R_{[5:6][4:6]}\right]$$

$$R_{pp} := \left[R_{[1,4][1:3]} - \operatorname{argmin}_X \| X R_{[2:3][1:3]} - R_{[1,4][1:3]} \| R_{[2:3][1:3]}, R_{[1,4][4:6]}\right]$$

Osserviamo che, nel caso in cui l'ingresso sia il campione unitario (indipendentemente dall'ordine del modello scelto) tale espressione si semplifica notevolmente: risulta infatti che $R_{[2:3][1:3]} = \mathbb{O}$ (essendo una porzione della matrice R che riguarda solo gli ingressi futuri, che sono appunto nulli) quindi

$$R_{fp} := R_{[5:6][1:6]} \qquad R_{pp} := R_{[1,4][1:6]} \qquad O_i = R_{[5:6][1:6]} R_{[1,4][1:6]}^{+} R_{[1,4][1:6]}$$

e siccome, a prescindere dalla scelta di ingressi e uscite, $R_{[1,4][1:6]} = \left[R_{[1,4][1:4]}, \mathbb{O}\right]$, risulta infine che

$$\begin{aligned} O_i &= R_{[5:6][1:6]} \left[R_{[1,4][1:4]}, \mathbb{O}\right]^{+} \left[R_{[1,4][1:4]}, \mathbb{O}\right] \\ &= R_{[5:6][1:6]} \begin{pmatrix} R_{[1,4][1:4]}^{+} \\ \mathbb{O} \end{pmatrix} \left[R_{[1,4][1:4]}, \mathbb{O}\right] \\ &= \left[R_{[5:6][1:4]}, R_{[5:6][5:6]}\right] \begin{pmatrix} R_{[1,4][1:4]}^{+} R_{[1,4][1:4]} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{pmatrix} \\ &= \left[R_{[5:6][1:4]} R_{[1,4][1:4]}^{+} R_{[1,4][1:4]}, \mathbb{O}_{li}\right] \end{aligned}$$

Supponiamo che $\operatorname{rk}(R_{[1,4][1:4]}) = r$, allora

$$R_{[1,4][1:4]}^{+} R_{[1,4][1:4]} = \begin{pmatrix} \mathbb{I}_r & \mathbb{O}_{r \times (2m+l)i-r} \\ \mathbb{O}_{(2m+l)i-r \times r} & \mathbb{O}_{(2m+l)i-r} \end{pmatrix}$$

Di conseguenza

$$O_i = \left[R_{[5:6][1:4]} R_{[1,4][1:4]}^{+} R_{[1,4][1:4]}, \mathbb{O}_{li}\right] = \left[R_{[5:6][1:4]}[:, 0:r], \mathbb{O}_{li \times 2(m+l)i-r}\right]$$

Osserviamo infine che, essendo U_f matrice nulla (in quanto contiene gli input futuri che in questo caso sono tutti nulli) lo spazio delle sue righe è span(0), quindi la proiezione sul complemento ortogonale del suo spazio delle righe $\Pi_{U_f^{\perp}}$ è l'applicazione identità. Perciò

$$O_i \Pi_{U_f^{\perp}} = O_i$$

Tali conclusioni valgono sia per il sistema di ordine 1, ma in generale per sistemi di qualunque ordine in cui si sia scelto di prendere come ingresso il campione unitario.

Tornando al nostro modello di ordine 1: ricordiamo che la matrice $R_{[5:6][1:4]}$ assume la seguente forma

$$R_{[5:6][1:4]} = \begin{pmatrix} y_i & \dots & \dots & y_{3i-1} & v_i \\ y_{i+1} & \dots & \dots & y_{3i} & v_{i+1} \\ \vdots & & & \vdots & \\ y_{2i-1} & \dots & \dots & y_{4i-2} & v_{2i-1} \end{pmatrix} \mathbb{O}$$

$$\text{con } v_k = \lambda^{2i-1}(\lambda x_0 + 1)\sqrt{\frac{\lambda^{2(j-2i)} - 1}{\lambda^2 - 1}} \cdot \lambda^k$$

In questo momento non siamo interessati a determinare l'intera decomposizione a valori singolari, ma è sufficiente dedurre la forma delle matrici U_1 e S_1 della'espressione 4.4. È utile a tal proposito ricordare il significato geometrico di tali matrici: la matrice U_1 costituisce una base ortonormale per lo spazio delle colonne della matrice che si sta decomponendo; la matrice V_1 similmente ha per colonne una base ortonormale dello spazio delle righe dell matrice.
Ricordando che $y_k = \lambda^{k-1}(\lambda x_0 + 1)$, la matrice O_i ha rango 1 in quanto tutte le righe sono multiplo della prima. Pertanto la matrice U_1 sarà costituita da un'unica colonna ed è determinata in modo unico: coinciderà infatti con la prima colonna normalizzata. Discorso analogo per la matrice V_1 che coinciderà con la prima riga normalizzata. Quindi

$$U_1 = \sqrt{\frac{1-\lambda^2}{1-\lambda^{2i}}} \begin{pmatrix} 1 \\ \lambda \\ \lambda^2 \\ \vdots \\ \lambda^{i-1} \end{pmatrix}$$

$$S_1 = \sqrt{\frac{1-\lambda^{2i}}{1-\lambda^2}}\sqrt{\frac{1-\lambda^{2r}}{1-\lambda^2}}\lambda^{i-1}(\lambda x_0 + 1)$$

$$V_1^T = \sqrt{\frac{1-\lambda^2}{1-\lambda^{2r}}} \begin{pmatrix} 1 & \lambda & \lambda^2 & \dots & \lambda^{r-1} & 0 & \dots & 0 \end{pmatrix}$$

Risulta dunque

$$\Gamma_i = \left(\frac{1-\lambda^{2r}}{1-\lambda^{2i}}\right)^{1/4} \sqrt{\lambda^{i-1}(\lambda x_0 + 1)} \begin{pmatrix} 1 \\ \lambda \\ \lambda^2 \\ \vdots \\ \lambda^{i-1} \end{pmatrix}$$

ed é pertanto determinata in modo esatto a meno della moltiplicazione per uno scalare (si può verificare facilmente che la moltiplicazione per uno scalare non influenza l'esattezza della stima dell'autovalore λ in (4.3)).

4.1.3 La matrice B risulta nulla

Nelle sezioni precedenti abbiamo visto nel dattaglio perchè in questa situazione semplificata la singolarità non impedisca all'algoritmo di determinare l'unico

autovalore della matrice in maniera precisa. Apparentemente quindi il fatto che l'ingresso fosse prevalentemente nullo non ha influenza sul metodo. Abbiamo però osservato che la matrice B stimata risulta sempre identicamente nulla. Tale fenomeno ha la seguente giustificazione intuitiva: poichè per la maggiorparte degli istanti temporali l'ingresso è nullo, la matrice B ""sparisce" dalle equazioni del sistema. Tuttavia vi è anche una spiegazione algebrica ed essa risiede nel sistema che andiamo a risolvere per il calcolo di B e D:

$$B, D = argmin \left\| vec(\mathcal{P}) - \Big[\sum_{k=1}^{i} \mathcal{Q}_k^T \otimes \mathcal{N}_k \Big] vec\big(\begin{smallmatrix} D \\ B \end{smallmatrix}\big) \right\|_F$$

Ricordiamo infatti che le matrici $\mathcal{Q}_k$ non sono altro che sottomatrici della matrice U_f:

$$U_f = \begin{pmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \\ \vdots \\ \mathcal{Q}_i \end{pmatrix}$$

Le entrate di U_f però hanno come entrate gli ingressi u_k con $i \leq k \leq N-1$, che sono tutti nulli. Si osservi inoltre che tale considerazione é valida per sistemi di ordine generico. Possiamo quindi dire che

Lemma 4.1.2. *Si consideri il generico sistema* state-space

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k + Du_k \end{cases}$$

di cui si intende stimare le matrici A, B, C, D tramite i metodi subspace. *Sia N il numero di istanti temporali discreti per cui abbiamo a disposizione le misurazioni $u_0, \ldots, u_{N-1}, y_0, \ldots, y_{N-1}$ degli input e output del sistema* state-space *e sia i il parametro che determina il numero di ""blocchi riga" delle matrici di Hankel degli input e degli output (4.1).*
Allora, se l'ingresso del sistema è tale che $u_k = 0 \; \forall \, i \leq k \leq N-1$, le stime delle matrici B e D risultano nulle indipendentemente dal loro valore reale.

4.2 Sistemi di ordine 1 con altri ingressi

4.2.1 Ingresso sinusoidale

Stavolta come ingresso abbiamo preso

$$u_k = \sin(2\pi k \cdot 0.01)$$

Anche stavolta gli indici di condizionamento delle matrici risultano estremamente elevati, ma nonostante ciò la matrice A risulta stimata in modo quasi perfetto.
Per quanto riguarda la stima delle matrici B_f e D_f si presenta la seguente situazione: nel caso in cui $\lambda < 1$, ossia nel caso di un sistema stabile, le matrici vengono stimate piuttosto bene e l'errore relativo nella stima delle temperature risulta molto basso; nel caso invece di sistema instabile, le matrici B_f e D_f

tendono a esplodere. Ricordiamo che nel calcolo delle matrici B e D è coinvolta la matrice $\mathcal{P}$ definita nel seguente modo:

$$\mathcal{P} := \begin{pmatrix} \Gamma_{i-1}^{+} Z_{i+1} \\ Y_{i|i} \end{pmatrix} - \begin{pmatrix} A \\ C \end{pmatrix} \Gamma_i^{+} Z_i$$

Nel caso il sistema sia instabile, si osserva che la metà inferiore della matrice $\mathcal{P}$

$$Y_{i|i} - C\Gamma_i^{+} Z_i$$

risulta avere entrate molto grandi, che crescono al crescere di λ. Ricordiamo che il calcolo di A e C è avvenuto tramite la risoluzione ai minimi quadrati del sistema

$$\begin{pmatrix} \Gamma_{i-1}^{+} Z_{i+1} \\ Y_{i|i} \end{pmatrix} = \left(\begin{array}{c|c} A & \\ C & \mathcal{K} \end{array}\right) \begin{pmatrix} \Gamma_i^{+} Z_i \\ U_f \end{pmatrix}$$

per cui ci si aspetterebbe che le entrate di $\mathcal{P}$ siano tutte piccole. Il fatto che invece le sue entrate inferiori siano grandi indica che il calcolo di C sembra esser stato fatto in modo errato, il che si ripercuote quindi anche sul calcolo di B e D.

4.2.2 Ingresso di tipo rumore bianco

Nel caso l'ingresso sia un rumore bianco la situazione é simile alla precedente. L'unico autovalore di A viene stimato sempre in modo estremamente preciso; i condizionamenti delle matrici non sono però disastrosi come nei precedente casi: nel caso il sistema sia stabile sono bassissimi (< 10), mentre mano a mano che λ supera la soglia dell'1 il condizionamento della matrice di Hankel degli output cresce sempre di piú e con esso il condizionamento del sistema (4.2). Questo aumento sembra ripercuotersi sulle stime delle matrici B e D, che però sono ancora accettabili per $\lambda < 1.5$. La matrice $\mathcal{P}$ in questi casi presenta lo stesso problema del caso di ingresso sinusoidale, ossia un esplosione delle entrate della parte inferiore della matrice.

4.3 Sistemi di ordine 2 diagonalizzabili con ingresso il campione unitario

Consideriamo adesso un sistema di ordine 2

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = x_k \end{cases}$$

con

$$A = \begin{pmatrix} \lambda & 0 \\ 0 & \mu \end{pmatrix} \qquad B \in \mathbb{R}^2$$

Anche in questo caso l'ingresso scelto è il campone unitario

$$u_k = \begin{cases} 1, & \text{se } k = 0 \\ 0, & \forall\, k = 1, \ldots, (N-1) \end{cases}$$

4.3.1 L'origine della singolarità delle matrici di Hankel

Anche in questo caso ci troviamo in una situazione di singolarità. La matrice di Hankel degli ingressi è la medesima del caso unidimensionale, pertanto è singolare. Anche la ragione della singolarità della matrice del sistema (4.2) é del tutto analoga al caso precedente.
Consideriamo ora la matrice di Hankel degli output. Nel caso in cui l'ingresso è il campione unitario, con la scelta delle matrici A, B, C, D che abbiamo fatto, il $k-$esimo output è

$$y_k = A^{k-1}(Ax_0 + B)$$

Definite le seguenti due matrici in $\mathbb{R}^{4i\times 2}$

$$M_1 = \begin{pmatrix} \mathbb{I} \\ A \\ A^2 \\ \dots \\ A^{2i-1} \end{pmatrix} \qquad M_2 = \begin{pmatrix} \mathbb{O} \\ \mathbb{I} \\ A \\ \dots \\ A^{2i-2} \end{pmatrix}$$

si verifica facilmente che le colonne della matrice di Hankel degli output hanno forma

$$Y_{0|2i-1} = [M_1x_0+M_2B, M_1(Ax_0+B), M_1A(Ax_0+B), M_1A^2(Ax_0+B), \dots, M_1A^{2i-2}(Ax_0+B)]$$

Quindi la matrice $Y_{0|2i-1}$ ha tutte le colonne che sono combinazione lineare delle colonne di M_1 ed M_2, ha quindi al più rango 4 ed è quindi singolare come nel caso unidimensionale.

4.3.2 Autovalori distinti: la ragione del calcolo esatto

Sperimentalmente ci troviamo in una situazione simile a quella riscontrata per modelli di ordine 1: nonostante la singolarità delle matrici di Hankel, gli autovalori vengono stimati in modo estremamente preciso. Vediamo quindi nello specifico il perchè di ciò.

L'espressione esplicita dell'equazione per il calcolo di A

Analogamente a prima, il primo passo da fare é comprendere la forma della matrice R^T della decomposizione QR^T della matrice

$$\frac{1}{\sqrt{j}}(U^T_{0|2i-1}, Y^T_{0|2i-1}) = \frac{1}{\sqrt{j}} \left(\begin{array}{cccc|ccc} 1 & 0 & \dots & 0 & y_0^T & \dots & y_{2i-1}^T \\ 0 & 0 & \dots & 0 & y_1^T & \dots & y_{2i}^T \\ & & & & \dots & & \\ 0 & 0 & \dots & 0 & y_{2i-1}^T & \dots & y_{4i-1}^T \\ \hline 0 & 0 & \dots & 0 & y_{2i}^T & \dots & y_{4i}^T \\ & & & & \dots & & \\ 0 & 0 & \dots & 0 & y_{j-1}^T & \dots & y_{j+2i-2}^T \end{array}\right)$$

Analogamente a prima, poiché le prime $2i$ colonne della matrice in questione hanno le entrate subdiagonali nulle, le prime $2i$ matrici Q_k sono identiche. Ci

manca adesso da ridurre in forma tringolare la sottomatrice

$$\mathcal{Y} = \begin{pmatrix} y_{2i}^T & \dots & y_{4i}^T \\ \dots & & \\ y_{j-1}^T & \dots & y_{j+2i-2}^T \end{pmatrix} = \begin{pmatrix} (Ax_0+B)^T A^{2i-1} & \dots & (Ax_0+B)^T A^{4i-1} \\ & \dots & \\ (Ax_0+B)^T A^{j-2} & \dots & (Ax_0+B)^T A^{j+2i-3} \end{pmatrix}$$

Per brevitá denotiamo $(Ax_0+B)^T = (\alpha_0, \alpha_1)$,allora otteniamo

$$\mathcal{Y} = \begin{pmatrix} \alpha_0\lambda^{2i-1} & \alpha_1\mu^{2i-1} & \alpha_0\lambda^{2i} & \alpha_1\mu^{2i} & \dots & \alpha_0\lambda^{4i-1} & \alpha_1\mu^{4i-1} \\ \dots & & & & & & \\ \alpha_0\lambda^{j-2} & \alpha_1\mu^{j-2} & \alpha_0\lambda^{j-1} & \alpha_1\mu^{j-1} & \dots & \alpha_0\lambda^{j+2i-3} & \alpha_1\mu^{j+2i-3} \end{pmatrix} \tag{4.5}$$

Detta quindi $\mathcal{Y}_n$ la $n-$esima colonna di $\mathcal{Y}$ (con $n = 1, \dots, 2i$), si vede che

$$\mathcal{Y}_{2k+1} = \alpha_0\lambda^{2i+k-1} \begin{pmatrix} 1 \\ \lambda \\ \dots \\ \lambda^{j-2i-1} \end{pmatrix} \qquad \mathcal{Y}_{2k} = \alpha_1\mu^{2i+k-2} \begin{pmatrix} 1 \\ \mu \\ \dots \\ \mu^{j-2i-1} \end{pmatrix}$$

Purtroppo i vettori $v_d := (1, \lambda, \dots, \lambda^{j-2i-1})^T$ e $v_p := (1, \mu, \dots, \mu^{j-2i-1})^T$ sono linearmente indipendenti, quindi non basta usare un'unica matrice di Householder per rendere la matrice triangolare superiore ma ne servono 2 (che chiameremo P_1 e P_2): la prima ""sistemerá" le colonne dispari, la seconda le colonne pari. La prima delle due matrici di Householder (quella che agisce sulle colonne dispari) deve essere inoltre calcolata esplicitamente per determinare come trasforma le colonne pari.
Cerchiamo dunque la matrice P_1 tale che

$$P_1 v_d = \|v_d\| e_1$$

quindi detto

$$w := \|v_d\|e_1 - v_d = (\sqrt{\alpha}-1, -\lambda, \dots, -\lambda^{j-2i-1})^T \qquad \text{con } \alpha := \|v_d\|^2 = \sum_{k=0}^{j-2i-1} \lambda^{2k}$$

abbiamo che $P_1 = \left(\mathbb{I} - \frac{ww^T}{\|w\|^2}\right) = \left(\mathbb{I} - \frac{ww^T}{2\sqrt{\alpha}(\sqrt{\alpha}-1)}\right)$; ora

$$P_1 v_p = v_p - \frac{w^T v_p}{2\sqrt{\alpha}(\sqrt{\alpha}-1)} w = v_p - \beta w$$

$$\text{con } \beta := \frac{\sqrt{\alpha} - 1 - \sum_{k=1}^{j-2i-1} (\lambda\mu)^k}{2\sqrt{\alpha}(\sqrt{\alpha}-1)}$$

Quindi

$$P_1\mathcal{Y}_{2k+1} = \alpha_0\lambda^{2i+k-1}\sqrt{\alpha}e_1$$

$$P_1\mathcal{Y}_{2k} = \alpha_1\mu^{2i+k-2} \begin{pmatrix} 1-\beta(\sqrt{\alpha}-1) \\ \mu+\beta\lambda \\ \mu^2+\beta\lambda^2 \\ \dots \\ \mu^{j-2i-1}+\beta\lambda^{j-2i-1} \end{pmatrix}$$

Quindi, una volta applicata anche la seconda simmetria avremo

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & P_2 & \\ 0 & & & \end{pmatrix} P_1[\mathcal{Y}_{2k-1}, \mathcal{Y}_{2k}] = \begin{pmatrix} \alpha_0\lambda^{2i+k-2}\sqrt{\alpha} & \alpha_1\mu^{2i+k-2}[1-\beta(\sqrt{\alpha}-1)] \\ 0 & \alpha_1\mu^{2i+k-2}\sqrt{\sum_{t=1}^{j-2i-1}(\mu^t+\beta\lambda^t)^2} \\ 0 & 0 \\ \dots & \dots \\ 0 & 0 \end{pmatrix}$$

