

UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA



*Corso di Laurea Triennale in
Ingegneria Elettronica*

*Identificazione di sistemi non lineari mediante il metodo
particle-Bernstein polynomials*

*Nonlinear system identification with particle-Bernstein
polynomials method*

Relatore:
DOTT. TURCHETTI CLAUDIO

Laureando:
CAPPONI MICHELE

Correlatore:
DOTT.SSA FALASCHETTI LAURA

ANNO ACCADEMICO 2019-2020

Indice

1	Introduzione	5
1.1	Obiettivo	5
1.2	Struttura della tesi	5
2	Identificazione di sistemi dinamici non lineari	7
2.1	Generalità	7
2.2	Tecniche di identificazione di sistemi dinamici non lineari	8
2.2.1	Serie di Volterra	8
2.2.2	Modelli strutturati a blocchi	9
2.2.3	Reti neurali artificiali	10
2.2.4	Modelli NARMAX	11
2.3	Un approccio alternativo all'identificazione dei sistemi dinamici non lineari	12
2.4	Rappresentazione delle curve come combinazione di polinomi	13
2.5	I polinomi di Bernstein	14
2.6	Polinomi Particle-Bernstein	15
3	Sperimentazione	17
3.1	Dataset scelto per la sperimentazione	18
3.2	Misure disponibili nel dataset	18
3.3	Descrizione dei segnali di ingresso applicati	19
3.3.1	Eccitazione di tipo "Sine-sweep with a linear, negative rate"	19
3.3.2	Eccitazione di tipo "Multisine with a full frequency grid"	19
3.3.3	Eccitazione di tipo "Multisine with a random frequency grid"	20
3.4	Elaborazione del Dataset	20
3.5	Risultati della sperimentazione	23
3.5.1	Hardware utilizzato durante la sperimentazione	23
4	Conclusioni e possibili sviluppi futuri	27
4.1	Conclusioni	27
4.2	Possibili sviluppi futuri	28
	Bibliografia	31

Elenco delle figure

31

Capitolo 1

Introduzione

1.1 Obiettivo

Il tema affrontato nella tesi riguarda l'identificazione di sistemi dinamici non lineari. Negli ultimi anni sono state proposte diverse tecniche per risolvere questo problema, tuttavia la maggior parte dei metodi noti richiede una complessa indagine al fine di individuare il miglior modello per la rappresentazione del sistema. La determinazione del modello che più si adatta ai dati è uno degli step fondamentali e cruciali nelle tecniche di identificazione più diffuse e in molti casi risulta essere un'operazione molto costosa in termini computazionali.

Il seguente lavoro si pone come obiettivo l'identificazione di un sistema dinamico reale, non lineare, sperimentando un'approccio alternativo che non richiede l'individuazione di un modello e permette di trasformare il problema dinamico in un problema di regressione di una funzione non lineare, statica. Il grande vantaggio che si ottiene in seguito alla trasformazione è che, per approssimare funzioni non lineari statiche esistono molti algoritmi di machine learning consolidati. In questo caso, è stata utilizzata una recente tecnica di rappresentazione polinomiale particolarmente efficiente, basata sui polinomi di Bernstein che, come verrà mostrato in seguito, consente di ricavare i coefficienti del polinomio limitando il costo computazionale e riducendo i tempi di addestramento della rete.

1.2 Struttura della tesi

La tesi è articolata in quattro capitoli:

- nel secondo capitolo verranno brevemente descritti i metodi più diffusi per l'identificazione dei sistemi dinamici non lineari e saranno discussi i principali vantaggi e svantaggi che questi offrono. In seguito, sarà dettagliatamente analizzata la tecnica sperimentata in questo lavoro e, in questo contesto, verranno introdotti i polinomi di Bernstein, mostrando come que-

sti possano essere utilizzati per approssimare generiche funzioni statiche, non lineari;

- nel terzo capitolo verrà descritto il dataset scelto per la sperimentazione e saranno illustrate le elaborazioni svolte al fine di renderlo compatibile con la tecnica di identificazione seguita;
- nel quarto ed ultimo capitolo verranno discussi i risultati ottenuti e saranno proposti alcuni possibili sviluppi futuri.

Capitolo 2

Identificazione di sistemi dinamici non lineari

2.1 Generalità

L'identificazione di sistemi può essere definita come l'insieme delle tecniche finalizzate alla determinazione di un modello matematico di un sistema ottenuto mediante una serie di misurazioni degli ingressi e delle uscite e, in alcuni casi, attraverso una parziale conoscenza del sistema stesso [3].