In conclusione

$$R = \frac{1}{\sqrt{j}} \left(\begin{array}{c|ccc|c|c} 1 & 0 & \dots & 0 & & \\ \hline 0 & & & & & \\ \vdots & & \mathbb{O} & & \mathbb{O} & \\ 0 & & & & & \\ \hline y_0 & \dots & \dots & y_{2i-1} & M_0 & \\ \vdots & & & & & \mathbb{O} \\ y_{i-1} & \dots & \dots & y_{3i-2} & M_{i-1} & \\ y_i & \dots & \dots & y_{3i-1} & M_i & \mathbb{O} \\ y_{i+1} & \dots & \dots & y_{3i} & M_{i+1} & \\ \vdots & & & \vdots & & \mathbb{O} \quad \mathbb{O} \\ y_{2i-1} & \dots & \dots & y_{4i-2} & M_{2i-1} & \end{array} \right) \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ R_{[5:6][1:4]} \\ \\ R_{[6:6][1:5]} \end{array}$$

$$\text{con } M_k = \begin{pmatrix} \alpha_0\lambda^{2i+k-1}\sqrt{\alpha} & 0 \\ \alpha_1\mu^{2i+k-1}\gamma & \alpha_1\mu^{2i+k-1}\delta \end{pmatrix} \qquad \begin{aligned} \gamma &:= [1-\beta(\sqrt{\alpha}-1)] \\ \delta &:= \sqrt{\sum_{t=1}^{j-2i-1}(\mu^t+\beta\lambda^t)^2} \end{aligned}$$

Quindi si può notare che

$$[R_{[5:6][1:4]}, \mathbb{O}_{li\times l}] = \frac{1}{\sqrt{j}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \lambda & 0 \\ 0 & \mu \\ \dots & \dots \\ \lambda^{i-1} & 0 \\ 0 & \mu^{i-1} \end{pmatrix} \begin{pmatrix} \lambda^{i-1}\alpha_0 & \dots & \lambda^{3i-2}\alpha_0 & \lambda^{3i-1}\alpha_0\sqrt{\alpha} & 0 & \mathbb{O}_{1\times 2(i-1)} \\ \mu^{i-1}\alpha_1 & \dots & \mu^{3i-2}\alpha_1 & \mu^{3i-1}\alpha_1\gamma & \mu^{3i-1}\alpha_1\delta & \mathbb{O}_{1\times 2(i-1)} \end{pmatrix}$$

$$R_{[6:6][1:5]} = \frac{1}{\sqrt{j}} \begin{pmatrix} \lambda & 0 \\ 0 & \mu \\ \dots & \dots \\ \lambda^{i-1} & 0 \\ 0 & \mu^{i-1} \end{pmatrix} \begin{pmatrix} \lambda^{i-1}\alpha_0 & \dots & \lambda^{3i-2}\alpha_0 & \lambda^{3i-1}\alpha_0\sqrt{\alpha} & 0 & \mathbb{O}_{1\times 2(i-1)} \\ \mu^{i-1}\alpha_1 & \dots & \mu^{3i-2}\alpha_1 & \mu^{3i-1}\alpha_1\gamma & \mu^{3i-1}\alpha_1\delta & \mathbb{O}_{1\times 2(i-1)} \end{pmatrix}$$

Pertanto, rinominando

$$W_1 := \begin{pmatrix} \lambda & 0 \\ 0 & \mu \\ \dots & \dots \\ \lambda^{i-1} & 0 \\ 0 & \mu^{i-1} \end{pmatrix}$$

$$W_2 := \begin{pmatrix} \lambda^{i-1}\alpha_0 & \dots & \lambda^{3i-2}\alpha_0 & \lambda^{3i-1}\alpha_0\sqrt{\alpha} & 0 & \mathbb{O}_{1\times 2(i-1)} \\ \mu^{i-1}\alpha_1 & \dots & \mu^{3i-2}\alpha_1 & \mu^{3i-1}\alpha_1\gamma & \mu^{3i-1}\alpha_1\delta & \mathbb{O}_{1\times 2(i-1)} \end{pmatrix}$$

il sistema nell'incognita $\boldsymbol{X}$

$$\Gamma^+_{i-1}R_{[6:6][1:5]} - \boldsymbol{X}\cdot\Gamma^+_i[R_{[5:6][1:4]}, \mathbb{O}_{li\times l}] = 0$$

diventa

$$\left[\Gamma^+_{i-1}W_1 - \boldsymbol{X}\cdot\Gamma^+_i\begin{pmatrix}\mathbb{I}\\ W_1\end{pmatrix}\right]\cdot\frac{1}{\sqrt{j}}W_2 = 0$$

Mostriamo adesso che se le matrici Γ_{i-1} e Γ_i sono calcolate in modo esatto la soluzione del sistema é esattamente $\boldsymbol{X} = A$.
Se le matrici fossero esatte avremmo che

$$\Gamma_i = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \lambda & 0 \\ 0 & \mu \\ \dots & \dots \\ \lambda^{i-1} & 0 \\ 0 & \mu^{i-1} \end{pmatrix} \Longrightarrow \Gamma_i = \begin{pmatrix}\mathbb{I}\\ W_1\end{pmatrix} \qquad \Gamma_{i-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \lambda & 0 \\ 0 & \mu \\ \dots & \dots \\ \lambda^{i-2} & 0 \\ 0 & \mu^{i-2} \end{pmatrix} \Longrightarrow \Gamma_{i-1}A = W_1$$

Quindi il sistema che dobbiamo risolvere diventa

$$\left[\Gamma^+_{i-1}\Gamma_{i-1}A - \boldsymbol{X}\cdot\Gamma^+_i\Gamma_i\right]\cdot\frac{1}{\sqrt{j}}W_2 = 0$$

Osserviamo però che le matrici Γ_i e Γ_{i-1} hanno rango per colonne massimo, quindi $\Gamma^+_{i-1}\Gamma_{i-1} = \Gamma^+_i\Gamma_i = \mathbb{I}_2$. Il sistema diventa dunque

$$\left[A - \boldsymbol{X}\right]\cdot\frac{1}{\sqrt{j}}W_2 = 0$$

Dobbiamo quindi cercare la matrice $\boldsymbol{X}$ in modo che le righe della matrice $(A-\boldsymbol{X})$ appartengano al nucleo sinistro di W_2. Ma siccome W_2 ha tutte le righe linearmente indipendenti, ha anche nucleo sinistro banale, quindi l'unica soluzione del sistema è $\boldsymbol{X} = A$.

Il calcolo di Γ_i

Come già osservato per il sistema di ordine 1, la matrice O_i dalla cui decomposizione a valori singolari si ricava Γ_i, assume forma

$$O_i = \left[R_{[5:6][1:4]}R^+_{[1,4][1:4]}R_{[1,4][1:4]}, \mathbb{O}_{li}\right] = \left[R_{[5:6][1:4]}[:, 0:r], \mathbb{O}_{li\times 2(m+l)i-r}\right]$$

dove stavolta

$$R^{\lambda\mu}_{[5:6][1:4]} = \begin{pmatrix} y_i & \dots & \dots & y_{3i-1} & M_{i,\lambda\mu} & \\ y_{i+1} & \dots & \dots & y_{3i} & M_{i+1,\lambda\mu} & \\ \vdots & & & \vdots & & \mathbb{O} \\ y_{2i-1} & \dots & \dots & y_{4i-2} & M_{2i-1,\lambda\mu} & \end{pmatrix}$$

$$\text{con } M_{k,\lambda\mu} = \begin{pmatrix} \alpha_0\lambda^{2i+k-1}\sqrt{\alpha} & 0 \\ \alpha_1\mu^{2i+k-1}\gamma & \alpha_1\mu^{2i+k-1}\delta \end{pmatrix}$$

Occorre fare una premessa. Dal punto di vista teorico l'algoritmo *subspace* dovrebbe determinare da solo l'ordine del sistema, precisamente ponendo $n := \text{rk}(O_i)$. Nella pratica però l'algoritmo permette all'utente di scegliere l'ordine. Una volta fissato l'ordine di stima n l'algoritmo definisce S_1 nel seguente modo

$$S_1 := S[0:n, 0:n]$$

Se quindi dal punto di vista teorico S_1 è la matrice diagonale contenente i valori singolari non nulli, dal punto di vista pratico S_1 è ancora diagonale ma con le ultime entrate diagonali potenzialmente nulle. Vedremo che ciò è esattamente quello che accade nel caso $\lambda = \mu$.

In questo caso la matrice $R_{[5:6][1:4]}$ ha rango 2, quindi ci sono molti modi in cui possiamo scegliere le colonne di U_1 e V_1 in modo che siano una base ortonormale dello spazio delle righe e delle colonne. É quindi difficile capire attraverso i conti algebrici se la matrice Γ_i venga calcolata bene. Tuttavia avendo noi a disposizione un'espressione algebrica della matrice O_i, é sufficiente controllare che numericamente il suo calcolo risulti esatto, come effettivamente accade.

4.3.3 Autovalori uguali: i primi problemi

Nel caso di autovalori uguali purtroppo la stima degli autovalori da parte dei metodi *subspace* risulta invece errata: si verifica infatti che il primo autovalore viene stimato il modo esatto, mentre il secondo risulta errato, talvolta addirittura instabile benché il valore reale che dovrebbe assumere sia inferiore a 1.

L'espressione esplicita dell'equazione per il calcolo di A

Nel caso di autovalori distinti avevamo notato che la matrice (4.5) di cui dobbiamo ricavarci la forma triangolare superiore aveva tutte le colonne multiple di uno di questi due vettori: $v_d := (1, \lambda, \ldots, \lambda^{j-2i-1})^T$ e $v_p := (1, \mu, \ldots, \mu^{j-2i-1})^T$. Nel caso però di autovalori uguali i due vettori coincidono, quindi basta una sola matrice dell'algoritmo di Householder a rendere la matrice triangolare superiore, e il risultato è (analogamente a come fatto per il sistema di ordine 1)

$$R = \frac{1}{\sqrt{j}} \left(\begin{array}{c|ccc|c|c} 1 & 0 & \dots & 0 & & \\ \hline 0 & & & & & \\ \vdots & & \mathbb{O} & & & \mathbb{O} \\ 0 & & & & & \\ \hline y_0 & \dots & \dots & y_{2i-1} & M_0 & \\ \vdots & & & & & \mathbb{O} \\ y_{i-1} & \dots & \dots & y_{3i-2} & M_{i-1} & \\ y_i & \dots & \dots & y_{3i-1} & M_i & \mathbb{O} \\ y_{i+1} & \dots & \dots & y_{3i} & M_{i+1} & \\ \vdots & & & \vdots & & \mathbb{O} \quad \mathbb{O} \\ y_{2i-1} & \dots & \dots & y_{4i-2} & M_{2i-1} & \end{array} \right) \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ R_{[5:6][1:4]} \\ \\ R_{[6:6][1:5]} \end{array}$$

$$\text{con } M_k = \lambda^{2i+k-1}\sqrt{\alpha}\begin{pmatrix}\alpha_0 \\ \alpha_1\end{pmatrix}$$

In tal caso

$$[R_{[5:6][1:4]}, \mathbb{O}_{li\times l}] = \frac{1}{\sqrt{j}}\lambda^{i-1}\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \lambda & 0 \\ 0 & \lambda \\ \dots & \dots \\ \lambda^{i-1} & 0 \\ 0 & \lambda^{i-1} \end{pmatrix} \begin{pmatrix} \alpha_0 & \dots & \lambda^{2i-1}\alpha_0 & \lambda^{2i}\alpha_0\sqrt{\alpha} & \mathbb{O}_{1\times 2i-1} \\ \alpha_1 & \dots & \lambda^{2i-1}\alpha_1 & \lambda^{2i}\alpha_0\sqrt{\alpha} & \mathbb{O}_{1\times 2i-1} \end{pmatrix}$$

$$R_{[6:6][1:5]} = \frac{1}{\sqrt{j}}\lambda^{i-1}\begin{pmatrix} \lambda & 0 \\ 0 & \lambda \\ \dots & \dots \\ \lambda^{i-2} & 0 \\ 0 & \lambda^{i-2} \\ \lambda^{i-1} & 0 \\ 0 & \lambda^{i-1} \end{pmatrix} \begin{pmatrix} \alpha_0 & \dots & \lambda^{2i-1}\alpha_0 & \lambda^{2i}\alpha_0\sqrt{\alpha} & \mathbb{O}_{1\times 2i-1} \\ \alpha_1 & \dots & \lambda^{2i-1}\alpha_1 & \lambda^{2i}\alpha_0\sqrt{\alpha} & \mathbb{O}_{1\times 2i-1} \end{pmatrix}$$

Pertanto, rinominando

$$W_1 := \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \\ \dots & \dots \\ \lambda^{i-2} & 0 \\ 0 & \lambda^{i-2} \\ \lambda^{i-1} & 0 \\ 0 & \lambda^{i-1} \end{pmatrix} \qquad W_2 := \begin{pmatrix} \alpha_0 & \dots & \lambda^{2i-1}\alpha_0 & \lambda^{2i}\alpha_0\sqrt{\alpha} & \mathbb{O}_{1\times 2i-1} \\ \alpha_1 & \dots & \lambda^{2i-1}\alpha_1 & \lambda^{2i}\alpha_0\sqrt{\alpha} & \mathbb{O}_{1\times 2i-1} \end{pmatrix}$$

il sistema nell'incognita $\boldsymbol{X}$

$$\Gamma^+_{i-1}R_{[6:6][1:5]} - \boldsymbol{X}\cdot\Gamma^+_i[R_{[5:6][1:4]}, \mathbb{O}_{li\times l}] = 0$$

diventa

$$\left[\Gamma_{i-1}^{+}W_1 - \boldsymbol{X}\cdot\Gamma_i^{+}\begin{pmatrix}\mathbb{I}\\ W_1\end{pmatrix}\right]\cdot\frac{1}{\sqrt{j}}\lambda^{i-1}W_2 = 0 \tag{4.6}$$

Quindi, analogamente al caso $\lambda \neq \mu$, ci troviamo a dover risolvere il sistema

$$\big[A - \boldsymbol{X}\big]\cdot\frac{1}{\sqrt{j}}W_2 = 0$$

Questa volta però il nucleo sinistro di W_2 é non banale: infatti

$$\ker_{sx}(W_2) = \text{span}\left\{\begin{pmatrix}\alpha_1\\ -\alpha_0\end{pmatrix}\right\}$$

Quindi, ammesso che la matrice Γ_i sia calcolata in modo esatto, ci sono infinite soluzioni, ossia

$$\boldsymbol{X}_{k,h} = A + k\begin{pmatrix}\alpha_1 & -\alpha_0\\ 0 & 0\end{pmatrix} + h\begin{pmatrix}0 & 0\\ \alpha_1 & -\alpha_0\end{pmatrix}$$

Con un rapido conto del polinomio caratteristico, si dimostra che gli autovalori di tale matrice sono

$$\lambda \quad \text{e} \quad \lambda + (k\alpha_1 - h\alpha_0)$$

.

Da una prima analisi quindi, nel caso in cui gli autovalori non siano distinti, non abbiamo la garanzia che gli autovalori vengano calcolati in modo esatto, ma sappiamo che (sempre ammesso che Γ_i sia calcolata in modo esatto) certamente almeno uno dei due autovalori sarà determinato in modo preciso.

Il calcolo di Γ_i

In tal caso

$$R^{\lambda\mu}_{[5:6][1:4]} = \begin{pmatrix} y_i & \cdots & \cdots & y_{3i-1} & M_{i,\lambda\mu} & \\ y_{i+1} & \cdots & \cdots & y_{3i} & M_{i+1,\lambda\mu} & \\ \vdots & & & \vdots & & \mathbb{O}\\ y_{2i-1} & \cdots & \cdots & y_{4i-2} & M_{2i-1,\lambda\mu} & \end{pmatrix}$$

$$M_{k,\lambda\lambda} = \lambda^{2i+k-1}\sqrt{\alpha}\begin{pmatrix}\alpha_0 & 0\\ \alpha_1 & 0\end{pmatrix}$$

Essa ha rango 1, pertanto la decomposizione a valori singolari ""ridotta" $U_1S_1V_1^T$ dovrebbe essere *teoricamente* analoga a quella ricavata dai sistemi di ordine 1, ossia

$$U_1 = \sqrt{\frac{1}{\alpha_0^2+\alpha_1^2}}\sqrt{\frac{1-\lambda^2}{1-\lambda^{2i}}}\begin{pmatrix}\alpha_0\\ \alpha_1\\ \lambda\alpha_0\\ \lambda\alpha_1\\ \vdots\\ \lambda^{i-1}\alpha_0\\ \lambda^{i-1}\alpha_1\end{pmatrix} = \sqrt{\frac{1}{\alpha_0^2+\alpha_1^2}}\sqrt{\frac{1-\lambda^2}{1-\lambda^{2i}}}\begin{pmatrix}1 & 0\\ 0 & 1\\ \lambda & 0\\ 0 & \lambda\\ \cdots & \cdots\\ \lambda^{i-2} & 0\\ 0 & \lambda^{i-2}\\ \lambda^{i-1} & 0\\ 0 & \lambda^{i-1}\end{pmatrix}\begin{pmatrix}\alpha_0\\ \alpha_1\end{pmatrix}$$

$$S_1 = \sqrt{\alpha_0^2 + \alpha_1^2}\sqrt{\frac{1-\lambda^{2i}}{1-\lambda^2}}\sqrt{\frac{1-\lambda^{2r}}{1-\lambda^2}}\lambda^{i-1}$$

$$V_1^T = \sqrt{\frac{1-\lambda^2}{1-\lambda^{2r}}}\begin{pmatrix}1 & \lambda & \lambda^2 & \dots & \lambda^{r-1} & 0 & \dots & 0\end{pmatrix}$$

Per cui dal punto di vista teorico dovrebbe risultare

$$\Gamma_i = \left(\frac{1-\lambda^{2r}}{1-\lambda^{2i}}\right)^{1/4}\left(\frac{1}{\alpha_0^2+\alpha_1^2}\right)^{1/4}\sqrt{\lambda^{i-1}}\begin{pmatrix}1 & 0\\ 0 & 1\\ \lambda & 0\\ 0 & \lambda\\ \dots & \dots\\ \lambda^{i-2} & 0\\ 0 & \lambda^{i-2}\\ \lambda^{i-1} & 0\\ 0 & \lambda^{i-1}\end{pmatrix}\begin{pmatrix}\alpha_0\\ \alpha_1\end{pmatrix}$$

Proviamo dunque a vedere cosa accadrebbe se l'algoritmo fosse libero di stimare un sistema di ordine 1. Siccome l'algoritmo sta provando a stimare un sistema di ordine 1 non può risolvere l'equazione (4.6), che è valida solo se noi stiamo provando a stimare un sistema di ordine 2, ma deve usare l'equazione per l'ordine 1

$$\left[\Gamma_{i-1}^+ W_1 - \boldsymbol{x}\cdot\Gamma_i^+\begin{pmatrix}\mathbb{I}\\ W_1\end{pmatrix}\right]\cdot\frac{1}{\sqrt{j}}\lambda^{i-1}W_2 = 0$$

Siccome

$$\Gamma_i = c\begin{pmatrix}\mathbb{I}\\ W_1\end{pmatrix}\begin{pmatrix}\alpha_0\\ \alpha_1\end{pmatrix}\qquad \Gamma_{i-1} = \frac{c}{\lambda}W_1\begin{pmatrix}\alpha_0\\ \alpha_1\end{pmatrix}$$

$$\text{con } c := \left(\frac{1-\lambda^{2r}}{1-\lambda^{2i}}\right)^{1/4}\left(\frac{1}{\alpha_0^2+\alpha_1^2}\right)^{1/4}\sqrt{\lambda^{i-1}}$$

l'equazione diventa

$$\frac{1}{c}\begin{pmatrix}\alpha_0\\ \alpha_1\end{pmatrix}^+\left[\lambda W_1^+ W_1 - \boldsymbol{x}\cdot\begin{pmatrix}\mathbb{I}\\ W_1\end{pmatrix}^+\begin{pmatrix}\mathbb{I}\\ W_1\end{pmatrix}\right]\cdot\frac{1}{\sqrt{j}}\lambda^{i-1}W_2 = 0$$

$$\left[\lambda - \boldsymbol{x}\right]\frac{1}{\sqrt{j}}\lambda^{i-1}\frac{1}{c(\alpha_0^2+\alpha_1^2)}\begin{pmatrix}\alpha_0 & \alpha_1\end{pmatrix}W_2 = 0$$

siccome $(\alpha_0, \alpha_1)W_2 \neq 0$, l'unica soluzione possibile é $\boldsymbol{x} = \lambda$