L'identificazione può essere applicata a qualsiasi sistema in cui gli input e gli output possono essere misurati come ad esempio processi industriali, sistemi di controllo, processi economici, sistemici elettronici e molti altri.

La maggior parte dei metodi di identificazione impongono degli step comuni da seguire per la creazione di un modello adeguato del sistema:

- **raccolta dei dati:** selezione, pre-elaborazione ed elaborazione di un set di dati appropriato;
- **postulati del modello:** assunzioni iniziali che vengono fatte sulla struttura del modello basate sulla conoscenza del fenomeno e su un'analisi globale dei dati precedentemente raccolti. Le assunzioni sono solo di tipo strutturale ed includono sempre una serie di parametri incogniti da determinare successivamente;
- **identificazione dei parametri:** applicazione di algoritmi di identificazione al fine di determinare i parametri incogniti da cui il modello dipende.
- **convalida del modello:** processo di conferma del modello concettuale e di dimostrazione di un'adeguata corrispondenza tra i risultati computazionali del modello ed i dati effettivi.

2.2 Tecniche di identificazione di sistemi dinamici non lineari

Storicamente, l'identificazione di sistemi non lineari si è sviluppata ponendo l'attenzione su classi specifiche di sistemi per cui sono stati adottati diversi modelli di base, particolarmente noti sono [3] :

- modelli basati su serie di Volterra;
- modelli strutturati a blocchi;
- reti neurali artificiali;
- modelli NARMAX;

2.2.1 Serie di Volterra

Le serie di Volterra generalizzano i concetti di risposta impulsiva e di convoluzione per sistemi dinamici, non lineari, tempo invarianti e consentono di espandere in serie un operatore non lineare [2].

Considerando un sistema di input-output dinamico non lineare, rappresentato da un generico operatore G , con ingresso $u(t)$, ed uscita $y(t)$,

$$y(t) = G(u(t)) \quad (2.1)$$

è possibile espandere in serie di Volterra l'operatore G . La serie di Volterra dell'operatore G è definita come:

$$y(t) = h_0 + \sum_{n=1}^N H_n u(t) \quad (2.2)$$

dove

$$H_n(u(t)) = \int_a^b \dots \int_a^b h_n(\tau_1, \dots, \tau_n) \prod_{j=1}^n u(t - \tau_j) d\tau_j \quad (2.3)$$

con $a, b \in \{-\infty, +\infty\}$ ed $N \in \{0, 1, 2, \dots\}$.

$h_n(\tau_1, \dots, \tau_n)$ è detto kernel di Volterra di ordine n e rappresenta la risposta all'impulso di ordine n .

Le serie di Volterra permettono, a differenza di altre classi di espansione, di ricostruire anche l'effetto memoria dei sistemi non lineari e per questo motivo possono essere considerati, in linea generale, molto efficaci nella rappresentazione di diverse classi di sistemi non lineari. Tuttavia, la determinazione dei kernel multidimensionali è un'operazione molto costosa in termini computazionali e ciò rappresenta una criticità. Per superare le difficoltà di calcolo dei coefficienti, in molte applicazioni specifiche, la serie può essere troncata mantenendo solo il termine lineare e

quello quadratico. Queste approssimazioni non sono sempre accettabili se si vuole ottenere una descrizione completa di sistemi che presentino delle non linearità significative di ordine superiore. Questo metodo è stato successivamente esteso includendo negli sviluppi anche kernel di ordine superiore al secondo [3], tuttavia la stima dei coefficienti di ordine più alto richiede una grande quantità di dati. Nel caso discreto la serie di Volterra di un uscita $y(k)$, $k \in \mathbb{N}$ si può scrivere:

$$\begin{aligned}
 y(k) = & h_0 + \sum_{m_1=1}^M h_1(m_1)u(k - m_1) + \sum_{m_1=1}^M \sum_{m_2=1}^M h_2(m_1, m_2)u(k - m_1)u(k - m_2) \\
 & + \sum_{m_1=1}^M \sum_{m_2=1}^M \sum_{m_3=1}^M h_3(m_1, m_2, m_3)u(k - m_1)u(k - m_2)u(k - m_3) + \dots
 \end{aligned}
 \tag{2.4}$$

dove $u(k)$, $y(k)$; $k \in \mathbb{N}$ sono rispettivamente gli input e gli output misurati e $h_i(m_1, \dots, m_i)$ costituisce il kernel di Volterra di ordine i .