Vediamo ora cosa succede se chiediamo all'algortimo di stimare un sistema di ordine 2.
Con questa richiesta, forziamo l'algoritmo a considerare anche il secondo valore singolare s_2, la seconda colonna u_2 di U e la seconda colonna v_2 di V, quindi la decomposizione usata non è quella appena descritta. Vediamo quindi la Γ_i realmente utilizzata dall'algoritmo. La decomposizione a valori singolari usata sarà

$$\tilde{U}_1 = [U_1, u_2]\qquad \tilde{S}_1 = \begin{bmatrix}S_1 & 0\\ 0 & 0\end{bmatrix}\qquad \tilde{V}_1 = [V_1, v_2]$$

I vettori u_2, v_2 non sono determinati in modo unico, ma sono un generico vettore ortogonale al primo; tuttavia ciò non ha importanza in quanto $s_2 = 0$, e risulta quindi semplicemente

$$\Gamma_i = \left(\frac{1-\lambda^{2r}}{1-\lambda^{2i}}\right)^{1/4}\left(\frac{1}{\alpha_0^2+\alpha_1^2}\right)^{1/4}\sqrt{\lambda^{i-1}}\begin{pmatrix}\alpha_0 & 0\\ \alpha_1 & 0\\ \lambda\alpha_0 & 0\\ \lambda\alpha_1 & 0\\ \vdots & \\ \lambda^{i-1}\alpha_0 & 0\\ \lambda^{i-1}\alpha_1 & 0\end{pmatrix}$$

$$= \left(\frac{1-\lambda^{2r}}{1-\lambda^{2i}}\right)^{1/4}\left(\frac{1}{\alpha_0^2+\alpha_1^2}\right)^{1/4}\sqrt{\lambda^{i-1}}\begin{pmatrix}1 & 0\\ 0 & 1\\ \lambda & 0\\ 0 & \lambda\\ \cdots & \cdots\\ \lambda^{i-2} & 0\\ 0 & \lambda^{i-2}\\ \lambda^{i-1} & 0\\ 0 & \lambda^{i-1}\end{pmatrix}\begin{pmatrix}\alpha_0 & 0\\ \alpha_1 & 0\end{pmatrix}$$

Proviamo quindi a sostituire tale matrice nel sistema (4.6) e vediamo se la matrice A viene stimata bene. Abbiamo ottenuto che

$$\Gamma_i = c\begin{pmatrix}\mathbb{I}\\ W_1\end{pmatrix}\begin{pmatrix}\alpha_0 & 0\\ \alpha_1 & 0\end{pmatrix} \qquad \Gamma_{i-1} = \frac{c}{\lambda}W_1\begin{pmatrix}\alpha_0 & 0\\ \alpha_1 & 0\end{pmatrix}$$

$$\text{con } c := \left(\frac{1-\lambda^{2r}}{1-\lambda^{2i}}\right)^{1/4}\left(\frac{1}{\alpha_0^2+\alpha_1^2}\right)^{1/4}\sqrt{\lambda^{i-1}}$$

Quindi l'equazione (4.6) diventa

$$\frac{1}{c}\begin{pmatrix}\alpha_0 & 0\\ \alpha_1 & 0\end{pmatrix}^{+}\left[\lambda W_1^{+}W_1 - \boldsymbol{X}\cdot\begin{pmatrix}\mathbb{I}\\ W_1\end{pmatrix}^{+}\begin{pmatrix}\mathbb{I}\\ W_1\end{pmatrix}\right]\cdot\frac{1}{\sqrt{j}}\lambda^{i-1}W_2 = 0$$

Si dimostra che $\left(\begin{smallmatrix}\alpha_0 & 0\\ \alpha_1 & 0\end{smallmatrix}\right)^{+} = \frac{1}{\alpha_0^2+\alpha_1^2}\left(\begin{smallmatrix}\alpha_0 & \alpha_1\\ 0 & 0\end{smallmatrix}\right)$ quindi otteniamo

$$\frac{1}{c(\alpha_0^2+\alpha_1^2)}\begin{pmatrix}\alpha_0 & \alpha_1\\ 0 & 0\end{pmatrix}\left[\lambda\mathbb{I}_2 - \boldsymbol{X}\right]\cdot\frac{1}{\sqrt{j}}\lambda^{i-1}W_2 = 0$$

Quindi tutte le soluzioni del sistema sono le matrici $\boldsymbol{X}$ tali che le righe di $\lambda\mathbb{I}_2 - \boldsymbol{X}$ appartengano al nucleo sinistro di W_2 oppure tali che le colonne di $\lambda\mathbb{I}_2 - \boldsymbol{X}$ appartengano al nucleo destro di $\left(\begin{smallmatrix}\alpha_0 & \alpha_1\\ 0 & 0\end{smallmatrix}\right)$. Siccome

$$\ker_{sx}(W_2) = \text{span}\left\{\begin{pmatrix}\alpha_1\\ -\alpha_0\end{pmatrix}\right\} \qquad \ker_{dx}\left(\begin{smallmatrix}\alpha_0 & \alpha_1\\ 0 & 0\end{smallmatrix}\right) = \text{span}\left\{\begin{pmatrix}\alpha_1\\ -\alpha_0\end{pmatrix}\right\}$$

Quindi ci sono infinite soluzioni, ossia

$$\boldsymbol{X}_{k,h,z,w} = A + k\begin{pmatrix}\alpha_1 & -\alpha_0\\ 0 & 0\end{pmatrix} + h\begin{pmatrix}0 & 0\\ \alpha_1 & -\alpha_0\end{pmatrix} + z\begin{pmatrix}\alpha_1 & 0\\ -\alpha_0 & 0\end{pmatrix} + w\begin{pmatrix}0 & \alpha_1\\ 0 & -\alpha_0\end{pmatrix}$$

Osservazione 1. Osserviamo che il fallimento del metodo nella stima degli autovalori in questo specifico caso era preannunciato. Sappiamo infatti che le relazioni ingresso-uscita non sono traducibili in un unico modello *state-space* ma in infiniti modelli, anche di ordini diversi. Ricordiamo che i metodi *subspace* possono individuare solo un'unica (a meno di un cambio di base) realizzazione: la realizzazione minima. Il modello che stiamo tentanto di stimare, diagonalizzabile di ordine 2 con autovalori coincidenti, *non* é una realizzazione minima in quanto crolla l'ipotesi di raggiugibilitá; infatti la matrice di raggiungibilitá di un sistema di ordine n $\mathcal{R} = [B, AB, A^2B, \dots, A^{n-1}]$ dovrebbe avere rango n, ma in questo caso, in cui $n = 2$, $B = \binom{b_0}{b_1}$, $A = \lambda \mathbb{I}_2$, risulta

$$\mathcal{R} = [B, AB] = \begin{pmatrix} b_0 & \lambda b_0 \\ b_1 & \lambda b_1 \end{pmatrix}$$

che ha rango 1.

4.4 Sistemi di ordine 2 non diagonalizzabili con ingresso il campione unitario

Vediamo cosa accade se perturbiamo leggermente l'entrata fuori diagonale di un sistema di ordine 2 diagonalizzabile, ossia consideriamo il sistema

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = x_k \end{cases}$$

con

$$A = \begin{pmatrix} \lambda & \epsilon \\ 0 & \lambda \end{pmatrix} \qquad B \in \mathbb{R}^2$$

Si noti che nel caso di $\epsilon = 1$, la matrice assume forma di blocco di Jordan.
Anche in questo caso l'ingresso scelto è il campone unitario

$$u_k = \begin{cases} 1, & \text{se } k = 0 \\ 0, & \forall\, k = 1, \dots, (N-1) \end{cases}$$

Stavolta non ci soffermeremo più di tanto sulla giustificazione della singolarità delle matrici di Hankel. Infatti la matrice degli input risulta sempre uguale a quella dei casi precedenti, mentre per la matrice degli output possiamo riprendere il discorso fatto nel caso di sistemi di ordine 2 diagonalizzabili: si osservi infatti che la discussione fatta nella sezione 4.3.1 non dipende dalle matrici A e B, ma solo dalla scelta particolare che abbiamo fatto dell'ingresso e delle matrici C e D, ed è quindi estendibile a questo caso.

4.4.1 La ragione del calcolo esatto di A

L'espressione esplicita dell'equazione per il calcolo di A

Nel caso in cui l'ingresso è il campione unitario, con la scelta delle matrici A, B, C, D che abbiamo fatto, il $k-$esimo output è

$$y_k = A^{k-1}(Ax_0 + B)$$

Si dimostra facilmente per induzione che

$$A^k = \lambda^{k-1} \begin{pmatrix} \lambda & \epsilon k \\ 0 & \lambda \end{pmatrix}$$

quindi, detto $(Ax_0 + B)^T = (\alpha_0, \alpha_1)$, risulta

$$y_k = \lambda^{k-1} \begin{pmatrix} \lambda\alpha_0 + \epsilon k \alpha_1 \\ \lambda\alpha_1 \end{pmatrix}$$

Analogamente al caso diagonalizzabile, dobbiamo innanzitutto calcolarci la decomposizione QR^T della matrice

$$\frac{1}{\sqrt{j}}(U_{0|2i-1}^T, Y_{0|2i-1}^T)$$

ma, come nel caso precedente, basta ridurre in forma triangolare solo la matrice

$$\mathcal{Y} = \begin{pmatrix} y_{2i}^T & \cdots & y_{4i}^T \\ \cdots & & \\ y_{j-1}^T & \cdots & y_{j+2i-2}^T \end{pmatrix} = [\mathcal{Y}_1, \ldots, \mathcal{Y}_{2i}]$$

con

$$\mathcal{Y}_{2k} = \alpha_1 \lambda^{2i+k-1} \begin{pmatrix} 1 \\ \lambda \\ \cdots \\ \lambda^{j-2i-1} \end{pmatrix}$$

$$\mathcal{Y}_{2k+1} = \lambda^{2i+k-1}[\alpha_0\lambda + \epsilon\alpha_1(2i+k)] \begin{pmatrix} 1 \\ \lambda \\ \cdots \\ \lambda^{j-2i-1} \end{pmatrix} + \epsilon\alpha_1 \lambda^{2i+k-1} \begin{pmatrix} 0 \\ \lambda \\ 2\lambda^2 \\ \cdots \\ (j-2i-1)\lambda^{j-2i-1} \end{pmatrix}$$

Per brevitá chiamiamo

$$v_{pd} = \begin{pmatrix} 1 \\ \lambda \\ \cdots \\ \lambda^{j-2i-1} \end{pmatrix} \qquad v_d = \begin{pmatrix} 0 \\ \lambda \\ 2\lambda^2 \\ \cdots \\ (j-2i-1)\lambda^{j-2i-1} \end{pmatrix}$$

$$c_k = \alpha_0\lambda + \epsilon\alpha_1(2i+k) \qquad c = \epsilon\alpha_1$$

Allora con questa notazione

$$\begin{aligned} \mathcal{Y}_{2k} &= \alpha_1 \lambda^{2i+k-1} v_{pd} \\ \mathcal{Y}_{2k+1} &= \lambda^{2i+k-1}(c_k v_{pd} + c v_d) \\ &= \lambda^{2i+k-1}[(c_0 + kc)v_{pd} + c v_d] \end{aligned}$$

Calcoliamo la matrice P_1 dell'algoritmo di Householder che dovrebbe annullare le entrate subdiagonali di $\mathcal{Y}_1 := \lambda^{2i-1} w_0$ e cerco di capire come agisce sui vettori $\mathcal{Y}_{2k}, \mathcal{Y}_{2k+1}$

$$w := \|w_0\| e_1 - w_0$$

$$P_1 = \left(\mathbb{I} - \frac{ww^T}{\|w\|^2} \right)$$

Siccome $\mathcal{Y}_{2k+1} = \lambda^{2i+k-1}[(c_0 + kc)v_{pd} + cv_d]$ basta che controllo cosa fa $P_1[(c_0 + kc)v_{pd} + cv_d]$; osservo preliminarmente che per costruzione $P_1 w_0 = \|w_0\| e_1$; allora

$$\begin{aligned} P_1[(c_0 + kc)v_{pd} + cv_d] &= P_1 w_0 + kcP_1 v_{pd} \\ &= \|w_0\| e_1 + kc(P_1 v_{pd}) \end{aligned}$$

Siccome $\mathcal{Y}_{2k} = \alpha_1 \lambda^{2i+k-1} v_{pd}$, in sintesi

$$\begin{aligned} P_1(\mathcal{Y}_{2k+1}) &= \lambda^{2i+k-1}[\|w_0\| e_1 + kc(P_1 v_{pd})] \\ P_1(\mathcal{Y}_{2k}) &= \alpha_1 \lambda^{2i+k-1} P_1 v_{pd} \end{aligned}$$

quindi se denotiamo $(P_1 v_{pd})^T = (z_0, z^T)$ e $\gamma_k := \|w_0\| + kcz_0$ dopo aver applicato la prima simmetria di Householder, la matrice $\mathcal{Y}$ assume la forma

$$P_1 \mathcal{Y} = \left(\begin{array}{c|cccccccccc} \|Y_1\| & \alpha_1\lambda^{2i}z_0 & \lambda^{2i}\gamma_1 & \alpha_1\lambda^{2i+1}z_0 & \dots & \lambda^{2i+k-1}\gamma_k & \alpha_1\lambda^{2i+k}z_0 & \dots & \lambda^{3i-2}\gamma_{i-1} & \alpha_1\lambda^{3i-1}z_0 \\ \hline 0 & \alpha_1\lambda^{2i}z & \lambda^{2i}cz & \alpha_1\lambda^{2i+1}z & \dots & \lambda^{2i+k-1}kcz & \alpha_1\lambda^{2i+k}z & \dots & \lambda^{3i-2}(i-1)cz & \alpha_1\lambda^{3i-1}z \end{array} \right)$$

Ora la porzione di matrice di cui dobbiamo annullare le entrate subdiagonali ha colonne tutte multiplo del vettore z e pertanto la seconda (e ultima) matrice di Householder agirá su tutte nello stesso modo, ossia:

$$P_2 P_1 \mathcal{Y} = \left(\begin{array}{c|cccccc} \|Y_1\| & \alpha_1\lambda^{2i}z_0 & \lambda^{2i}\gamma_1 & \alpha_1\lambda^{2i+1}z_0 & \dots & \lambda^{3i-2}\gamma_{i-1} & \alpha_1\lambda^{3i-1}z_0 \\ 0 & \alpha_1\lambda^{2i}\|z\| & \lambda^{2i}c\|z\| & \alpha_1\lambda^{2i+1}\|z\| & \dots & \lambda^{3i-2}(i-1)c\|z\| & \alpha_1\lambda^{3i-1}\|z\| \\ \hline 0 & & & \mathbb{O} & & \mathbb{O} & \end{array} \right)$$

In conclusione

$$R = \frac{1}{\sqrt{j}} \left(\begin{array}{c|ccc|c|c} 1 & 0 & \dots & 0 & & \\ \hline 0 & & & & & \\ \vdots & & \mathbb{O} & & & \mathbb{O} \\ 0 & & & & & \\ \hline y_0 & \dots & \dots & y_{2i-1} & M_0 & \\ \vdots & & & & & \mathbb{O} \\ y_{i-1} & \dots & \dots & y_{3i-2} & M_{i-1} & \\ y_i & \dots & \dots & y_{3i-1} & M_i & \mathbb{O} \\ y_{i+1} & \dots & \dots & y_{3i} & M_{i+1} & \\ \vdots & & & \vdots & & \mathbb{O} \quad \mathbb{O} \\ y_{2i-1} & \dots & \dots & y_{4i-2} & M_{2i-1} & \end{array} \right) \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ R_{[5:6][1:4]} \\ \\ R_{[6:6][1:5]} \end{array}$$

$$\text{con } M_0 = \begin{pmatrix} \|\mathcal{Y}_1\| & 0 \\ \alpha_1 \lambda^{2i} z_0 & \alpha_1 \lambda^{2i} \|z\| \end{pmatrix} \qquad M_k = \lambda^{2i+k-1} \begin{pmatrix} \gamma_k & kc\|z\| \\ \alpha_1 \lambda z_0 & \alpha_1 \|z\| \end{pmatrix}$$

Si può verificare che anche in questo caso le matrici $[R_{[5:6][1:4]}, \mathbb{O}_{li \times l}]$ e $R_{[6:6][1:5]}$ possono essere scomposte in un prodotto di matrici; definiamo i seguenti vettori e matrici:

$$v_k := \begin{pmatrix} \lambda^k \alpha_0 + \epsilon \lambda^{k-1} \alpha_1 k \\ \lambda^k \alpha_1 \\ \epsilon \lambda^{k-1} \alpha_1 \end{pmatrix} \qquad W_1 := \begin{pmatrix} \lambda & 0 & \lambda \\ 0 & \lambda & 0 \\ \lambda^2 & 0 & 2\lambda^2 \\ 0 & \lambda^2 & 0 \\ \lambda^3 & 0 & 3\lambda^3 \\ 0 & \lambda^3 & 0 \\ \dots & \dots & \dots \\ \lambda^{i-1} & 0 & (i-1)\lambda^{i-1} \\ 0 & \lambda^{i-1} & 0 \end{pmatrix}$$

$$W_2 := \frac{1}{\sqrt{j}} \left(\begin{array}{ccc|ccc} v_i & \dots & v_{3i-1} & \lambda^{3i-1}(\|w_0\| + icz_0) & \lambda^{3i-1} ic\|z\| & \mathbb{O}_{1 \times 2i-1} \\ & & & \lambda^{3i} \alpha_1 z_0 & \lambda^{3i-1} \alpha_1 \|z\| & \mathbb{O}_{1 \times 2i-1} \\ & & & \lambda^{3i} c z_0 & \lambda^{3i-1} c\|z\| & \mathbb{O}_{1 \times 2i-1} \end{array} \right)$$

Allora si ottiene che

$$[R_{[5:6][1:4]}, \mathbb{O}_{li \times l}] = \left(\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \hline & W_1 & \end{array} \right) W_2 \qquad R_{[6:6][1:5]} = W_1 W_2$$

Inoltre se Γ_i fosse calcolata in modo corretto, sarebbe

$$\Gamma_i = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \lambda & \epsilon \\ 0 & \lambda \\ \lambda^2 & 2\epsilon\lambda \\ 0 & \lambda^2 \\ \lambda^3 & 3\epsilon\lambda^2 \\ 0 & \lambda^3 \\ \dots & \dots \\ \lambda^{i-1} & (i-1)\epsilon\lambda^{i-2} \\ 0 & \lambda^{i-1} \end{pmatrix}$$

cioè

$$\Gamma_i = \left(\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \hline & W_1 & \end{array} \right) \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & \epsilon/\lambda \end{pmatrix}$$

$$\Gamma_{i-1} = W_1 \cdot \frac{1}{\lambda} \begin{pmatrix} 1 & -\epsilon/\lambda \\ 0 & 1 \\ 0 & \epsilon/\lambda \end{pmatrix}$$

quindi il sistema nell'incognita $\boldsymbol{X}$ che dobbiamo risolvere

$$\Gamma^+_{i-1} R_{[6:6][1:5]} - \boldsymbol{X} \cdot \Gamma^+_i [R_{[5:6][1:4]}, \mathbb{O}_{li \times l}] = 0$$

diventa

$$\left[\lambda \begin{pmatrix} 1 & -\epsilon/\lambda \\ 0 & 1 \\ 0 & \epsilon/\lambda \end{pmatrix}^+ W_1^+ W_1 - \boldsymbol{X} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & \epsilon/\lambda \end{pmatrix}^+ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ W_1 & & \end{pmatrix}^+ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ W_1 & & \end{pmatrix} \right] W_2 = 0$$

$$\left[\lambda \begin{pmatrix} 1 & -\epsilon/\lambda \\ 0 & 1 \\ 0 & \epsilon/\lambda \end{pmatrix}^+ - \boldsymbol{X} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & \epsilon/\lambda \end{pmatrix}^+ \right] W_2 = 0$$

Ma osservando che

$$\begin{pmatrix} 1 & -\epsilon/\lambda \\ 0 & 1 \\ 0 & \epsilon/\lambda \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & \epsilon/\lambda \end{pmatrix} \begin{pmatrix} 1 & -\epsilon/\lambda \\ 0 & 1 \end{pmatrix}$$

possiamo riscrivere l'equazione

$$\left[\lambda \begin{pmatrix} 1 & -\epsilon/\lambda \\ 0 & 1 \end{pmatrix}^{-1} - \boldsymbol{X} \right] \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & \epsilon/\lambda \end{pmatrix}^+ W_2 = 0$$

$$\left[\begin{pmatrix} \lambda & \epsilon \\ 0 & \lambda \end{pmatrix} - \boldsymbol{X} \right] \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{\lambda}{\sqrt{\epsilon^2+\lambda^2}} & \frac{\epsilon}{\sqrt{\epsilon^2+\lambda^2}} \end{pmatrix} W_2 = 0$$

$$\boldsymbol{X} = \begin{pmatrix} \lambda & \epsilon \\ 0 & \lambda \end{pmatrix}$$

Nel caso in cui quindi le matrici Γ_i, Γ_{i-1} siano stimate bene, la stima della matrice A avviene in modo corretto. Per verificare se il alcolo di tali matrici avvenga in modo corretto, occorrerebbe avere a disposizione (analogamente ai casi diagonalizzabili) una decomposizione a valori singolari della matrice $R_{[5:6][1:4]}$. Trattandosi però di una matrice di rango ≥ 1 (come nel caso diagonalizzabile con autovalori distinti), la decomposizione a valori singolari non é unica e anche calcolarne una delle tante possibili non é banale.