2.2.2 Modelli strutturati a blocchi

Un modello che molto spesso viene utilizzato per la rappresentazione dei sistemi dinamici non lineari è il modello a blocchi strutturati, in cui il sistema viene descritto da una sequenza particolare di blocchi in cascata. Ogni blocco della catena isola una particolare caratteristica del sistema in modo da semplificare il più possibile l'identificazione. Tipicamente, i blocchi che vengono maggiormente utilizzati sono blocchi statici non lineari e blocchi dinamici lineari. La disposizione dei blocchi è ciò che, in molti casi, distingue un modello dall'altro. Schemi a blocchi particolarmente noti ed utilizzati sono:

- sistemi Hammerstein (Fig. 2.1);
- sistemi Wiener (Fig. 2.2);
- sistemi Hammerstein-Wiener (Fig. 2.3);
- sistemi Wiener-Hammerstein (Fig. 2.4);

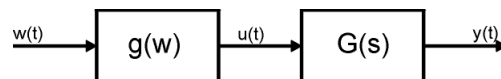


Figura 2.1: Sistemi Hammerstein: la funzione $g(w)$ costituisce un blocco non lineare statico e precede la $G(s)$, che rappresenta una rete lineare dinamica nel dominio di Laplace.

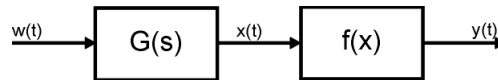


Figura 2.2: Sistemi Wiener: la funzione $G(s)$ costituisce un blocco lineare dinamico nel dominio di Laplace e precede la funzione $f(x)$ che rappresenta le non linearità statiche del sistema

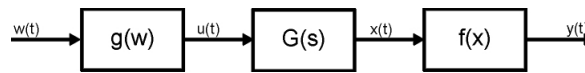


Figura 2.3: Sistemi Hammerstein-Wiener: la funzione $G(s)$ è una funzione lineare dinamica, preceduta e seguita da due blocchi non lineari statici $g(w)$ ed $f(x)$.

Modelli di questo tipo rappresentano con buona approssimazione molti sistemi fisici coinvolti nel campo dell'ingegneria e presentano dei vantaggi nella descrizione di problemi legati ad esempio all'automazione, in cui è possibile separare gli effetti non lineari legati a sensori, attuatori ed altri componenti.

2.2.3 Reti neurali artificiali

Una rete neurale è un modello artificiale che cerca di imitare, seppur molto parzialmente, la struttura computazionale di una rete neurale biologica. La struttura è costituita da un certo numero di semplici unità di elaborazione collegate tra loro (Fig: 2.5).

Le reti neurali sono composte da almeno tre strati:

- un layer di input, ovvero un'insieme di nodi d'accesso costituito dagli ingressi del sistema;
- una successione di hidden layer, dove avviene l'elaborazione degli input;
- un layer di output, in cui sono accessibili i risultati prodotti della rete.

Gli ingressi transitano attraverso la rete e vengono elaborati e combinati nei vari nodi fornendo una o più uscite. Durante la fase di addestramento i nodi intermedi e le loro connessioni si modificano in maniera adattiva al fine di approssimare la relazione di ingresso-uscita desiderata. Nell'apprendimento supervisionato, la rete opera sulla differenza tra l'output effettivo e l'output desiderato cercando di minimizzare ad ogni passo l'errore di previsione.

Esistono diversi tipi di reti neurali ed alcune architetture, come ad esempio le reti ricorrenti, riescono a rappresentare anche sistemi dinamici [3]. Tuttavia, in molti casi, il grande numero di nodi tende a mascherare la struttura del modello e rende difficile l'individuazione dei suoi parametri, infatti, i modelli prodotti dalle reti neurali, anche se molto efficienti, non sempre sono individuabili ed esprimibili in forma analitica e ciò obbliga spesso ad accettare i risultati solo in seguito ad una convalida e non consente di generalizzare il modello a classi di sistemi simili.

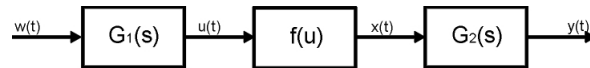


Figura 2.4: Sistemi Wiener-Hammerstein: la funzione $f(u)$ è una funzione non lineare statica, seguita e preceduta da due blocchi dinamici lineari rappresentati rispettivamente da $G_1(s)$ e da $G_2(s)$ nel dominio di Laplace