Capitolo 5

Da equazione parabolica a sistema DLTI *state-space*

Abbiamo visto nei capitoli precedenti come stimare i parametri di un modello *state-space*. Tali modelli sono spesso la versione discreta di un'equazione alle derivate parziali. Nei capitoli successivi si intende studiare i problemi inerenti all'applicazione di tali metodi, nello specifico ci si concentrerà sulla stima dei parametri dell'equazione del calore. Prima occorre quindi ricondursi dall'equazione continua a un modello discreto. In questo capitolo si illustrerà sinteticamente la procedura di discretzzazione spaziale e temporale di un'equazione parabolica. Il modello discreto dedotto servirà poi sia per generare i dati sperimentali, sia poi per testare l'efficienza degli algoritmi per la stima dei parametri a partire da tali dati simulati.
In questo capitolo sono state tralasciate tutte le giustificazioni teoriche relative all'analisi funzionale e tutte le stime di convergenza e stabilità dei metodi utilizzati per la discretizzazioni. Per una trattazione più completa si rimanda a [8] e [9].

Supponiamo di voler risolvere la seguente equazione parabolica

$$c(x)\frac{\partial u(t,x)}{\partial t} = div(k(x)\nabla u(t,x)) + f(t,x) \qquad (t,x) \in [0,T] \times \Omega \subset \mathbb{R} \times \mathbb{R}^n$$

che rappresenta la trasmissione del calore in un mezzo conduttivo con conduttività termica $k(x)$ e capacità termica $c(x)$. Per procedere alla risoluzione del problema sará necessaria l'imposizione delle condizioni iniziali; Assumeremo dunque che siano noti i valori della temperatura su una porzione del bordo del dominio $\Gamma_D \subset \partial\Omega$ e assumeremo che sia noto il flusso attraverso la porzione del bordo $\Gamma_N \subset \partial\Omega$, dove $\Gamma_D \cup \Gamma_N = \partial\Omega$; assumeremo inoltre anche che la temperatura iniziale del corpo su tutto il dominio sia nota. Tali condizioni prendono il nome di

condizioni iniziali:	$u(0,x) = u_1(x)$	
condizioni di Dirichlet:	$u(t,x) = u_2(x)$	in Γ_D
condizioni di Neumann:	$\nabla u(t,x) \cdot \vec{n} = \bar{q}(x)$	in Γ_N

dove con $\vec{n}$ indichiamo la normale uscente a Γ_N e dove $u_1(x), u_2(x)$ e $\bar{q}(x)$ sono fissate.

5.1 La discretizzazione spaziale

5.1.1 Formulazione debole e variazionale

Nella risoluzione dei problemi che coinvolgono equazioni nelle derivate parziali uno dei problemi è la ricerca di funzioni che abbiano la giusta regolaritá ($\mathcal{C}^1, \mathcal{C}^2, \dots$) su tutto il dominio su cui abbiamo definito l'equazione. Spesso inoltre tali soluzioni esistono solo localmente e per tempi finiti. Nell'ottica di utilizzare le equazioni in campo applicativo per prevedere i fenomeni fisici questo costituisce un grosso limite. Si predilige dunque la ricerca di soluzioni un po' meno regolari (ad esempio non derivabili in qualche punto) ma con altre proprietà utili dal punto di vista applicativo.
Cerchiamo dunque una soluzione $u(t,x)$ tale che per ogni t la funzione $u(t,\cdot)$ appartenga allo spazio di funzioni

$$V = H^1_{\Gamma_D}(\Omega) := \{v : \Omega \longrightarrow \mathbb{R} \text{ t.c.} \quad v, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \in L^2(\Omega), \quad v(x) = 0 \ \forall x \in \Gamma_D\}$$

È ragionevole pensare che se $u(t,x)$ risolve l'equazione del calore su quasi tutto Ω, valga allora

$$\int_\Omega \Big(c\frac{\partial u}{\partial t} - div(k\nabla u) - f\Big) v \, d\Omega = 0 \qquad \forall v \in V$$

Applicando le regole di derivazione, il teorema della divergenza e sfruttando le condizioni di Neumann e il fatto che le funzioni v si annullano sulla porzione di bordo Γ_D otteniamo

$$\begin{aligned}
\int_\Omega c\frac{\partial u}{\partial t} v \, d\Omega &= \int_\Omega div(k\nabla u) v \, d\Omega + \int_\Omega f v \, d\Omega \\
&= \int_\Omega div(k\nabla u \, v) \, d\Omega - \int_\Omega k\nabla u \nabla v \, d\Omega + \int_\Omega f v \, d\Omega \\
&= \int_\Gamma k\nabla u \, v \cdot \vec{n} \, d\Gamma - \int_\Omega k\nabla u \nabla v \, d\Omega + \int_\Omega f v \, d\Omega \\
&= \int_{\Gamma_N} kqv \, d\Gamma - \int_\Omega k\nabla u \nabla v \, d\Omega + \int_\Omega f v \, d\Omega
\end{aligned}$$

Il problema diventa dunque

Problema 1. Siano

- $V := \{v : \Omega \longrightarrow \mathbb{R} \text{ t.c.} \quad v, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \in L^2(\Omega), \quad v(x) = 0 \, \forall x \in \Gamma_D\}$
- $(c\partial_t u, v) := \int_\Omega c\partial_t u v \, d\Omega$
- $a(u,v) := \int_\Omega k\nabla u \, \nabla v \, d\Omega$
- $(f,v) := \int_\Omega f v \, d\Omega$
- $(q,v)_{\Gamma_N} := \int_{\Gamma_N} kqv \, d\Gamma$

Vogliamo trovare $u \in V$ tale che

$$(c\partial_t u, v) + a(u,v) = (f,v) + (q,v)_{\Gamma_N} \quad \forall v \in V \tag{5.1}$$

5.1.2 Formulazione agli elementi finiti (FEM)

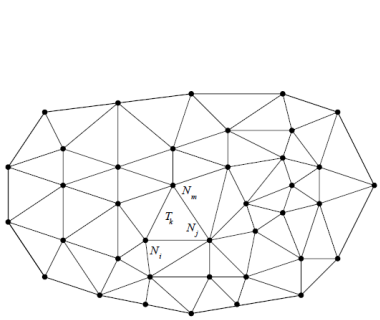
Supponiamo di star cercando la soluzione u in uno spazio V. L'idea è quella di cercare una approssimazione u_h in uno spazio $V_h \subset V$ (dove l'indice h é un parametro che indica con che accuratezza V_h approssima V: tanto più h è piccolo, tanto più accurata è l'approssimazione) in cui u_h possa essere espressa in modo "comodo".
La situazione migliore si ha quando V è uno spazio separabile ed ammette dunque una base infinita $\{\phi_1, \dots, \phi_n, \dots\}$. Questo significa che u può essere scritta come combinazione lineare infinita delle funzioni ϕ_i. Una combinazione lineare infinita non è trattabile a livello numerico, ma la scelta del sottospazio V_h è naturale: scegliamo di cercare una approssimazione u_h in uno sottospazio V_h ottenuto prendendo lo spazio generato da un numero *finito* delle funzioni di base ϕ_i in modo da poter esprimere u_h come combinazione lineare finita di funzioni $\sum_{i=1}^{n} u_i(t) \cdot \phi_i(x, y)$. Il problema si ridurrà dunque alla determinzione delle funzioni reali $u_1(t), \dots, u_n(t)$.

La scelta delle funzioni ϕ_i

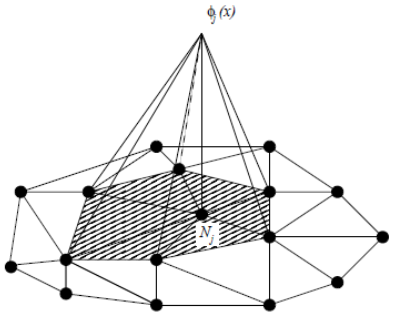
Il metodo degli elementi finiti prende il nome dal modo in cui sono costruite le funzioni ϕ_i. L'idea di base infatti é quella di discretizzare il dominio spaziale Ω dividendolo in regioni molto piccole (dette appunto elementi finiti) e scegliere le funzioni ϕ_i in modo che siano nulle su gran parte del dominio e il loro supporto sia localizzato su pochi elementi.
Nel caso unidimensionale $\Omega = [a, b] \subset \mathbb{R}$ una scelta abbastanza naturale é quella di scegliere un passo di discretizzazione $\Delta x = h = \frac{b-a}{n}$, definire i punti $x_i := a + ih$ e scegliere la funzione ϕ_i in modo che sia lineare a tratti e si annulli in tutti i nodi eccetto il nodo i-esimo in cui vale 1.
Analogamente per il caso bidimensionale, partizionamo il dominio $\Omega \in \mathbb{R}^2$ in elementi triangolari T; l'insieme dei triangoli della partizione si dice triangolazione $\mathcal{T}$. Enumeriamo poi i vertici degli elementi della triangolazione (x_i, y_i), detti nodi. Sia $h = \max\{diam(T) : T \in \mathcal{T}\}$; esso é proprio il parametro ""di precisione" nominato prima: si osservi infatti che tanto più sarà piccolo, tanti più sarano i triangoli, quindi i vertici e le funzioni, e quindi tanto più sarà accurata la approssimazione V_h di V. Volendo scegliere le funzioni ϕ_i con lo stesso criterio del caso unidimensionale, abbiamo:

$$\phi_i(x_j, y_j) := \begin{cases} 1, & \text{se } i = j \\ 0, & \text{altrimenti} \end{cases}$$

La scelta di discretizzare lo spazio in simplessi e usare funzioni lineari tuttavia non é l'unica possibile. Si possono scegliere elementi di forme diverse e funzioni polinomiali di grado superiore a seconda del problema trattato.
Si osservi dunque che la formulazione agli elementi finiti deriva in pratica da una discretizzazione spaziale del problema (si parla di semidiscretizzazione in quanto il tempo resta una variabile continua).



(a) *esempio di triangolazione.*



(b) *una funzione di base.*

Costruzione algebrica del problema FEM

Osserviamo ora che se u_h soddisfa la (5.1) per tutte le v in V_h, allora in particolare la soddisfa per tutte le ϕ_j, cioè

$$(c\partial_t u_h, \phi_j) + a(u_h, \phi_j) = (f, \phi_j) + (q, \phi_j)_{\Gamma_N} \quad \forall j = 1 \dots, n$$

Sostituiamo ora l'espressione esplicita dell'approssimazione $u_h(t) = \sum_{i=1}^{n} u_i(t) \cdot \phi_i(x, y)$ nei due membri di sinistra:

$$\begin{aligned}(c\partial_t u_h, \phi_j) &= \int_\Omega c\partial_t u_h \phi_j \, d\Omega \\ &= \int_\Omega c\frac{\partial}{\partial t}\left(\sum_{i=1}^{n} u_i(t) \cdot \phi_i\right)\phi_j \, d\Omega \\ &= \sum_{i=1}^{n} \frac{\partial u_i(t)}{\partial t} \int_\Omega c\phi_i\phi_j \, d\Omega \\ &= \sum_{i=1}^{n} \dot{u}_i(t) \cdot (c\phi_i, \phi_j)\end{aligned} \qquad \begin{aligned}a(u_h, \phi_j) &= \int_\Omega k\nabla u_h \, \nabla\phi_j \, d\Omega \\ &= \int_\Omega k\nabla\left(\sum_{i=1}^{n} u_i(t) \cdot \phi_i\right) \nabla\phi_j \, d\Omega \\ &= \sum_{i=1}^{n} u_i(t) \cdot \int_\Omega k\nabla\phi_i \, \nabla\phi_j \, d\Omega \\ &= \sum_{i=1}^{n} u_i(t) \cdot a(\phi_i, \phi_j)\end{aligned}$$

Otteniamo quindi da risolvere il

Problema 2. Siano

- $P \in M_n(\mathbb{R})$ con $p_{ij} := \int_\Omega c\phi_i\phi_j \, d\Omega$ (matrice di massa)
- $H \in M_n(\mathbb{R})$ con $h_{ij} := \int_\Omega k\nabla\phi_i\nabla\phi_j \, d\Omega$ (matrice di rigidezza)
- $\boldsymbol{u} = (u_1, \dots, u_n) \in \mathbb{R}^n$, $u_i = u_i(t)$
- $\dot{\boldsymbol{u}} = (\dot{u}_1, \dots, \dot{u}_n) \in \mathbb{R}^n$
- $b \in \mathbb{R}^n$ con $b_j := \int_\Gamma kq\phi_j \, d\Gamma + \int_\Omega f\phi_j \, d\Omega$

Vogliamo trovare $u(x) := \sum_{i=1}^{n} u_i(t)\phi_i(x)$ tale che

$$P\dot{\boldsymbol{u}} + H\boldsymbol{u} = b$$

5.1.3 La tecnica del *mass-lumping* nel caso 1-dimensionale e 2-dimensionale

Spesso, come avverrá nel nostro caso, siamo interessati a isolare $\dot{u}$ nell' equazione differenziale appena ricavata, moltiplicanto l'intera equazione per P^{-1}. Sarebbe quindi comodo avere una matrice P in forma diagonale, in modo da ridurre notevolmente sia i conti analitici che il costo numerico del calcolo dell'inversa. Occorre quindi cercare un'approssimazione P_{ML} della matrice P che sia diagonale.
Nel caso unidimensionale la scelta piú naturale é quella di non calcolare gli integrali p_{ij} in modo esatto, bensí usare una formla di quadratura, ad esempio utilizzando metodo dei trapezi, ossia

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2}(f(b)+f(a))$$

Con la scelta delle funzioni di base e degli elementi finiti che abbiamo fatto si verificherá infatti che la matrice assume forma diagonale.
Per generalizzare il procedimento al caso bidimensionale é sufficiente osservare che, nel caso unidimensionale, approssimare con la regola dei trapezi gli integrali é del tutto equivalente a definire la matrice P_{ML} nel seguente modo:

$$P_{ML} = \mathrm{diag}(d_1, \ldots, d_n), \qquad \text{con } d_i = \sum_j p_{ij}$$

In altre parole quindi si *condensano* le entrate di tutta una riga nella sola entrata diagonale, da cui il nome di *mass-lumping*. Tale procedimento) é generalizzabile al caso bidimensionale nel caso si usino elementi lineari. Per elementi finiti quadratici, invece, la procedura di somma per righe precedentemente descritta genererebbe una matrice di massa P_{ML} singolare. Una strategia di diagonalizzazione alternativa alla precedente consiste nel definire invece la matrice M_{PL} nel seguente modo:

$$P_{ML} = \mathrm{diag}(d_1, \ldots, d_n), \qquad \text{con } d_i = \frac{p_{ii}}{\sum_j p_{jj}}$$

Nel caso monodimensionale, per elementi finiti lineari e quadratici, le due definizioni della matrice P_{ML} coincidono.

5.2 La discretizzazione temporale

Come abbiamo visto, il metodo di Galerkin permette di passare da un problema di Cauchy alle derivate parziali a un sistema di equazioni differenziali ordinarie. La risoluzione numerica delle ODE utilizza dei metodi iterativi che approssimano le derivate con rapporti incrementali. Riassumiamo qui i cosiddetti θ-metodi, tralasciando le analisi di stabilitá, basti sapere che per $\theta \leq 1/2$ ci sono delle stime che impongono di prendere il passo di discretizzazione temporale Δt inferiore a un certo *bound* perché sia garantita la stabilitá.

5.2.1 I θ-metodi

Definiamo dunque questa classe di metodi dipendenti dal parametro θ.

Equazione autonoma

Supponiamo di avere una generica equazione differenziale

$$\dot{u} = f(u) \tag{5.2}$$

di cui vogliamo calcolare la soluzione per un certo intervallo temporale. Discretizziamo dunque tale intervallo in sotto-intervalli della forma $[k\Delta t, (k+1)\Delta t]$. Fissato unodi questi sotto-intervalli e scelto $\theta \in [0,1]$ usiamo le notazioni

$$u_k := u(k\Delta t), \qquad u_{k+1} := u((k+1)\Delta t)$$

$$u_{k+\theta} := \theta u_{k+1} + (1-\theta)u_k$$

Discretizzando la (5.1) otteniamo

$$\frac{u_{k+1} - u_k}{\Delta t} = f(u_{k+\theta})$$

cioé

$$\frac{u_{k+1} - u_k}{\Delta t} = f(\theta u_{k+1} + (1-\theta)u_k)$$

Equazione non autonoma

Se l'equazione é non autonoma prima di dedurre il θ-metodo dobbiamo renderla autonoma:

$$\begin{cases} \dot{u}(t) = f(u(t), t) \\ u(0) = u_0 \end{cases} \longrightarrow \quad \begin{cases} \dot{u}(t) = f(u(t), v(t)) \\ u(0) = u_0 \\ \dot{v}(t) = 1 \\ v(0) = 0 \end{cases}$$

Applicando il θ-metodo abbiamo

$$\frac{u_{k+1} - u_k}{\Delta t} = f(u_{k+\theta}, v_{k+\theta})$$
$$\frac{v_{k+1} - v_k}{\Delta t} = 1$$

La seconda equazione, come prevedibile, é tautogica in quanto $v_k = t_k$. Osserviamo infine che $v_{k+\theta} = \theta v_{k+1} + (1-\theta)v_k = t_k + \theta\Delta t$. In conclusione il θ-metodo nel caso non autonomo é

$$\frac{u_{k+1} - u_k}{\Delta t} = f(u_{k+\theta}, t_k + \theta\Delta t)$$

5.2.2 Il caso *state-space*

Quello che noi abbiamo ottento dal metodo di Galerkin é dunque un'equazione della forma

$$\dot{x}(t) = A_c x(t) + B_c u(t)$$

dove $x(t)$ rappresenta l'evoluzione della temperatura e $u(t)$ la forzante. Per applicare il metodo iterativo e risolvere l'equazione occorrerebbe conoscere la

temperatura iniziale su tutto il dominio $x(0)$. Nella pratica spesso non é disponibile una conoscenza diretta della temperatura ma sono note invece delle grandezze che si assume abbiano un legame lineare con le altre grandezze del problema. Di conseguenza il sistema assumerá forma

$$\begin{cases} \dot{x}(t) = A_c x(t) + B_c u(t) \\ y(t) = C_c x(t) \end{cases}$$

Da questi problemi continui, con una discretizzazione temporale, otteniamo appunto il modello DLTI *state-space*.
Come giá ricordato, per θ piccolo ci sono problemi di stabilitá del metodo iterativo, quindi per semplicitá utilizziamo $\theta = 1$ (detto metodo di Eulero implicito). La discretizzazione porta a un sistema di equazioni alle differenze di questo tipo:

$$\begin{cases} x_{k+1} = A_f x_k + B_f u_k \\ y_k = C_f x_k \end{cases} \qquad \text{con} \qquad \begin{aligned} A_f &= (\mathbb{I} - \Delta t A_c)^{-1} \\ B_f &= \Delta t (\mathbb{I} - \Delta t A_c)^{-1} B_c \\ C_f &= C_c \end{aligned}$$

Il metodo di Eulero implicito quindi permette di costruire una mappa che associa le matrici del sistema continuo A_c, B_c, C_c alle matrici del sistema discreto A_f, B_f, C_f. Per comoditá riportiamo anche la mappa inversa:

$$A_c = \frac{1}{\Delta t}(\mathbb{I} - A_f^{-1}) \qquad B_c = \frac{1}{\Delta t} A_f^{-1} B_f \qquad C_c = C_f$$

Capitolo 6

Stima dei parametri per l'equazione del calore unidimensionale

qullo Si vuole adesso applicare i metodi *subspace* per stimare i parametri dell'equazione del calore unidimensionale. In pratica, data una sbarretta di lunghezza L e di spessore trascurabile(modellizzata quindi con un segmento di lunghezza L), il cui generico punto $x \in [0, L]$ ha temperatura $g_0(x)$, conduttività termica $k(x)$ e capacità termica $c(x)$, immaginiamo di immettere ad una estremità del calore, ad esempio colpendo l'estremità con un flash $f_0(t)$. Ció che vogliamo fare é riuscire a stimare la capacitá termica e la conduttivitá termica nel dominio a partire dalla conoscenta dell'input f_0 e di come varia la temperatura nel dominio a seguito di questa immissione di calore. L'intero procedimento sará articolato nei seguenti passi:

- Innanzitutto bisogna generare i dati sperimentli (gli input e gli output del modello). Si considera quindi una sbarretta di caratteristiche note (capacitá e conduttivitá), si immette del calore e si registra come varia la temperatura. Non potendo fare l'esperimento vero e proprio., quello che si fa é generare tali dati numericamente. A partire quindi dall'equazione del calore, ricaveremo il modello *state-space* associato utilizzando il metodo degli elementi finiti e applicando in seguito Eulero implicito.

- una volta generati gli input e gli output, supporremo di non conoscere i parametri che descrivono il dominio e proveremo a stimarli. Questo significa stimare le matrici del sistema *state-space* discreto, convertirle in base fisica, da esse ricavare le matrici del sistema *state-space* continuo e infine dalle entrate di tali matrici ricavare i parametri di conduttivitá e capacitá termica.

6.1 Il modello per la generazione dei dati

Ricaviamo la formulazione FEM del seguente problema unidimensionale. Il problema di Cauchy che descrive l'evoluzione della temperatura per un certo

intervalli di tempo $[t_0, t_f]$ sarà dunque:

$$\begin{cases} c(x)\frac{\partial T(t,x)}{\partial t} - \frac{\partial}{\partial x}\left(k(x)\frac{\partial T(t,x)}{\partial x}\right) = 0, \quad (t,x) \in [t_0, t_f] \times [0, L] \\ T(t_0, x) = g_0(x) \\ \frac{\partial T}{\partial x}(t, 0) = -f_0(t) \\ \frac{\partial T}{\partial x}(t, L) = 0 \end{cases}$$

Il motivo per cui la condizione di Neumann riferita alla porzione di bordo $x = 0$ riporta a membro destro un segno ""-" è più chiaro se si pensa alla trascrizione generale delle condizioni di Neumann

$$k(x)\nabla u(t, x) \cdot \vec{n} = \bar{q}(x)$$

$\vec{n}$ indica la normale uscente al bordo: infatti la condizione di Neumann è riferita a un flusso *uscente*, mentre nel nostro caso abbiamo un calore *entrante*, da cui la necessità del cambio di segno del membro destro.

6.1.1 La formulazione FEM

Dividiamo l'intervallo $[0, L]$ in intervalli regolari $[x_k, x_{k+1}]$ con $x_k := \frac{L}{n} \cdot k$, dove n è fissato. Definiamo le funzioni ϕ_k come descritto prima, quindi la loro espressione analitica è

$$\phi_k := \left(1 - \frac{n}{L}|x - x_k|\right) \cdot \chi_{[x_{k-1}, x_{k+1}]}$$

Il ricavo della formulazione FEM avviene come nel caso multidimensionale: moltiplichiamo l'equazione per la funzione ϕ_i e integriamo (nel caso unidimensionale, al posto del teorema della divergenza, si usa la formula di integrazione per parti)

$$\int_0^L c\frac{\partial T}{\partial t}\phi_i \, dx - \int_0^L \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)\phi_i \, dx = 0$$

$$\int_0^L c\frac{\partial T}{\partial t}\phi_i \, dx - \left(\int_0^L \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\phi_i\right) dx - \int_0^L k\frac{\partial T}{\partial x}\frac{\partial \phi_i}{\partial x} \, dx\right) = 0$$

$$\int_0^L c\frac{\partial T}{\partial t}\phi_i \, dx + \int_0^L k\frac{\partial T}{\partial x}\frac{\partial \phi_i}{\partial x} \, dx = \int_0^L \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\phi_i\right) dx$$

$$\int_0^L c\frac{\partial T}{\partial t}\phi_i \, dx + \int_0^L k\frac{\partial T}{\partial x}\frac{\partial \phi_i}{\partial x} \, dx = \left[k\frac{\partial T}{\partial x}\phi_i\right]_0^L$$

$$\int_0^L c\frac{\partial T}{\partial t}\phi_i \, dx + \int_0^L k\frac{\partial T}{\partial x}\frac{\partial \phi_i}{\partial x} \, dx = \left(k(L)\frac{\partial T}{\partial x}(L, t)\phi_i(L)\right) - \left(k(0)\frac{\partial T}{\partial x}(0, t)\phi_i(0)\right)$$

scrivendo $T(t, x)$ nella forma $T(t, x) = \sum_{j=0}^{n+1} T_j(t)\phi_j(x)$ e applicando le condizioni di Neumann al membro destro dell'equazione si ottiene

$$\int_0^L c\frac{\partial}{\partial t}\left(\sum_j T_j\phi_j\right)\phi_i \, dx \quad + \int_0^L k\frac{\partial}{\partial x}\left(\sum_j T_j\phi_j\right)\frac{\partial \phi_i}{\partial x} \, dx \quad = k(0)f_0(t)\phi_i(0)$$

$$\sum_j \frac{\partial T_j}{\partial t}\int_0^L c\phi_j\phi_i \, dx \qquad + \sum_j T_j \int_0^L k\frac{\partial \phi_j}{\partial x}\frac{\partial \phi_i}{\partial x} \, dx \qquad = k(0)f_0(t)\phi_i(0)$$

Pertanto il sistema di ODE risultante è

$$P\dot{T} + HT = b$$

$$\text{con } T = (T_0(t), \dots, T_{n-1}(t))$$

$$p_{ij} = \int_0^L c\phi_i\phi_j\,dx \qquad a_{ij} = \int_0^L k\phi_i'\phi_j'\,dx \qquad b_i := f_0(t)k(0)\phi_i(0)$$

6.1.2 Il calcolo delle entrate delle matrici P e H

Approssimando $c(x), k(x)$ come funzioni costanti sugli elementi finiti, cioé

$$c(x) := \sum_{k=0}^{n} c_k \cdot \chi_{[x_k, x_{k+1}]}, \qquad k(x) := \sum_{k=0}^{n} k_k \cdot \chi_{[x_k, x_{k+1}]}$$

le matrici del sistema di ODE sono facilmente calcolabili.
Osserviamo intanto che, avendo le funzioni un supporto ristretto a pochi elementi finiti, gran parte degli integrali risultano nulli, infatti

$$p_{ij} = 0 \qquad \text{se } j \neq i-1, i, i+1$$
$$h_{ij} = 0 \qquad \text{se } j \neq i-1, i, i+1$$

Inoltre é chiaro dall'espressione degli integrali che le matrici siano simmetriche.
Calcoliamo prima le entrate di P:

$$\begin{aligned}
p_{ii} &= \int_0^L c(x)\phi_i^2(x)\,dx = \int_{x_{i-1}}^{x_{i+1}} c(x)\phi_i^2\,dx = c_{i-1}\int_{x_{i-1}}^{x_i} \phi_i^2\,dx + c_i\int_{x_i}^{x_{i+1}} \phi_i^2\,dx \\
&= c_{i-1}\int_{x_{i-1}}^{x_i}\left(1 - \frac{n}{L}|x - x_i|\right)^2 dx + c_i\int_{x_i}^{x_{i+1}}\left(1 - \frac{n}{L}|x - x_i|\right)^2 dx \\
&= c_{i-1}\int_{x_i - L/n}^{x_i}\left(1 + \frac{n}{L}(x - x_i)\right)^2 dx + c_i\int_{x_i}^{x_i + L/n}\left(1 - \frac{n}{L}(x - x_i)\right)^2 dx \\
&= c_{i-1}\left[x + \frac{n^2}{L^2}\frac{(x-x_i)^3}{3} + \frac{n}{L}(x-x_i)^2\right]_{x_i - L/n}^{x_i} + c_i\left[x + \frac{n^2}{L^2}\frac{(x-x_i)^3}{3} - \frac{n}{L}(x-x_i)^2\right]_{x_i}^{x_i + L/n} \\
&= c_{i-1}\frac{L}{3n} + c_i\frac{L}{3n} \\
p_{i,i+1} &= \int_0^L c(x)\phi_i(x)\phi_{i+1}(x)\,dx = c_i\int_{x_i}^{x_{i+1}} \phi_i\phi_{i+1}\,dx \\
&= c_i\int_{x_i}^{x_{i+1}}\left(1 - \frac{n}{L}|x - x_i|\right)\left(1 - \frac{n}{L}|x - x_{i+1}|\right)dx \\
&= c_i\int_{x_i}^{x_{i+1}}\left(1 - \frac{n}{L}(x - x_i)\right)\left(1 + \frac{n}{L}(x - x_{i+1})\right)dx \\
&= -c_i\frac{n^2}{L^2}\int_{x_i}^{x_{i+1}}(x - x_i)(x - x_{i+1})\,dx \\
&= -c_i\frac{n^2}{L^2}\left[\frac{x^3}{3} - (x_i + x_{i+1})\frac{x^2}{2} + x_i x_{i+1}x\right]_{x_i}^{x_{i+1}} \\
&= c_i\frac{L}{6n}
\end{aligned}$$

dove nell'ultimo passaggio basta sostituire ricordando che $x_k = k\frac{L}{n}$.
Calcoliamo ora le entrate di H:

$$\begin{aligned}
h_{ii} &= \int_0^L k(x)(\phi_i'(x))^2\,dx = \int_{x_{i-1}}^{x_{i+1}} k(x)(\phi_i'(x))^2\,dx = k_{i-1}\int_{x_{i-1}}^{x_i}(\phi_i')^2\,dx + k_i\int_{x_i}^{x_{i+1}}(\phi_i')^2\,dx \\
&= k_{i-1}\int_{x_{i-1}}^{x_i}\left(\frac{n}{L}\right)^2 dx + k_i\int_{x_i}^{x_{i+1}}\left(-\frac{n}{L}\right)^2 dx \\
&= \frac{n}{L}(k_{i-1}+k_i) \\
h_{i,i+1} &= \int_0^L k(x)\phi_i'(x)\phi_{i+1}'(x)\,dx = k_i\int_{x_i}^{x_{i+1}}\phi_i'\phi_{i+1}'\,dx \\
&= k_i\int_{x_i}^{x_{i+1}}\left(-\frac{n}{L}\right)\left(\frac{n}{L}\right)dx \\
&= \frac{n}{L}k_i
\end{aligned}$$

Infine notiamo che

$$b_i = f_0(t)k(0)\phi_i(0) = \begin{cases} f_0(t)k_0 & \text{se } i=0 \\ 0 & \text{altrimenti} \end{cases}$$

In conclusione le matrici risultano allora

$$P = \frac{L}{6n}\begin{pmatrix}
2c_0 & c_0 & 0 & 0 & \dots & \dots & 0 \\
c_0 & 2(c_0+c_1) & c_1 & 0 & \dots & \dots & 0 \\
0 & c_1 & 2(c_1+c_2) & c_2 & & \dots & 0 \\
0 & 0 & c_2 & \ddots & \ddots & & 0 \\
\vdots & \vdots & & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \vdots & & \ddots & 2(c_{n-2}+c_{n-1}) & c_{n-1} \\
0 & 0 & 0 & 0 & \dots & c_{n-1} & 2c_{n-1}
\end{pmatrix}$$

$$H = \frac{n}{L}\begin{pmatrix}
k_0 & -k_0 & 0 & 0 & \dots & \dots & 0 \\
-k_0 & k_0+k_1 & -k_1 & 0 & \dots & \dots & 0 \\
0 & -k_1 & k_1+k_2 & -k_2 & & \dots & 0 \\
0 & 0 & -k_2 & \ddots & \ddots & & 0 \\
\vdots & \vdots & & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \vdots & & \ddots & k_{n-2}+k_{n-1} & -k_{n-1} \\
0 & 0 & 0 & 0 & \dots & -k_{n-1} & k_{n-1}
\end{pmatrix}$$

$$b = \begin{pmatrix} f_0(t)k_0 & 0 & \dots & 0 \end{pmatrix}$$

La tecnica del *mass-lumping*

Vediamo anche come calcolare la matrice P utilizzando la tecnica del *mass-lumping*. Per svolgere i conti abbiamo utilizzato la formula dei trapezi, ma si osservi che, come preannunciato, se proviamo a calcolarla sommando le entrate della matrice P per righe otteniamo lo stesso risultato.

Calcoliamo le entrate, tenendo a mente che $\phi_i(x_j) = \delta_{ij}$:

$$\begin{aligned}
p_{ii} &= c_{i-1}\int_{x_{i-1}}^{x_i} \phi_i^2(x)\,dx + c_i\int_{x_i}^{x_{i+1}} \phi_i^2(x)\,dx \\
&\approx c_{i-1}\frac{L}{2n}\left[\phi_i^2(x_{i-1}) + \phi_i^2(x_i)\right] + c_i\frac{L}{2n}\left[\phi_i^2(x_i) + \phi_i^2(x_{i+1})\right] \\
&= \frac{L}{2n}(c_{i-1} + c_i) \\
p_{i,i+1} &= c_i\int_{x_i}^{x_{i+1}} \phi_i\phi_{i+1}\,dx \\
&\approx c_i\frac{L}{2n}\left[\phi_i(x_i)\phi_{i+1}(x_i) + \phi_i(x_{i+1})\phi_{i+1}(x_{i+1})\right] \\
&= 0
\end{aligned}$$

Con questa approssimazione la matrice P diventa quindi

$$P = \frac{L}{2n}\begin{pmatrix}
c_0 & 0 & 0 & 0 & \dots & \dots & 0 \\
0 & c_0 + c_1 & 0 & 0 & \dots & \dots & 0 \\
0 & 0 & c_1 + c_2 & 0 & & \dots & 0 \\
0 & 0 & 0 & \ddots & \ddots & & 0 \\
\vdots & \vdots & & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \vdots & & \ddots & c_{n-2} + c_{n-1} & 0 \\
0 & 0 & 0 & 0 & \dots & 0 & c_{n-1}
\end{pmatrix}$$

Si può osservare che (una volta terminata la discretizzazione temporale e calcolate le temperature) se l'ordine del modello non è troppo piccolo ($n > 5$) le temperature vengono calcolate in modo pressochè identico indipendentemente che si usi o meno l'approssimazione *mass-lumping*.

6.1.3 La discretizzazione temporale

Portiamo le equazioni

$$P\dot{T} + HT = b$$

nella forma canonica dei modelli *state-space* continui, moltiplicando tutta l'equazione per P^{-1}; otteniamo cioè

$$\dot{x}(t) = A_c x(t) + B_c u(t)$$

$$\text{con } A_c = -P^{-1}H, \quad B_c = k_0 P^{-1}, \quad x(t) = T(t), \quad u(t) = (f_0(t), 0, \dots, 0)$$

Manca ora da trascrivere la seconda equazione del modello in modo da avere il sistema in forma canonica:

$$\begin{cases} \dot{x}(t) = A_c x(t) + B_c u(t) \\ y(t) = C_c x(t) \end{cases}$$

Come abbiamo detto il problema di stima dei parametri consiste nel, dato il calore immesso nel primo nodo (gli ingressi) e misurata l'evoluzione della temperatura lungo la sbarretta, sfruttare tali dati per ricostruire i parametri fisici

della sbarretta. Come già accennato, tipicamente le temperature $x(t)$ non sono misurabili, ma possiamo misurare direttamente solo alcune di esse o addirittura solo delle grandezze dipendenti in modo lineare dalle temperature. Tali grandezze (o sottoinsieme di temperature) saranno le $y(t)$ e il loro legame lineare con le temperature sarà espresso dalla matrice C_c. Il caso più semplice è chiaramente quello in cui tutte le temperature sono misurabili e quindi banalmente $C_c = \mathbb{I}$.

Adesso applichiamo Eulero implicito come descritto nella sezione (5.2.2). La discretizzazione porta a un sistema di equazioni alle differenze di questo tipo:

$$\begin{cases} x_{k+1} = A_f x_k + B_f u_k \\ y_k = C_f x_k \end{cases} \qquad \text{con} \qquad \begin{aligned} A_f &= (I + \Delta t P^{-1} H)^{-1} \\ B_f &= k_0 \left(\frac{1}{\Delta t} P + H \right)^{-1} \\ C_f &= C_c \end{aligned}$$

6.2 L'applicazione del metodo subspace

Dopo aver discretizzato temporalmente il modello ODE, possiamo generare i dati sperimentali. Quello che passeremo in input al codice che stima le matrici del sistema é dunque:

- Una matrice u in cui ogni riga è associato un nodo della discretizzazione spaziale: in ciascuna riga ci sarà il calore immesso nel nodo nel corso del tempo. Osserviamo peró che in realtà nel nostro caso immettiamo calore solo nel primo nodo. Dunque solo la prima riga della matrice avrà valori non nulli. Scegliamo quindi di non passare l'intera matrice all'algoritmo che applica gli algoritmi *subspace*, ma di passare solo la prima riga. Questo fa sì che, se per generare i dati sperimentali abbiamo usato il modello discreto
$$x[:, k+1] = A_f x[:, k] + B_f u[:, k]$$
invece il metodo *subspace* tenterá di stimare le matrici del modello
$$x[:, k+1] = A_f x[:, k] + Bf[:, 0] u[0, k]$$

- Una matrice y strutturata similmente a u: ciascuna riga conterrà i valori che assume la temperatura nell'intervallo di tempo considerato, la quale è stata generata con il modello appena costruito.