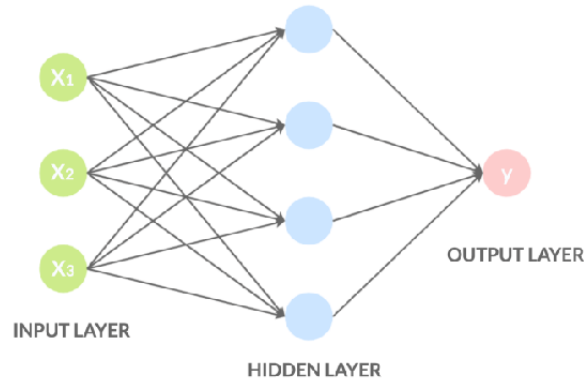


Figura 2.5: Semplice modello strutturale di una rete neurale artificiale

2.2.4 Modelli NARMAX

Uno dei modelli più affermati per la rappresentazione dei sistemi dinamici non lineari è il modello NARMAX (modello autoregressivo nonlineare a media mobile con input esogeni) definito come:

$$y(k) = F[y(k-1), y(k-2), \dots, y(k-n_y), u(k-d), u(k-d-1), \dots, u(k-d-n_u), e(k-1), e(k-2), \dots, e(k-n_e)] + e(k) \quad (2.5)$$

dove le sequenze $y(k)$, $u(k)$, $e(k)$ rappresentano rispettivamente l'output, l'input e il rumore; n_y , n_u , n_e rappresentano rispettivamente i massimi lag temporali nell'output, nell'input e nel rumore; d rappresenta il ritardo temporale (solitamente $d = 1$) e la funzione $F(\cdot)$ è una generica funzione non lineare che caratterizza il sistema.

Nel dominio del tempo discreto il modello NARMAX è uno degli approcci di maggior successo per l'identificazione di sistemi non lineari ed è molto efficace in diverse applicazioni del mondo reale. Tuttavia, una delle maggiori difficoltà nell'identificazione del sistema utilizzando il modello NARMAX è la selezione di un modello parsimonioso nel numero di parametri che riesca a rappresentare adeguatamente la dinamica del sistema [5], inoltre le tecniche più diffuse per la determinazione dei parametri da cui il modello NARMAX dipende (ad esempio, minimizzazione dell'errore di previsione) non sempre riescono a fornire risultati soddisfacenti per scopi di previsione a lungo termine a causa di meccanismi di selezione dei termini che includono regressori ridondanti nel modello.

2.3 Un approccio alternativo all'identificazione dei sistemi dinamici non lineari

Consideriamo un sistema di input-output dinamico rappresentato da un operatore T che agisce su un ingresso u restituendo un uscita y :

$$y = T(u), \quad u \in \mathbb{R}^m, \quad y \in \mathbb{R}^n \quad (2.6)$$

$u = (u(1), \dots, u(n))$ ed $y = (y(1), \dots, y(n))$ sono vettori reali dipendenti dal tempo disceto $t = 1, \dots, n$.

I sistemi di interesse sono sistemi dinamici non lineari quindi considereremo il caso in cui l'operatore T agisce sull'ingresso u e sul tempo t [5].

Supponiamo che l'ingresso u e l'uscita y appartengano a due classi di segnali con rispettive matrici di covarianza R_{uu} ed R_{yy} . Le matrici di covarianza possono essere quindi decomposte come segue:

$$R_{uu} = \psi \Lambda \psi^T, \quad R_{uu} \in \mathbb{R}^{m \times m} \quad (2.7)$$

$$R_{yy} = \phi \Gamma \phi^T, \quad R_{yy} \in \mathbb{R}^{n \times n} \quad (2.8)$$

Con Λ, Γ matrici diagonali composte rispettivamente dagli autovalori di R_{uu} ed R_{yy} e ψ, ϕ matrici le cui colonne sono composte rispettivamente dagli autovettori di R_{uu} ed R_{yy} .

L'insieme dei vettori contenuti nelle colonne di ψ costituisce una base rispetto alla quale possono essere espressi tutti gli ingressi appartenenti alla classe di segnali con matrice di covarianza R_{uu}

$$u = \psi x, \quad x \in \mathbb{R}^m \quad (2.9)$$

equivalentemente, le colonne di ϕ costituiscono una base per i segnali di uscita

$$y = \phi k, \quad k \in \mathbb{R}^n \quad (2.10)$$

E' possibile osservare che i vettori x e k sono indipendenti dal tempo.

Combinando 2.6, 2.9 e 2.10, si ottiene una relazione tra i vettori x e k :

$$k = f(x) \quad (2.11)$$

Essendo x e k indipendenti dal tempo, anche $f(\cdot)$ è una funzione non lineare ed indipendente dal tempo. Stimare la funzione $f(\cdot)$ corrisponde ad identificare l'operatore T , legato ad f dalla relazione:

$$T(\cdot) = \phi f(\psi^T(\cdot)) \quad (2.12)$$

Infine, per ridurre la dimensionalità del problema, è possibile applicare una PCA, trascurando così le componenti dell'ingresso corrispondenti agli autovalori più piccoli. Supponendo di disporre gli autovalori della matrice Ruu in ordine crescente, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, e riordinando i corrispondenti autovettori nella matrice ψ è possibile partizionare ψ :

$$\psi = (\psi_d \psi_\eta) \tag{2.13}$$

in cui la matrice $\psi_d \in \mathbb{R}^{m \times d}$ è una sottomatrice di ψ le cui d colonne sono composte dagli autovettori corrispondenti agli autovalori più significativi.