- L'ordine del sistema che vogliamo stimare. Possiamo infatti scegliere che il modello stimato abbia un ordine leggermente più grande dell'ordine del modello con cui abbiamo generato i dati. Gli eventuali benefici dell'amento o calo dell'ordine del sistema stimato saranno analizzati nel dettaglio più avanti

- L'indice i (nel codice chiamato `nbr`) usato per definire, nella teoria, le matrici $U_{0|2i-1}, Y_{0|2i-1}$.

Il codice restituirà delle matrici A_s, B_s, C_s, D_s: tali matrici, come descritto nel Capitolo **??**, non sono necessariamente le matrici nella base corretta, ma necessitano di un cambiamento di base in modo da ricavare una stima effettiva di A_f, B_f.

6.3 Il calcolo dei parametri note le matrici stimate

Una volta che abbiamo una stima delle matrici A_f, B_f possiamo ricavare una stima delle matrici A_c, B_c, sapendo che

$$A_c = \frac{1}{\Delta t}(\mathbb{I} - A_f^{-1}) \qquad B_c = \frac{1}{\Delta t} A_f^{-1} B_f$$

Le entrate delle matrici A_c, B_c sono una funzione dei parametri di conduttività e capacità termica, quindi, nota l'espressione simbolica di tali matrici, è possibile ricavare delle formule per il calcolo dei parametri. Per ricavare l'espressione simbolica di A_c, B_c, basta ricordare che esse sono state calcolate con le formule $A_c = -P^{-1}H, B_c = k_0 P^{-1}$ e utilizzare le espressioni simboliche di P e H. Il problema dell'applicare queste formule sta nel fatto che, se consideriamo l'espressione di P calcolata in modo esatto (e dunque non quella approssimata con il *mass-lumping*), la matrice risulta tridiagonale e quindi difficile da invertire. Invece se noi scegliamo di calcolare A_c, B_c usando l'espressione simbolica della P approssimata, il calcolo risulta banale e si ottiene

$$A_c = -\frac{2}{h^2}\begin{pmatrix} \frac{k_0}{c_0} & -\frac{k_0}{c_0} & 0 & 0 & \dots & \dots & 0 \\ -\frac{k_0}{c_0+c_1} & \frac{k_0+k_1}{c_0+c_1} & -\frac{k_1}{c_0+c_1} & 0 & \dots & \dots & 0 \\ 0 & -\frac{k_1}{c_1+c_2} & \frac{k_1+k_2}{c_1+c_2} & -\frac{k_2}{c_1+c_2} & & \dots & 0 \\ 0 & 0 & -\frac{k_2}{c_2+c_3} & \ddots & \ddots & & 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \frac{k_{n-2}+k_{n-1}}{c_{n-2}+c_{n-1}} & -\frac{k_{n-1}}{c_{n-2}+c_{n-1}} \\ 0 & 0 & 0 & 0 & \dots & -\frac{k_{n-1}}{c_{n-1}} & \frac{k_{n-1}}{c_{n-1}} \end{pmatrix}$$

$$B_c = \frac{2}{h}k_0\begin{pmatrix} \frac{1}{c_0} & 0 & 0 & 0 & \dots & \dots & 0 \\ 0 & \frac{1}{c_0+c_1} & 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & \frac{1}{c_1+c_2} & 0 & & \dots & 0 \\ 0 & 0 & 0 & \ddots & \ddots & & 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \frac{1}{c_{n-2}+c_{n-1}} & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & \frac{1}{c_{n-1}} \end{pmatrix}$$

dove $h = L/n$.
Ricordiamo ancora che, avendo immesso nel codice non l'intera matrice degli input ma solo la riga relativa al primo nodo, il codice restituisce una stima solo della prima colonna della matrice B_c.

6.3.1 Espressione analitica dei coefficienti

Dalle espressioni simboliche delle matrici possiamo ricavare ora delle formule ricorsive per la stima dei coefficienti k_i, c_i:

$$\begin{cases} c_0 \\ c_1 = c_0\left(\frac{a_{0,1}}{a_{1,0}} - 1\right) \\ c_i = (c_{i-1} + c_{i-2})\left(\frac{a_{i-1,i}}{a_{i,i-1}}\right) - c_{i-1} \end{cases} \qquad \begin{cases} k_0 = \frac{h}{2} c_0 b_{0,0} \\ k_i = \frac{a_{i,i+1}}{a_{i,i-1}} k_{i-1} \end{cases}$$

Ci si accorge tuttavia di un problema: abbiamo detto che l'utilizzo o meno dell'approssimazione *mass-lumping* nel corso del calcolo delle temperature non produce effetti sulle stesse. Tuttavia, se si utilizzano come dati per la stima delle matrici le temperature calcolate utilizzando la matrice P esatta e poi si prova a calcolare i coefficienti k_i, c_i utilizzando queste formule, i risultati sono completamente sbagliati: questo perché, benché le dinamiche generali del sistema nei due casi siano molto simili, le matrici del sistema sono abbastanza diverse. Perció, ai fini della stima dei coefficienti, anche per la generazione dei dati dovremo usare necessariamente la matrice P condensata.

6.4 La stima dello stato iniziale

Il codice 'subid' non calcola lo stato iniziale. L'idea é la seguente: lo stato iniziale x_0 é la prima colonna della matrice

$$X_0 = (x_0, x_1, \cdots, x_{j-1})$$

L'algoritmo utilizzato per la stima delle matrici si basa sul teorema di proiezione ortogonale il quale dice che la matrice definita come

$$Z_i := Y_f / \begin{pmatrix} W_p \\ U_f \end{pmatrix}$$

coincide con

$$Z_i = \Gamma_i X_i + H_i U_f$$

Inoltre formule analoghe sono descritte nell'algoritmo stesso per descrivere Z_{i+1}. In generale possiamo quindi ricavare che

$$X_{i+k} = \Gamma^+_{i-k}\left(Y_{i+k|2i-1} / \begin{pmatrix} Y_{0|i+k-1} \\ U_{0|2i-1} \end{pmatrix} - H_{i-k} U_{i+k|2i-1}\right)$$

e quindi

$$X_0 = \Gamma^+_{2i}\left(Y_{0|2i-1} / U_{0|2i-1} - H_{2i} U_{0|2i-1}\right)$$

Per un calcolo più ottimizzato si rimanda all'Appendice B.

6.5 I risultati: il problema dell'instabilitá

Applicando l'algoritmo *subspace* per stimare il modello dell'equazione del calore dai dati, nella stragrande maggioranza delle situazioni emergono autovalori errati, spesso instabili.

Nella sezione 3.5 abbiamo riportato due metodi per calcolare A in modo da forzarne la stabilitá. Nella prossima sezione riporteremo i risultati della sperimentazione del primo di questi due metodi e vedremo che purtroppo non sono buoni: benché riesca a rendere la matrice A stabile, esso porta alla stima di un modello che non descrive bene le relazioni di ingresso-uscita. Nelle sezioni successive vedremo quindi altre tecniche proposte per stabilizzare la matrice A.

6.5.1 Risultati sul primo metodo di stabilizzazione

Il primo metodo di stabilizzazione descritto alla sezione 3.5 non sembra purtroppo efficace. Come già accennato, tale metodo é efficace nel forzare gli autovalori di A ad essere inferiori ad 1, tuttavia sono molto differenti dagli autovalori della matrice A vera. Questo si traduce ovviamente in un errore nella stima degli output piuttosto vistoso anche per ordini piuttosto bassi. Addirittura, nel caso in cui il tradizionale algoritmo *subspace* dia già una stima di A stabile, la forzatura della stablitá peggiora le stime degli autovalori, come nel caso di ordine 3. Alcuni risultati sono riportati in tabella 6.1 e in figura 6.1.

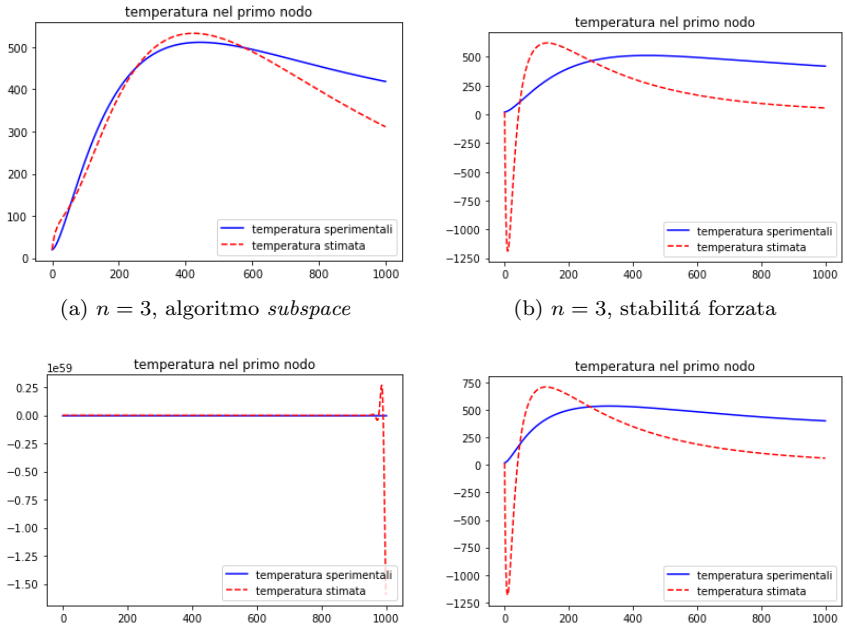
aut. veri	aut. stimati	aut. stabili forzati
0.9615	0.9424	0.0560-0.2538i
0.9651	0.9863	0.0560+0.2538i
0.9745	0.9957	0.2762
0.9864	1.0000	0.8976
0.9961	1.1104-0.2595i	0.8997-0.0009i
1.	1.1104+0.2595i	0.8997+0.0009i

aut. veri	aut. stimati	aut. stabili forzati
0.9936	0.9768	0.1111
0.9968	0.9971	0.8944
1.	0.9999	0.9004

Tabella 6.1: Le due tabelle riportano i risultati per il modello di ordine 3 e 6. La prima colonna riporta gli autovalori della matrice A; la seconda colonna riporta gli autovalori del modello stimato senza forzare la stabilitá; infine la terza colonna riporta gli autovalori del modello stimato forzando la stabilitá.

6.5.2 L'intervento sull'ordine del modello

Dal fallimento del metodo precedente per garantire la stabilitá si é reso quindi necessario elaborare una strategia alternativa per indurre A ad essere stabile. Per farlo si é andati ad analizzare piú a fondo il fenomeno.
Inizialmente si era attribuito il fenomeno dell'instabilitá al forte malcondizionamento delle matrici di Hankel e del sistema lineare usato per il calcolo di A e C. Ci si é quindi concentrati sulla diagnosi e diminuzione di tali indici di malcondizionamento. I tentativi per la riduzione del malcondizionamento (introduzione di rumore bianco nei dati in input, calcolo degli angoli principali tra lo spazio delle righe della matrice di Hankel degli input e lo spazio delle righe della matrice di Hankel degli stati stimati dall'algoritmo *subspace*, regolarizzazione col metodo di Tychonov) hanno portato a concludere che anche in presenza di indici di condizionamento non cosí elevati, le situazioni di instabilitá tendevano a non sparire.



(a) $n = 3$, algoritmo *subspace*

(b) $n = 3$, stabilitá forzata

(c) $n = 6$, algoritmo *subspace*

(d) $n = 6$, stabilitá forzata

Figura 6.1: I grafici rappresentano le uscite dei modelli stimati (in rosso) confrontate con quelle dei modelli veri (in blu) per gli ordini $n = 3, 6$ nel caso in cui non si forzi la stabilitá o la si imponga col primo metodo descritto nella sezione 3.5

Si é dunque andati ad approfondire lo studio di questo fenomeno, come illustrato nel Capitolo 4. I conti analitici e in un secondo momento le analisi sperimentali hanno messo in luce come, per modelli di ordine 1, con la scelta particolare che abbiamo fatto dell'ingresso, il singolo autovalore venisse stimato alla perfezione, confermando che il malcondizionamento delle matrici considerate non influiva sulla buona stima dello stesso; abbiamo visto infatti attraverso calcolo simbolico che, poiché l'ingresso era quasi nullo, tutte e tre le matrici non solo erano malcondizionate ma addirittura singolari.
Dall'analisi dei sistemi diagonalizzabili di ordine 2 é emerso che nel caso i due autovalori della matrice A coincidano, l'algoritmo *subspace* riesce a identificare solo il primo dei due autovalori mentre il secondo autovalore risulta errato. Ulteriori sperimentazioni hanno messo in luce che un fenomeno analogo si verifica nel caso in cui gli autovalori della matrice A siano molto vicini tra loro. Questo perché, come abbiamo osservato, nel caso in cui gli autovalori coincidano, stiamo tentando di stimare un modello che non costituisce una realizzazione minima della relazione ingresso-uscita e pertanto i metodi *subspace* non sono in grado di individuarla nel modo corretto.
Nel caso dell'equazione del calore unidimensionale, se si provano a calcolare gli autovalori del modello che genera i dati ci si accorge che ci si trova proprio nella situazione di autovalori molto ravvicinati tra loro e vicini ad 1, il che porta a pensare che ci si trovi in una situazione analoga a quella appena descritta.

La soluzione sembra quella quindi di intervenire sull'ordine di stima del modello, richiedendo all'algoritmo di stimare un modello di ordine inferiore a quello

effettivo.

6.5.3 La stima dell'ordine ridotto: i criteri di parsimonia

In generale nelle situazioni pratiche l'ordine del sistema da stimare é ignoto, quindi uno dei problemi dell'identificazione di un sistema é la determinazione dell'ordine. Da un punto di vista teorico i metodi *subspace* e il Teorema di identificazione riportato nel Capitolo 3 offrono anche una tecinica per la stima dell'ordine attraverso la determinazione del rango della matrice $W_1 O_i W_2$. Tuttavia da un punto di vista numerico la situazione é meno chiara e l'algoritmo richiede l'ordine da tastiera. Occorrono quindi dei metodi che permettano di dare una stima dell' ordine.
Se siamo interessati ad approssimare dei dati con un modello, un ordine di stima troppo basso potrebbe comportare una descrizione non sufficientemente precisa delle relazioni che intercorrono tra i dati; d'altra parte un ordine troppo elevato potrebbe portare a una descrizione fin troppo dettagliata che, oltre a ripercuotersi sulle dimensioni del problema, i costi computazionali e i tempi di calcolo, finirebbe per descrivere componenti dei dati che saremmo invece interessati a rimuovere, ad esempio un eventuale rumore. Inoltre anche nel caso in cui l'ordine sia noto si puó scegliere di provare ridurre oppure aumentare l'ordine di stima, come accade nel nostro caso.
Il principio generale per la scelta dell'ordine di stima è il seguente: poichè in generale un aumento dell'ordine comporta una diminuzione dell'errore di stima dei dati, conviene incrementare tale ordine fino a quando la decrescita dell'errore risulta significativa; nel momento in cui l'aumento dell'ordine non comporta un grosso guadagno in termini di approssimazione dei dati, è bene arrestarsi. Tale principio viene detto *principio di parsimonia*.

Per quantificare quindi quanto un ordine sia conveniente in questo senso vengono formulati i *criteri di parsimonia*. Ne riportiamo qui alcuni; denotiamo con:

- n_t il numero delle osservazioni dei dati
- n il numero di parametri del modello, quindi l'ordine
- V la *funzione di perdita*: essa rappresenta essenzialmente l'errore di stima pesato sul numero di osservazioni:

$$V = \frac{1}{n_t} \|y - y_{stimato}\|^2$$

I criteri che presentiamo sono:

- AIC (Akaike's Information Criterion):

$$\log\left[V\left(1 + \frac{2n}{n_t}\right)\right]$$

- MDL (Minimum Description Lenght):

$$V\left(1 + \frac{n \log(n_t)}{n_t}\right)$$

I due criteri il più delle volte si equivalgono: l'ordine di stima che minimizza l'uno, minimizza anche l'altro.

Usando i criteri di parsimonia, emerge che per assicurarci delle buone stime degli output, dobbiamo richiedere all'algoritmo di stimare un modello di ordine pari circa a 6, indipendentemente dall'ordine del modello da cui abbiamo generato i dati. Questo significa che il modello stimato risulta molto distante dal modello vero e quindi l'estrazione dei parametri dalle matrici rischia di dare dei risultati piuttosto abbozzati.

6.5.4 Il rimedio all'instabilitá: la riduzione per troncamento

La scelta di chiedere all'algoritmo direttamente un modello di ordine ridotto non sembra del tutto efficace ai fini di ottenere una buona approssimazione del modello originario, in quanto ci troviamo a dover richiedere che l'ordine di stima sia notevolmente più basso di quello vero.
Il secondo tentativo quindi é stato chiedere all'algoritmo di stimare il modello dandogli in input l'ordine corretto ed effettuare una riduzione per troncamento del modello stimato. Questo seconda soluzione ha permesso di selezionare gli autovalori della matrice A del sistema stimato scartando gli autovalori instabili, negativi o troppo piccoli. La strategia é stata individuata in due tentativi: il primo metodo descritto é un troncamento *a posteriori* del modello stimato; il secondo metodo é un troncamento delle sole matrici A e C stimate, usate poi per calcolare B e D.

Primo metodo

Il primo tentativo di riduzione si é articolato nei seguenti passaggi

- Una volta ottenuta una stima delle matrici A, B, C, D, si é calcolata una trasformazione T che portasse in forma diagonale Δ la matrice A:

  $$\Delta = \text{diag}(\lambda_{1,s}, \dots, \lambda_{r,s}, \lambda_{1,i}, \dots, \lambda_{n-r,i}) = TAT^{-1}$$

  dove per semplicitá supponiamo che gli autovalori siano ordinati in modulo in senso crescente e quindi gli autovalori instabili siano gli ultimi:
  $|\lambda_{1,s}| \leq \dots \leq |\lambda_{r,s}| \leq 1 < |\lambda_{1,i}| \leq \dots \leq |\lambda_{n-r,i}|$

- usando la trasformazione T si é ottenuto un sistema algebricamente equivalemte all'originale:

  $$\Delta = TAT^{-1} = \underset{r}{\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}}\begin{matrix} r \\ n-r \end{matrix} \qquad B_\Delta = TB = \underset{m}{\begin{pmatrix} B_1 \\ B_2 \end{pmatrix}}\begin{matrix} r \\ n-r \end{matrix}$$
  $$C_\Delta = CT^{-1} = \underset{r}{\begin{pmatrix} C_1 \end{pmatrix}}\begin{matrix} l \end{matrix} \qquad D_\Delta = \underset{m}{D}\begin{matrix} l \end{matrix}$$

- prendiamo come sistema stimato ridotto il sistema di matrici A_{11}, B_1, C_1, D

Nel procedimento appena illustrato abbiamo assunto di voler eliminare solo gli autovalori instabili, ma ovviamente con un troncamento maggiore del sistema

possiamo scegliere di rimuovere tutte le righe e le colonne associate agli autovalori che non riteniamo validi. Sperimentalmente si verifica che tra gli autovalori rimasti dall'elisione di quelli instabili e negativi, quelli che risultano più accurati sono quelli più grandi. Si puó quindi decidere di ridurre progressivamente l'ordine rimuovendo via via gli autovalori minori fino a ottenere un modello che approssima bene le relazioni di ingresso-uscita. Si osservi che ció é possibile anche senza conoscere gli autovalori del sistema originale da stimare, in quanto la scelta degli autovalori ""buoni" si fa solo per confronto tra l'output del sistema vero e l'output del sistema stimato troncato.

All'atto pratico questo metodo non si é poi verificato cosí efficace: la matrice A_{11} risulta ovviamente stabilizzata, eppure il modello spesso non approssima bene le relazioni ingresso-uscita. Probabilmente ció é dovuto alle matrici B e D, le quali sono state calcolate a partire da una matrice A instabile e troncate a posteriori e sono quindi inesatte. Infatti, dopo aver effettuato il calcolo di A e C, l'algoritmo *subspace* utilizza queste stime per ricalcolare la matrice Γ_i, che però, se A é instabile, ha delle entrate piuttosto elevate che si allontanano da qulli che dovrebbero essere i veri valori contenuti in Γ_i. L'idea dunque é di stabilizzare A *prima* che avvenga il calcolo di B e D.

Secondo metodo

Siccome vogliamo stabilizzare la matrice A prima che venga utilizzata per la stima di B e D, bisogna andare ad aggiungere direttamente le istruzioni all'interno dell'algoritmo *subspace*. In pratica ció che si fá é:

- Calcolare A e C utilizzando come ordine di stima l'ordine effettivo del sistema: $n_x := n$
- Analogamente a prima, calcolare una trasformazione T che porti in forma diagonale Δ la matrice A; eseguire poi il cambio di base anche per C

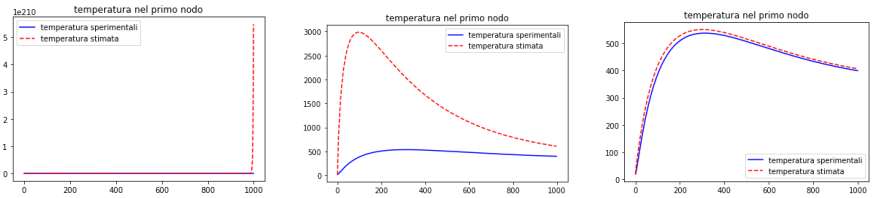
$$\Delta = TAT^{-1} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \qquad C_\Delta = CT^{-1} = \begin{pmatrix} C_1 & C_2 \end{pmatrix}$$

- si prosegue l'algoritmo utilizzando, invece che le matrici A e C e l'ordine $n_x := n$, le matrici A_{11} e C_1 e l'ordine di stima $n_x := r$, dove r é il numero di autovalori conservati della matrice A

Infatti siccome l'algoritmo prevede il ricalcolo della matrice Γ_i una volta effettuate le stime di A e C, non é necessario effettuare nessun cambio di base o troncamento in tutto il resto dell'algoritmo e le matrici B e D ottenute dall'algoritmo *subspace* non necessiteranno del troncamento.

Nonostante la matrice A troncata ottenuta con questo metodo e la matrice A troncata a posteriori del metodo precedente siano identiche, questa strategia si rivela essere molto più efficace rispetto alla precedente e permette di ottenere risultati piuttosto buoni (in termini di riproduzione della relazione ingresso-uscita) anche per ordini abbastanza grandi.

Ad esempio, per dati sperimentali generati da un modello di ordine $n =$



(a) (b) (c)

Figura 6.2: I dati sperimentali sono stati generati per tempi $t_k = k \cdot 0.01, k = 0, \ldots, 1000$ da un modello di ordine 17. Il primo grafico confronta le temperature reali e le temperature generate dal modello stimato dall'algoritmo *subspace* usando $i = 10$. Il secondo grafico mostra il confronto con le temperature generate dal modello stimato troncato a posteriori, di cui si sono conservati 13 autovalori stabili. Infine l'ultimo grafico mostra le temperature ottenute dal modello ricavato troncando le matrici A e C all'ordine 13 prima del calcolo di B e D.

$9, 10, \ldots, 20$ per tempi $t_k = k \cdot 0.01, k = 0, \ldots, 1000$ con $i = 10$, possiamo vedere dalla tabella come scegliere l'ordine r con cui troncare il sistema stimato perché le stime delle temperature risultino buone.

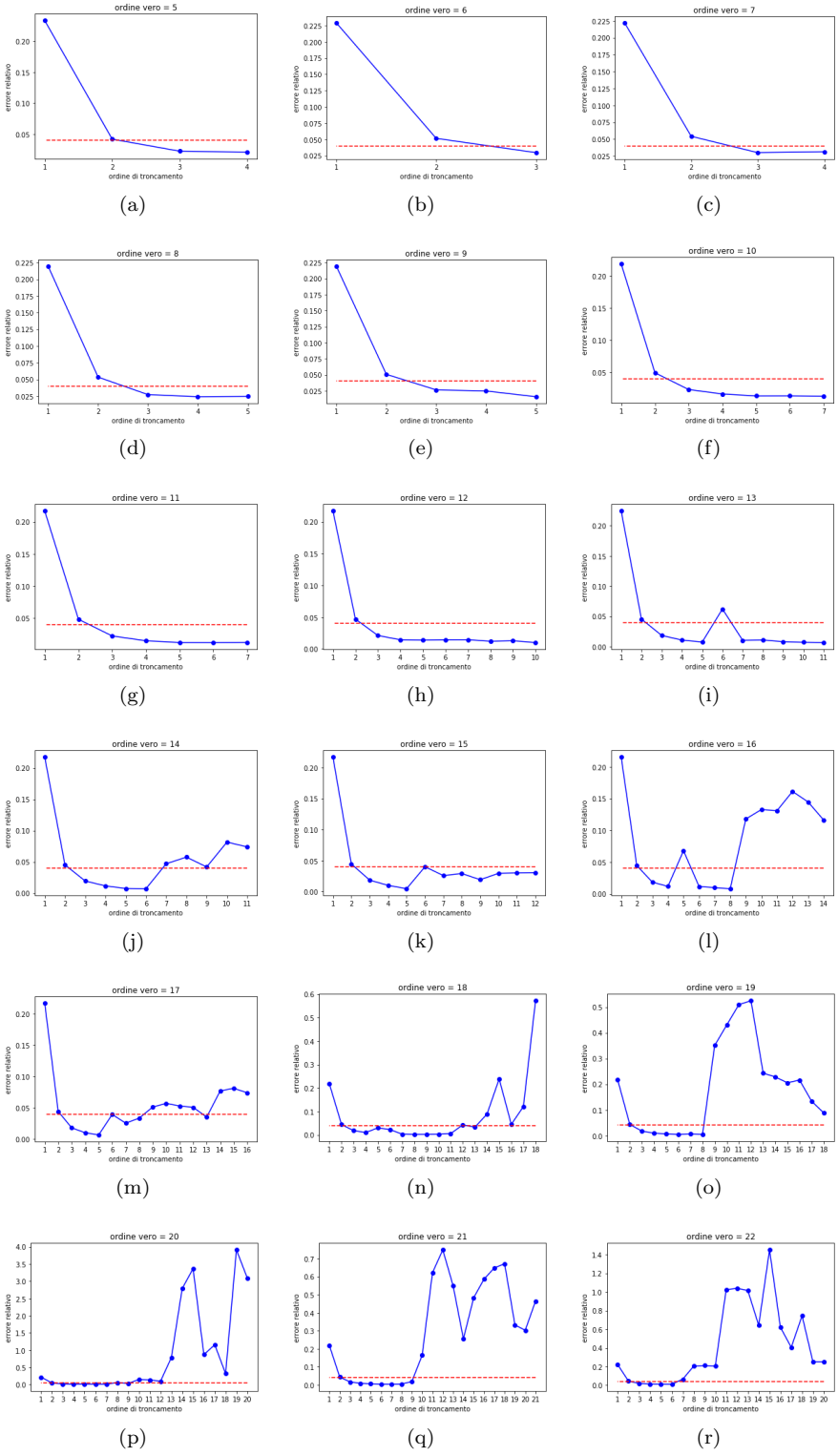
n	r	err	n	r	err	n	r	err
5	4	2.105%	11	7	1.237%	17	5	0.662%
6	3	2.968%	12	10	1.002%	18	11	0.542%
7	4	3.127%	13	11	0.674%	19	8	0.471%
8	5	2.459%	14	6	0.711%	20	6	0.961%
9	5	1.568%	15	5	0.479%	21	7	0.380%
10	7	1.287%	16	8	0.801%	22	6	1.129%

Tabella 6.2: In tabella per ogni ordine n é riportato l'ordine di troncamento r per cui si ha il minimo errore relativo *err*

n	r	err	n	r	err	n	r	err
5	4	2.105%	11	7	1.237%	17	13	3.504%
6	3	2.968%	12	10	1.002%	18	13	3.281%
7	4	3.127%	13	11	0.674%	19	8	0.471%
8	5	2.459%	14	9	4.180%	20	9	3.048%
9	5	1.568%	15	12	3.072%	21	9	1.862%
10	7	1.287%	16	8	0.801%	22	6	1.129%

Tabella 6.3: In tabella per ogni ordine n é riportato l'ordine di troncamento r massimo per cui si ha un errore relativo $err < 4\%$.

A differenza di quanto accadeva per i criteri di parsimonia, che eravamo costretti a richiedere un modello molto più ridotto di quello vero, con questo metodo, almeno per ordini inferiori a 18, possiamo ottenere delle buone stime degli output scartando solo qualche autovalore. Purtroppo per ordini superiori si vede che per avere buone stime i valori da scartare sono molti di piú.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

(p) (q) (r)

Figura 6.3

Appendice A

Equivalenza tra due diverse formulazioni del filtro di Kalman

Nei capitoli 2 e 3 abbiamo fornito due formulazioni diverse per il filtro di Kalman. Nella seconda formulazione le equazioni ricorsive di stima della matrice di covarianza dell'errore $P_{k|j} := \mathbb{E}[(x_k - x_{k|j})(x_k - x_{k|j})^T]$ sono rimpiazzate dalle equazioni per la stima della matrice di covarianza dello stato stimato $\Pi_{k|j} := \mathbb{E}[x_{k|j}x_{k|j}^T]$. La scelta di mantenere entrambe le formulazioni all'interno del testo è dovuta in parte per facilitare la trattazione dei concetti, in parte per mantenere le formulazioni utilizzate nelle fonti di riferimento (precisamente [2] e [1]) e facilitare il confronto col testo originario.
Qui di seguito si intende mostrare come passare da una formulazione all'altra. Ci occuperemo solo delle equazioni di predizione.

Consideriamo, per il momento, il generico sistema *state-space* puramente stocastico:

$$\begin{cases} x_{k+1} = Ax_k + w_k \\ y_k = C_k x_k + v_k \end{cases}$$

$$\mathbb{E}\left[\begin{pmatrix} w_t \\ v_t \end{pmatrix}\begin{pmatrix} w_s \\ v_s \end{pmatrix}^T\right] = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}\delta_{ts}$$

e denotiamo

$$\begin{aligned} &P_{k|k-1} := \mathbb{E}[(x_k - x_{k|k-1})(x_k - x_{k|k-1})^T] \\ &\Pi_{k|k-1} := \mathbb{E}[x_{k|k-1}x_{k|k-1}^T] \\ &\Sigma := \mathbb{E}[x_k x_k^T] = \mathbb{E}[x_{k+1}x_{k+1}^T] \\ &G := \mathbb{E}[x_{k+1}y_k^T] \\ &\Lambda_0 := \mathbb{E}[y_k y_k^T] \end{aligned}$$

Allora le due formulazioni del filtro sono

Prima formulazione: Detta $K_k = (AP_{k|k-1}C^T + S)(CP_{k|k-1}C^T + R)^{-1}$, le equazioni di predizione del filtro di Kalman sono

$$\begin{aligned} x_{k+1|k} &= Ax_{k|k-1} + Bu_k + K_k[y_k - Cx_{k|k-1})] \\ P_{k+1|k} &= AP_{k|k-1}A^T + Q - K_k(CP_{k|k-1}C^T + R)K_k^T \end{aligned}$$

Seconda formulazione: Detta $K_k = (G - A\Pi_{k|k-1}C^T)(\Lambda_0 - C\Pi_{k|k-1}C^T)^{-1}$, le equazioni di predizione del filtro di Kalman sono

$$\begin{aligned} x_{k+1|k} &= Ax_{k|k-1} + Bu_k + K_k[y_k - Cx_{k|k-1}] \\ \Pi_{k+1|k} &= A\Pi_{k|k-1}A^T + K_k(G - A\Pi_{k|k-1}C^T)^T \end{aligned}$$

Osserviamo che

- Ricordando la decomposizione ortogonale (2.3) vediamo che

$$\begin{aligned} \Sigma =& \mathbb{E}[x_k x_k^T] \\ =& \mathbb{E}\Big[\Big(x_{k|k-1} + (x_k - x_{k|k-1})\Big)\Big(x_{k|k-1} + (x_k - x_{k|k-1})\Big)^T\Big] \\ =& \mathbb{E}[x_{k|k-1}x_{k|k-1}^T] + \mathbb{E}[(x_k - x_{k|k-1})(x_k - x_{k|k-1})^T] + \\ &+ \mathbb{E}[x_{k|k-1}(x_k - x_{k|k-1})^T] + \mathbb{E}[(x_k - x_{k|k-1})x_{k|k-1}^T] \\ =& \mathbb{E}[x_{k|k-1}x_{k|k-1}^T] + \mathbb{E}[(x_k - x_{k|k-1})(x_k - x_{k|k-1})^T] \\ =& \Pi_{k|k-1} + P_{k|k-1} \end{aligned}$$

- Inoltre

$$\begin{aligned} \Lambda_0 &= \mathbb{E}[y_k y_k^T] \\ &= \mathbb{E}[(C_k x_k + v_k)(C_k x_k + v_k)^T] \\ &= C\mathbb{E}[x_k x_k^T]C^T + \mathbb{E}[v_k v_k^T] \\ &= C\Sigma C^T + R \end{aligned} \qquad \begin{aligned} G &= \mathbb{E}[x_{k+1}y_k^T] \\ &= \mathbb{E}[(Ax_k + w_k)(C_k x_k + v_k)^T] \\ &= A\mathbb{E}[x_k x_k^T]C^T + \mathbb{E}[w_k v_k^T] \\ &= A\Sigma C^T + S \end{aligned}$$

$$\begin{aligned} \Sigma &= \mathbb{E}[x_{k+1}x_{k+1}^T] \\ &= \mathbb{E}[(Ax_k + w_k)(Ax_k + w_k)^T] \\ &= A\mathbb{E}[x_k x_k^T]A^T + \mathbb{E}[w_k w_k^T] \\ &= A\Sigma A^T + Q \end{aligned}$$

- Quindi è chiaro che le due definizioni di K_k nelle due formulazioni coincidono, poichè

$$\begin{aligned} G - A\Pi_{k|k-1}C^T &= (A\Sigma C^T + S) - A\Pi_{k|k-1}C^T \\ &= A(\Sigma - \Pi_{k|k-1})C^T + S \\ &= AP_{k|k-1}C^T + S \\ &= K_k(CP_{k|k-1}C^T + R) \end{aligned}$$

$$\begin{aligned}\Lambda_0 - C\Pi_{k|k-1}C^T &= (C\Sigma C^T + R) - C\Pi_{k|k-1}C^T \\ &= C(\Sigma - \Pi_{k|k-1})C^T + R \\ &= CP_{k|k-1}C^T + R\end{aligned}$$

quindi sono anche equivalenti le equazioni per la predizione degli stati.

- verifichiamo ora l'equivalenza delle equazioni per la stima di $P_{k|k-1}$ e $\Pi_{k|k-1}$:

$$\begin{aligned}P_{k+1|k} &= \Sigma - \Pi_{k+1|k} \\ &= [A\Sigma A^T + Q] - [A\Pi_{k|k-1}A^T + K_k(G - A\Pi_{k|k-1}C^T)^T] \\ &= A(\Sigma - \Pi_{k|k-1})A^T + Q - K_k(G - A\Pi_{k|k-1}C^T)^T \\ &= A(\Sigma - \Pi_{k|k-1})A^T + Q - K_k(CP_{k|k-1}C^T + R)^T K_k^T \\ &= AP_{k|k-1}A^T + Q - K_k(CP_{k|k-1}C^T + R)^T K_k^T\end{aligned}$$

Da ciascuno di questi filtri per sistemi puramente stocastici possiamo poi ricavare i filtri di Kalman generali come fatto nel Capitolo 2, verificando che anche i filtri per sistemi generali sono equivalenti.

Appendice B

Note tecniche sull'implementazione dei metodi subspace

B.1 Calcolo ottimizzato delle proiezioni ortogonali ed oblique

Abbiamo visto nella sezione(3.2.1) delle formule per il calcolo, date tre matrici $A \in \mathbb{R}^{p\times j}, B \in \mathbb{R}^{q\times j}, C \in \mathbb{R}^{r\times j}$, delle matrici $A/B \in \mathbb{R}^{p\times j}, A/_C B \in \mathbb{R}^{p\times j}$, ossia rispettivamente

$$\begin{aligned} A/B &= A\Pi_B \qquad \text{con } \Pi_B := B^T(BB^T)^+B \\ A/_C B &= (A/B^\perp)(C/B^\perp)^+C \end{aligned}$$

Un modo piú funzionale per il calcolo di queste matrici sfrutta le decomposizioni QR. Supponiamo infatti di aver decomposto le tre matrici A, B, C in modo che abbiano la stessa matrice Q, ossia

$$A = R_A Q^T \qquad B = R_B Q^T \qquad C = R_C Q^T \tag{B.1}$$

allora le formule per il calcolo di A/B e $A/_C B$ diventano

$$\begin{aligned} A/B &= R_A R_B^T (R_B R_B^T)^+ R_B Q^T \\ A/_C B &= [R_A(\mathbb{I} - R_B^T(R_B R_B^T)^+ R_B)][R_C(\mathbb{I} - R_B^T(R_B R_B^T)^+ R_B)]^+ R_C Q^T \end{aligned}$$

Osserviamo infine che le decomposizioni QR (B.1) sono equivalenti alla decomposizione

$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} R_A \\ R_B \\ R_C \end{pmatrix} Q^T$$

B.2 La risoluzione del sistema per il calcolo di A e C

Come si intuisce giá dalla sezone precedente, uno degli strumenti fondamentali degli algoritmi *subspace* é la decomposizione QR. Decomponendo in tal modo la matrice

$$\frac{1}{\sqrt{j}}\begin{pmatrix} U_{0|2i-1} \\ Y_{0|2i-1} \end{pmatrix} = RQ^T$$

si riesce a esprimere tutti i calcoli in termini solo di sottomatrici della matrice R (come giá si poteva intuire dal calcolo ottimizzato delle proiezioni), con notevoli vantaggi in termini di costi computazionali e costi di memoria.
Scriviamo nel dettaglio la decomposizione:

$$\frac{1}{\sqrt{j}}\begin{pmatrix} U_{0|2i-1} \\ Y_{0|2i-1} \end{pmatrix} = \frac{1}{\sqrt{j}}\begin{pmatrix} U_{0|i-1} \\ U_{i|i} \\ U_{i+1|2i-1} \\ Y_{0|i-1} \\ Y_{i|i} \\ Y_{i+1|2i-1} \end{pmatrix} \begin{matrix} mi \\ m \\ m(i-1) \\ li \\ l \\ l(i-1) \end{matrix}$$

$$RQ^T = \begin{matrix} mi \\ m \\ m(i-1) \\ li \\ l \\ l(i-1) \end{matrix} \begin{pmatrix} R_{11} & 0 & 0 & 0 & 0 & 0 \\ R_{21} & R_{22} & 0 & 0 & 0 & 0 \\ R_{31} & R_{32} & R_{33} & 0 & 0 & 0 \\ R_{41} & R_{42} & R_{43} & R_{44} & 0 & 0 \\ R_{51} & R_{52} & R_{53} & R_{54} & R_{55} & 0 \\ R_{61} & R_{62} & R_{63} & R_{64} & R_{65} & R_{66} \end{pmatrix} \begin{pmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \\ Q_4^T \\ Q_5^T \\ Q_6^T \end{pmatrix}$$