È possibile riscrivere la 2.9 come segue:

$$u = \psi_d x_d + \psi_\eta x_\eta, x_d \in \mathbb{R}^d \tag{2.14}$$

Trascurando i termini $\psi_\eta x_\eta$, corrispondenti alle componenti meno significative del segnale si ottiene:

$$u \cong \psi_d x_d, \psi_d \in \mathbb{R}^{m \times d}, x_d \in \mathbb{R}^d \tag{2.15}$$

in cui, ψ_d è una sottomatrice di ψ contenente gli autovettori corrispondenti agli autovalori più significativi $\lambda_1, \dots, \lambda_d$, x_d è il sottovettore di x contenente le coordinate. Una volta effettuate le trasformazioni appena descritte, il problema di identificazione del sistema dinamico non lineare si riduce alla regressione di una funzione non lineare statica. La tecnica appena illustrata è del tutto generale e può essere applicata a qualsiasi sistema dinamico non lineare senza la necessità di individuare preventivamente il miglior modello di rappresentazione del fenomeno e ciò rappresenta il punto di forza di questo approccio.

2.4 Rappresentazione delle curve come combinazione di polinomi

Descrivere le curve è un compito molto importante in matematica. Un approccio comune alla descrizione delle curve è l'uso di una rappresentazione esplicita in termini di una o più variabili $y = f(x)$, tuttavia questa rappresentazione presenta diversi problemi e non può essere utilizzata per descrivere qualsiasi tipo di curva. In molti casi è utile avere una descrizione analitica alternativa della funzione espressa in termini di somme di polinomi

$$p(t) = a_0 + a_1 t + a_2 t^2 + \dots + a_{n-1} t^{n-1} + a_n t^n \tag{2.16}$$

I polinomi sono strumenti matematici utili in quanto possono essere memorizzati semplicemente su sistemi informatici e possono essere integrati e assemblati per approssimare qualsiasi funzione.

Molte delle rappresentazioni polinomiali si basano sul noto teorema di approssimazione di Weierstrass secondo il quale ogni funzione continua $f : [0, 1] \rightarrow \mathbb{R}$ può

essere approssimata da un polinomio con precisione arbitraria. [4] Come è noto, il teorema di Taylor fornisce un metodo per approssimare funzioni differenziabili in punti specifici e può produrre talvolta una serie che non converge alla funzione originale, per questo motivo nel seguente paragrafo verrà illustrata una tecnica alternativa e più utile per i nostri scopi basata sui polinomi di Bernstein.

2.5 I polinomi di Bernstein

Consideriamo una generica funzione f dipendente da una sola variabile x . I polinomi di Bernstein di grado m nell'intervallo $[0, 1]$ sono definiti come [4]:

$$b_k^m = \binom{m}{k} x^k (1-x)^{m-k}, \quad k = 0, 1, \dots, m \quad x \in [0, 1] \quad (2.17)$$

essi formano una base completa dello spazio vettoriale dei polinomi di grado minore o uguale ad m .

I polinomi di Bernstein soddisfano una serie di proprietà. Una delle principali è che i coefficienti dei polinomi sono ottenuti valutando semplicemente la funzione f , che si vuole approssimare, in punti appartenenti ad una griglia fissa $\{k/m, k : 0, 1, \dots, m\}$,

$$B_m(x) = \sum_{k=0}^m f(k/m) b_k^m(x) \quad (2.18)$$

il polinomio $B_m(x)$ converge uniformemente ad f per $m \rightarrow \infty$. Questa proprietà è fondamentale per comprendere il vantaggio che si può trarre dall'utilizzo di questa base di polinomi rispetto ad altre basi note, infatti, in molti algoritmi di regressione, la determinazione dei coefficienti è la fase più costosa in termini computazionali.

La relazione 2.18 può essere generalizzata al caso in cui la funzione f dipenda da più variabili

$$B_m(x_1, x_2, \dots, x_d) = \sum_{k_1=0}^m \dots \sum_{k_d=0}^m f(k_1/m, \dots, k_d/m) \times b_k^m(x) \quad (2.19)$$

con $x = (x_1, \dots, x_d)$, $k = (k_1, \dots, k_d)$ e

$$b_k^m = \binom{m}{k_1} \dots \binom{m}{k_d} x_1^{k_1} (1-x_1)^{m-k_1} \dots x_d^{k_d} (1-x_d)^{m-k_d} \quad (2.20)$$

In questa rappresentazione è richiesta la conoscenza della funzione nei punti

$$x = (k_1/m, \dots, k_d/m), k_1 = 0, \dots, m, k_d = 0, \dots, m \quad (2.21)$$

e, come precedentemente indicato, non è necessario un algoritmo per determinare i coefficienti.

Uno degli inconvenienti di questa tecnica è che, per avere una buona approssimazione della funzione f , la griglia di punti da considerare deve essere sufficientemente densa e ciò corrisponde alla necessità di dover alzare il grado m del polinomio, aumentando il costo computazionale della 2.19. Inoltre, l'utilizzo della 2.19 presuppone la conoscenza dei valori della funzione in un set di punti specifici predeterminato e ciò costituisce un limite alla compatibilità di questo metodo con situazioni reali, in cui i campioni raccolti possono non corrispondere con i punti della griglia. Per rimuovere queste limitazioni un approccio alternativo verrà presentato nel seguente paragrafo.

2.6 Polinomi Particle-Bernstein

Nei polinomi di Bernstein le variabili m e k sono variabili a valori interi ed anche i coefficienti binomiali. Queste condizioni possono essere rilassate assumendo k appartenente ai numeri reali e ridefinendo di conseguenza un nuovo set di funzioni composto dai polinomi particle-Bernstein[4],

$$C_{\xi}^m(x) = \alpha_{\xi}^m x^{\xi} (1-x)^{m-\xi} = \alpha_{\xi}^m k_{\xi}^m(x), \quad \xi \in \mathbb{R}, \xi \in [0, m] \quad (2.22)$$

in cui si è posto ξ al posto di k per evidenziare il fatto che sono due variabili diverse ed appartengono ad insieme numerici differenti.

I coefficienti α_{ξ}^m sono scelti in modo tale che

$$\int_0^1 C_{\xi}^m(x) dx = 1 \quad (2.23)$$

È possibile dimostrare che:

$$\alpha_{\xi}^m = \frac{\Gamma(m+2)}{\Gamma(\xi+1)\Gamma(-\xi+m+1)} \quad (2.24)$$

in cui $\Gamma(\cdot)$ corrisponde alla funzione gamma.

Una proprietà importante di questi polinomi è che, come evidenziato dalla 2.23, scegliendo opportunamente i coefficienti α_{ξ}^m , i polinomi sottendono tutti la stessa area ed esibiscono il proprio valore massimo in $x = \xi/m$. Inoltre, scegliendo m sufficientemente alto, le funzioni C_{ξ}^m tendono a concentrarsi intorno al proprio valore massimo e possono essere sfruttate per approssimare una funzione $f(x) \in \mathbb{R}^1$ intorno al punto $x = \xi/m$,

$$f(\xi/m) \cong \int_0^1 f(x) C_{\xi}^m(x) dx \quad (2.25)$$

Essendo $f(x)$ la funzione che si desidera approssimare, non sarà completamente nota a priori ed è quindi necessario trovare un'espressione approssimata per il calcolo dell'integrale nella 2.25. In particolare, si cerca un'espressione che permetta di risolvere l'integrale partendo dalla conoscenza di una serie di campioni della funzione $f(x)$:

$$\int_0^1 f(x) C_\xi^m(x) dx \cong \frac{1}{N} \sum_{j=1}^N f(x^{(j)}) C_\xi^m(x^{(j)}) \quad (2.26)$$

in cui $\{x^{(j)}, j = 1, 2, \dots, N\}$ costituisce un insieme di punti per cui la $f(x)$ è nota ed $\{f(x^{(j)}), j = 1, 2, \dots, N\}$ è composto dall'insieme dei corrispondenti valori assunti da $f(x)$.

La 2.26 fornisce una stima della funzione $f(x)$ nel punto $x = \xi/m$.