Allora con questa decomposizione, denoteremo cosí le sottomatrici di R

$$R_{[1:4][1:3]} = \begin{pmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \\ R_{41} & R_{42} & R_{43} \end{pmatrix} \qquad R_{[1,4][1,3]} = \begin{pmatrix} R_{11} & 0 \\ R_{41} & R_{43} \end{pmatrix}$$

Il calcolo di Z_i e Z_{i+1}

Ricordiamo che

$$Z_i := Y_f / \left(\begin{smallmatrix} U_{0|2i-1} \\ Y_{0|i-1} \end{smallmatrix}\right) \qquad Z_{i+1} := Y_{f^-} / \left(\begin{smallmatrix} U_{0|2i-1} \\ Y_{0|i} \end{smallmatrix}\right)$$

si tratta dunque di sfruttare la decomposizione QR fatta sopra per riscrivere le proiezioni.
Osserviamo intanto che

$$\begin{aligned} Y_f &= R_{[5:6][1:6]}Q^T \\ \left(\begin{smallmatrix} U_{0|2i-1} \\ Y_{0|i-1} \end{smallmatrix}\right) &= R_{[1:4][1:6]}Q^T = \left[R_{[1:4][1:4]}, \mathbb{O}_{(2m+l)i \times li)}\right]Q^T \\ Y_{f^-} &= R_{[6:6][1:6]}Q^T \\ \left(\begin{smallmatrix} U_{0|2i-1} \\ Y_{0|i} \end{smallmatrix}\right) &= R_{[1:5][1:6]}Q^T = \left[R_{[1:5][1:5]}, \mathbb{O}_{(2m+l)i+l \times l(i-1)}\right]Q^T \end{aligned}$$

Ricordando che in generale vale $A/B = AB^T(BB^T)^+B$ che $QQ^T = \mathbb{I}$ allora

$$\begin{aligned}
Z_i &= R_{[5:6][1:6]}R^T_{[1:4][1:6]}(R_{[1:4][1:6]}R^T_{[1:4][1:6]})^+R_{[1:4][1:6]}Q^T \\
&= R_{[5:6][1:6]}\left(\begin{smallmatrix} R^T_{[1:4][1:4]} \\ 0 \end{smallmatrix}\right)\Big(\big[R_{[1:4][1:4]},0\big]\left(\begin{smallmatrix} R^T_{[1:4][1:4]} \\ 0 \end{smallmatrix}\right)\Big)^+\big[R_{[1:4][1:4]},0\big]Q^T \\
&= R_{[5:6][1:6]}\left(\begin{smallmatrix} R^T_{[1:4][1:4]} \\ 0 \end{smallmatrix}\right)(R_{[1:4][1:4]}R^T_{[1:4][1:4]})^{-1}R_{[1:4][1:4]}Q^T_{[1:4]} \\
&= R_{[5:6][1:6]}\left(\begin{smallmatrix} R^T_{[1:4][1:4]} \\ 0 \end{smallmatrix}\right)R^{-T}_{[1:4][1:4]}R^{-1}_{[1:4][1:4]}R_{[1:4][1:4]}Q^T_{[1:4]} \\
&= R_{[5:6][1:6]}\left(\begin{smallmatrix} \mathbb{I} \\ 0 \end{smallmatrix}\right)Q^T_{[1:4]} \\
&= R_{[5:6][1:4]}Q^T_{[1:4]}
\end{aligned}$$

e analogamente

$$Z_{i+1} = R_{[6:6][1:5]}Q^T_{[1:5]}$$

Il calcolo di O_i

Ricordando che

$$O_i = Y_f/_{U_f}W_p$$

e ricordano la formula per il calcolo delle proiezioni oblique $A/_BC = (A/B^\perp)(C/B^\perp)^+C$, vediamo che ci basta calcolare le proiezioni $Y_f/U_f^\perp$ e $W_p/U_f^\perp$. Analogamente a come abbiamo fatto sopra, notiamo che

$$\begin{aligned}
U_f &= R_{[2:3][1:6]}Q^T = \big[R_{[2:3][1:3]}, \mathbb{O}_{mi\times 2li}\big]Q^T \\
W_p &= R_{[1,4][1:6]}Q^T = \big[R_{[1,4][1:4]}, \mathbb{O}_{(m+l)i\times li}\big]Q^T
\end{aligned}$$

Calcoliamo per esteso solo $Y_f/U_f^\perp$ (analogamente calcoleremo $C/B^\perp$):

$$\begin{aligned}
Y_f/U_f &= R_{[5:6][1:6]}R^T_{[2:3][1:6]}(R_{[2:3][1:6]}R^T_{[2:3][1:6]})^+R_{[2:3][1:6]}Q^T \\
&= R_{[5:6][1:6]}\left(\begin{smallmatrix} R^T_{[2:3][1:3]} \\ 0 \end{smallmatrix}\right)\Big(\big[R_{[2:3][1:3]},0\big]\left(\begin{smallmatrix} R^T_{[2:3][1:3]} \\ 0 \end{smallmatrix}\right)\Big)^+\big[R_{[2:3][1:3]},0\big]Q^T \\
&= R_{[5:6][1:3]}R^T_{[2:3][1:3]}(R_{[2:3][1:3]}R^T_{[2:3][1:3]})^+\big[R_{[2:3][1:3]},0\big]Q^T
\end{aligned}$$

quindi

$$\begin{aligned}
Y_f/U_f^\perp &= Y_f - Y_f/U_f \\
&= R_{[5:6][1:6]}Q^T - R_{[5:6][1:3]}R^T_{[2:3][1:3]}(R_{[2:3][1:3]}R^T_{[2:3][1:3]})^+\big[R_{[2:3][1:3]},0\big]Q^T \\
&= \Big[\big[R_{[5:6][1:3]},R_{[5:6][4:6]}\big] - R_{[5:6][1:3]}R^T_{[2:3][1:3]}(R_{[2:3][1:3]}R^T_{[2:3][1:3]})^+\big[R_{[2:3][1:3]},0\big]\Big]Q^T \\
&= \big[R_{[5:6][1:3]} - R_{[5:6][1:3]}R^T_{[2:3][1:3]}(R_{[2:3][1:3]}R^T_{[2:3][1:3]})^+R_{[2:3][1:3]},R_{[5:6][4:6]}\big]Q^T
\end{aligned}$$

o piú brevemente

$$Y_f/U_f^\perp = R_{fp}Q^T \qquad \text{con } R_{fp} := \big[R_{[5:6][1:3]} - \text{argmin}_X\|XR_{[2:3][1:3]} - R_{[5:6][1:3]}\|R_{[2:3][1:3]}, R_{[5:6][4:6]}\big]$$

Analogamente

$$W_p/U_f^\perp = R_{pp}Q^T \qquad \text{con } R_{pp} := \big[R_{[1,4][1:3]} - \text{argmin}_X\|XR_{[2:3][1:3]} - R_{[1,4][1:3]}\|R_{[2:3][1:3]}, R_{[1,4][4:6]}\big]$$

In conclusione, abbiamo che

$$\begin{aligned}
O_i &= (R_{fp}Q^T)(R_{pp}Q^T)^+R_{[1,4][1:6]}Q^T \\
&= R_{fp}Q^TQ^{-T}R^+_{pp}R_{[1,4][1:6]}Q^T \\
&= R_{fp}R^+_{pp}R_{[1,4][1:6]}Q^T
\end{aligned}$$

Il calcolo di A e C

Possiamo adesso riscrivere il sistema

$$\begin{pmatrix} \Gamma_{i-1}^{+} Z_{i+1} \\ Y_{i|i} \end{pmatrix} = \left(\begin{array}{c|c} A & \\ C & \mathcal{K} \end{array}\right) \begin{pmatrix} \Gamma_i^{+} Z_i \\ U_f \end{pmatrix}$$

Osserviamo che

$$\begin{aligned} Z_{i+1} &= R_{[6:6][1:5]} Q_{[1:5]}^T \\ Y_{i|i} &= R_{[5:5][1:6]} Q^T = \left[R_{[5:5][1:5]}, \mathbb{O}_{l \times l(i-1)}\right] Q^T = R_{[5:5][1:5]} Q_{[1:5]}^T \\ Z_i &= R_{[5:6][1:4]} Q_{[1:4]}^T = \left[R_{[5:6][1:4]}, \mathbb{O}_{li \times l}\right] Q_{[1:5]}^T \\ U_f &= R_{[2:3][1:6]} Q^T = \left[R_{[2:3][1:5]}, \mathbb{O}_{mi \times l(i-1)}\right] Q^T = R_{[2:3][1:5]} Q_{[1:5]}^T \end{aligned}$$

Quindi sostituendo e semplificando $Q_{[1:5]}^T$ otteniamo il sistema

$$\begin{pmatrix} \Gamma_{i-1}^{+} R_{[6:6][1:5]} \\ R_{[5:5][1:5]} \end{pmatrix} = \left(\begin{array}{c|c} A & \\ C & \mathcal{K} \end{array}\right) \begin{pmatrix} \Gamma_i^{+} [R_{[5:6][1:4]}, \mathbb{O}_{li \times l}] \\ R_{[2:3][1:5]} \end{pmatrix}$$

Il calcolo dello stato iniziale

Nella sezione 6.4 abbiamo proposto la seguente formula per il calcolo dello stato iniziale

$$X_0 = \Gamma_{2i}^{+} \Big(Y_{0|2i-1} / U_{0|2i-1} - H_{2i} U_{0|2i-1} \Big) \tag{B.2}$$

intendiamo anche in questo caso riformulare il calcolo delle proiezioni. Vediamo che

$$\begin{aligned} U_{0|2i-1} &= R_{[1:3][1:6]} Q^T = \left[R_{[1:3][1:3]}, \mathbb{O}_{2mi \times 2li}\right] Q^T = R_{[1:3][1:3]} Q_{[1:3]}^T \\ Y_{0|2i-1} &= R_{[4:6][1:6]} Q^T \end{aligned}$$

Allora

$$\begin{aligned} Y_{0|2i-1} / U_{0|2i-1} &= R_{[4:6][1:6]} R_{[1:3][1:6]}^T (R_{[1:3][1:6]} R_{[1:3][1:6]}^T)^{+} R_{[1:3][1:6]} Q^T \\ &= R_{[4:6][1:6]} \begin{pmatrix} R_{[1:3][1:3]}^T \\ 0 \end{pmatrix} (R_{[1:3][1:3]} R_{[1:3][1:3]}^T)^{-1} R_{[1:3][1:3]} Q_{[1:3]}^T \\ &= R_{[4:6][1:6]} \begin{pmatrix} R_{[1:3][1:3]}^T \\ 0 \end{pmatrix} R_{[1:3][1:3]}^{-T} R_{[1:3][1:3]}^{-1} R_{[1:3][1:3]} Q_{[1:3]}^T \\ &= R_{[4:6][1:3]} Q_{[1:3]}^T \end{aligned}$$

Quindi la (B.2) diventa

$$X_0 = \Gamma_{2i}^{+} \Big(R_{[4:6][1:3]} - H_{2i} R_{[1:3][1:3]} \Big) Q_{[1:3]}^T$$

L'algoritmo completo

Riassumendo l'algoritmo in pratica diventa:

Algoritmo

1) Viene calcolata la decomposizione RQ^T della matrice

$$\frac{1}{\sqrt{j}}\begin{pmatrix} U_{0|2i-1} \\ Y_{0|2i-1} \end{pmatrix}$$

2) Si calcolano le matrici

$$O_i := Y_f/_{U_f} W_p = R_{fp} R_{pp}^{+} R_{[1,4][1:6]}$$

3) Eseguiamo la decomposizione a valori singolari pesata

$$O_i \Pi_{U_f^{\perp}} = USV^T$$

e poniamo $U_1 := U[:, 0:n]$ e $S_1 := S[0:n, 0:n]$ dove n é l'ordine di stima dato in input all'algoritmo

4) si pone $\Gamma_i = U_1 S_1^{1/2}$ e si imposta il sistema per il calcolo di A e C nell'incognita $\boldsymbol{X}$

$$\begin{pmatrix} \Gamma_{i-1}^{+} R_{[6:6][1:5]} \\ R_{[5:5][1:5]} \end{pmatrix} = \boldsymbol{X} \begin{pmatrix} \Gamma_i^{+}[R_{[5:6][1:4]}, \mathbb{O}_{li \times l}] \\ R_{[2:3][1:5]} \end{pmatrix}$$

Si porrá poi $A := \boldsymbol{X}[0:n, 0:n]$ e $C := \boldsymbol{X}[n:n+l, 0:n]$

5) Viene ricalcolata Γ_i usando le matrici A e C appena stimate

6) Con questa nuova stima della matrice Γ_i e con le soluzioni di A e C si calcola la matrice

$$\mathcal{P} = \begin{pmatrix} \Gamma_{i-1}^{+} R_{[6:6][1:5]} \\ R_{[5:5][1:5]} \end{pmatrix} - \begin{pmatrix} A \\ C \end{pmatrix} \begin{pmatrix} \Gamma_i^{+}[R_{[5:6][1:4]}, \mathbb{O}_{li \times l}] \\ R_{[2:3][1:5]} \end{pmatrix}$$

7) Ricordando che $\mathcal{Q} = U_f = \left[R_{[2:3][1:3]}, \mathbb{O}_{mi \times 2li}\right]$, si imposta e risolve il sistema

$$vec\left(\begin{smallmatrix} D \\ B \end{smallmatrix}\right) = \left[\sum_{k=1}^{i} \mathcal{Q}_k^T \otimes \mathcal{N}_k\right]^{+} vec(\mathcal{P})$$

ricavando cosí B e D

Appendice C

L'algoritmo di Householder

L'algoritmo di Householder per il calcolo della decomposizione QR consiste nel definire delle matrici Q_i ortogonali per costruzione tali che $Q_n^T \cdot \cdots \cdot Q_3^T Q_2^T Q_1^T M = R^T$ e porre poi $Q = Q_1 Q_2 \dots Q_n$; ciascuna matrice Q_i è costruita in modo da lasciare invariate le prime $i-1$ colonne della matrice $(Q_{i-1}^T \cdot \cdots \cdot Q_2^T Q_1^T M)$ e annullare le entrate subdiagonali della i-esima colonna. Per fare ciò la generica matrice Q_i^T ha forma

$$Q_i^T = \begin{pmatrix} \mathbb{I}_{i-1} & 0 \\ 0 & P_i \end{pmatrix}$$

dove P_i é un'opportuna matrice di simmetria. Per semplicitá di notazione, denotiamo $M_i := Q_i^T \cdot \cdots \cdot Q_2^T Q_1^T M$ il risultato dell'aver giá posto in forma triangolare superiore le prime i colone della matrice M. Supponiamo di essere arrivati a calcolare la matrice M_{k-1}, vogliamo annullare quindi le entrate subdiagonali della k-esima colonna attraverso la premoltiplicazione tramite la matrice Q_k. Siamo dunque nella seguente situazione.

$$\underbrace{\begin{pmatrix} \mathbb{I}_{k-1} & \mathbb{O} \\ \mathbb{O} & P_k \end{pmatrix}}_{Q_k^T} \underbrace{\begin{pmatrix} R_{[1:k-1,1:k-1]} & * & * & * \\ & v_1 & * & * \\ \mathbb{O} & v_2 & * & * \\ & \dots & * & * \\ & v_n & * & * \end{pmatrix}}_{M_{k-1}} = \underbrace{\begin{pmatrix} R_{[1:k-1,1:k-1]} & * & * & * \\ & \|v\| & * & * \\ \mathbb{O} & 0 & * & * \\ & \dots & * & * \\ & 0 & * & * \end{pmatrix}}_{M_k}$$

La matrice P_k é dunque la simmetria che permette di portare il vettore $v := (v_1, \dots, v_n)$ sul sottospazio $\mathrm{span}(e_1)$.

Il calcolo della matrice P_k

Descriviamo piú nel dettaglio la matrice P_k e l'opportuno iperpiano $\mathcal{I}$ che fá da asse di simmetria:

- Consideriamo il vettore w che porta v su $\|v\|e_1$, cioè:

  $$w := \|v\|e_1 - v$$

  si osservi che l'iperpiano $\mathcal{I}$ é l'iperpiano passante per l'origine ortogonale a w, che avrá dunque equazione

  $$\mathcal{I} : w^T x = 0$$

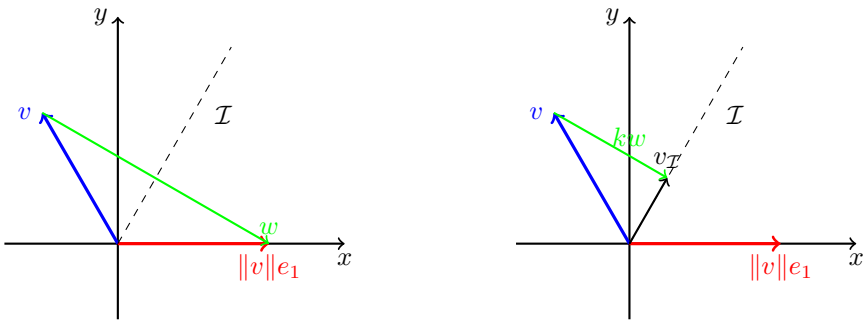


Figura C.1: Rappresentazione della simmetria P_k

- consideriamo preliminarmente la proiezione $v_{\mathcal{I}}$ di v su $\mathcal{I}$. Essa si ottiene sommando a v un multiplo di w, cioè

$$v_{\mathcal{I}} = v + kw \qquad \text{con } k \text{ t.c. } v_{\mathcal{I}} \in \mathcal{I}$$

  quindi per calcolare k basta porre

$$w^T(v + kw) = 0 \quad \Rightarrow \quad k = \frac{w^T v}{\|w\|^2}$$

- é chiaro allora che valga

$$\begin{aligned} \|v\|e_1 &= v - 2kw \\ &= v - 2\frac{w^T v}{\|w\|^2}w = v - 2\frac{ww^T}{\|w\|^2}v = \left(\mathbb{I} - 2\frac{ww^T}{\|w\|^2}\right)v \end{aligned}$$

  quindi

$$P_k := \mathbb{I} - 2\frac{ww^T}{\|w\|^2}$$

Bibliografia

[1] P. Van Overschee, B. De Moor, *Subspace identification for linear systems*, Kluwer Academic Publishers, Boston/London/Dordrecht, 1996

[2] T. Katayama, *Subspace methods for system identification*, Springer, 2005

[3] E. Fornasini, G.Marchesini, *Appunti di Teoria dei Sistemi*, Edizioni libreria Progetto Padova, 2003

[4] A. C. Antoulas, *Approximation of Large-Scale Dynamical System*, SIAM, 2005

[5] J.M.Maciejowski, *Guaranteed Stability with Subspace Methods*, Systems and Control Letters, Volume 26, Issue 2, 22 September 1995, Pages 153-156

[6] S. L. Lacy, D. S. Bernstein, *Subspace Identification With Guaranteed Stability Using Constrained Optimization*, IEEE Transactions on Automatic Control, Volume 48, Issue 7, July 2003

[7] Paul Bekker, *The Positive Semidefiniteness of Partitioned Matrices*, Linear Algebra and its Applications, Volume 111, December 1988, Pages 261-278

[8] A. Quarteroni, *Modellistica Numerica per Problemi Differenziali*, Springer, 2008

[9] M. Putti, *Dispense del corso di Metodi Numerici per le Equazioni Differenziali*, 2016

[10] G. H. Golub, C. F. Van Loan, *Matrix Computation*, The John Hopkins University Press, 1996

[11] J. Demmel, *Applied Numerical Linear Algebra*, SIAM, 1997