La trattazione può essere generalizzata semplicemente per funzioni multivariate,

$$f(x), x \in U \subseteq \mathbb{R}^d \quad (2.27)$$

l'espressione dei polinomi diventa

$$C_\xi^m(x) = \alpha_{\xi_1}^m x_1^{\xi_1} (1 - x_1)^{m - \xi_1} \dots x_d^{\xi_d} (1 - x_d)^{m - \xi_d} = \alpha_\xi^m k_\xi^m(x) \quad (2.28)$$

con $\xi = (\xi_1, \dots, \xi_d)$, $x = (x_1, \dots, x_d)$ e

$$\alpha_\xi^m = \prod_{t=1}^d \frac{\Gamma(m+2)}{\Gamma(\xi_t+1)\Gamma(-\xi_t+m+1)} \quad (2.29)$$

$$k_\xi^m = \prod_{t=1}^d x_t^{\xi_t} (1 - x_t)^{m - \xi_t} \quad (2.30)$$

Per ottenere una stima della funzione $f(x)$ nel punto $x = (\xi_1/m, \dots, \xi_d/m)$ è possibile utilizzare la stessa relazione 2.26 trovata in precedenza.

È importante notare che l'espressione 2.26 permette di ricondurre l'approssimazione della funzione ad una semplice sommatoria e ciò consentendo di limitare il carico computazionale.

Capitolo 3

Sperimentazione

La sperimentazione è stata effettuata considerando un sistema dinamico non lineare, reale, costituito da un aereo F-16 (Fig. 3.1).

I dati sperimentali utilizzati sono stati raccolti dalla Siemens in occasione della Master Class ‘Siemens LMS Ground Vibration Testing Master Class’, tenuta nel settembre 2014 presso la base militare di Saffraanberg (Sint-Truiden, Belgio) e sono pubblicamente disponibili [1].

L’esperimento condotto dalla Siemens nasce dall’esigenza di trovare un modello matematico che permetta di prevedere le oscillazioni di alcune parti della struttura dell’aereo durante il volo [6]. Le turbolenze dell’aria innescano infatti oscillazioni della struttura dell’aereo che si concentrano in particolare alle estremità delle ali, dove tipicamente sono applicati i propulsori secondari.

Nello studio di riferimento, che verrà descritto dettagliatamente nei prossimi paragrafi, sono state simulate ed applicate delle forze meccaniche in modo da riprodurre, a terra, una situazione analoga a quella che si presenta in regime di turbolenza nella fase di volo.

Durante la raccolta dati è stato osservato che, oltre un certo livello di ampiezza della forza, le oscillazioni mostrano delle risonanze entrando in regime non lineare. Il sistema in esame costituisce quindi un sistema dinamico non lineare che, in questo lavoro, si è cercato di identificare.



Figura 3.1: F-16; Test di vibrazione a terra presso base militare di Saffraanenberg.

3.1 Dataset scelto per la sperimentazione

Durante la campagna di test, sono stati montati due carichi fittizi sulle punte delle ali per simulare le proprietà di massa e inerzia dei dispositivi reali che tipicamente equipaggiano un F-16 in volo. (Fig. 3.2). La struttura dell'aeromobile è stata dotata di 145 sensori di accelerazione. Le vibrazioni strutturali sono state stimulate da uno shaker, collegato sotto l'ala destra, per applicare i segnali di ingresso. (Fig. 3.3)

Da un'indagine preliminare la fonte dominante di non linearità nella dinamica della struttura sembrava originarsi dalle interfacce di montaggio dei due carichi fittizi ed è stata quindi al centro dello studio di riferimento.

Figura 3.2: Carico fittizio montato sull'ala destra dell'F16



Figura 3.3: Shaker applicato all'ala destra



3.2 Misure disponibili nel dataset

Le misure sono state acquisite con una frequenza di campionamento di 400 Hz . Sono disponibili due segnali di ingresso distinti:

- la tensione misurata all'uscita dell'amplificatore del generatore di segnale, che funge da ingresso di riferimento;
- la forza effettiva fornita dallo shaker e misurata da una testa di impedenza.

Come segnali di uscita sono disponibili tre segnali di accelerazione, misurati rispettivamente:

- nella posizione in cui è applicato lo shaker (ovvero nella posizione dell'eccitazione);
- sull'ala destra vicino all'interfaccia non lineare di interesse;
- sul carico fittizio, comunque vicino all'interfaccia;

3.3 Descrizione dei segnali di ingresso applicati

I segnali di ingresso applicati al sistema tramite lo shaker a tre categorie [6]:

- **Eccitazione di tipo Sine-sweep with a linear, negative rate:** Eccitazione sinusoidale con frequenza variabile linearmente da 15 Hz a 2 Hz , con rate 0.005 Hz/s ;
- **Eccitazione di tipo Multisine with a full frequency grid:** Eccitazione periodica con banda compresa tra $2\text{ Hz} - 15\text{ Hz}$;
- **Eccitazione di tipo Multisine with a random frequency grid:** Eccitazione periodica con banda compresa tra 1 Hz e 60 Hz con sole armoniche dispari.

3.3.1 Eccitazione di tipo "Sine-sweep with a linear, negative rate"

Sono state applicate eccitazioni sinusoidali con frequenza variabile linearmente con un rate di $0,05\text{ Hz/s}$ (sweep down). L'intervallo di frequenze va da un massimo di 15 Hz ad un minimo di 2 Hz [6].

Come dati di riferimento sono stati forniti sette segnali di ingresso con sette rispettivi livelli di ampiezza differenti. Il livello più basso, con un'ampiezza di 4.8 N , può essere considerato come un set di dati lineare. I tre segnali successivi vengono forniti con tre livelli di ampiezza più elevati, in modo da innescare oscillazioni in regime non lineare e corrispondono rispettivamente a 28.8 N , 67.0 N e 95.6 N . Gli ultimi tre segnali di ingresso, di ampiezza rispettivamente pari a 19.2 N , 57.6 N e 86.0 N , sono stati proposti per testare i modelli stimati.

3.3.2 Eccitazione di tipo "Multisine with a full frequency grid"

Questa seconda categoria di ingressi è composta da segnali periodici le cui armoniche appartengono ad un range di frequenze comprese tra 2 Hz e 15 Hz .

Per ogni segnale di ingresso sono stati acquisiti nove periodi. Il numero di punti per periodo è 8192. Analogamente al caso precedente, vengono considerati sette livelli di eccitazione. Il primo segnale presente del Dataset ha un valore RMS di $12.4 N$ e può essere considerato in regime di oscillazione lineare. I tre ingressi successivi, che portano le oscillazioni in regime non lineare, hanno livelli RMS rispettivamente pari a $36.8 N$, $73.6 N$ e $97.8 N$. Gli ultimi tre segnali presenti all'interno del Dataset sono proposti, in questo studio, come segnali di test per il modello e corrispondono ai livelli RMS $24.6 N$, $61.4 N$ e $85.7 N$.

3.3.3 Eccitazione di tipo "Multisine with a random frequency grid"

L'ultima categoria di ingressi è stata ottenuta considerando solo le frequenze dispari. Inoltre, all'interno di ciascun gruppo di quattro armoniche dispari successive, 1 armonica è stata casualmente eliminata. La banda di frequenze interessate appartiene all'intervallo $1 Hz - 60 Hz$.

Per ogni livello, sono stati acquisiti tre periodi del segnale ma, diversamente dai due casi precedenti, sono state registrate dieci realizzazioni per livello variando randomicamente, in ogni realizzazione, l'armonica dispari scartata. Il numero di punti per periodo è 16384 e i dati sono stati campionati ad una frequenza di $200 Hz$.

Essendo disponibili nel dataset molteplici realizzazioni, il numero di livelli misurati per questa categoria di ingressi è stato ridotto a tre, ovvero $12.2 N$, $49.0 N$ e $97.1 N$ (espressi come valori RMS). Questi 3 livelli comportano oscillazioni non lineari.

Nello studio in esame viene suggerito di utilizzare, per ogni livello, nove realizzazioni per la stima e una realizzazione per testare il modello.

3.4 Elaborazione del Dataset

Per rendere compatibili i diversi dataset con l'algoritmo basato sul polinomio di Bernstein è stato necessario effettuare diverse elaborazioni ai dati. Le elaborazioni svolte sono comuni sia al dataset "Multisine excitation with a full frequency grid" che al dataset "Sine-sweep excitation with a linear, negative rate", per cui, nella seguente trattazione, si farà riferimento al secondo e si utilizzerà genericamente il termine dataset (sottintendendo il dataset "Sine-sweep") in modo da evitare inutili ripetizioni.

I dati disponibili all'interno del dataset sono strutturati come sei coppie di matrici ingresso-uscita (indicate in ordine crescente da "Level2" a "Level7"). Le due matrici di ogni coppia hanno dimensioni rispettivamente pari a 1×108477 , 3×108477 (eccetto "Level4" composto da due matrici di dimensioni rispettivamente pari a 1×116736 , 3×116736). Le tre righe delle matrici di uscita

corrispondono singolarmente ai valori misurati dai tre sensori di accelerazione, come precedentemente indicato nei paragrafi 3.3.1 e 3.3.2.

L'identificazione del sistema non lineare è avvenuta considerando singolarmente i tre segnali di accelerazione, ovvero suddividendo ognuna delle sei matrici di uscita in tre array di dimensione 1×108477 , contenenti rispettivamente i segnali misurati dal primo, secondo e terzo accelerometro. Questa scelta conduce ovviamente alla stima di tre distinte relazioni non lineari di input-output che legano l'ingresso con ognuno dei segnali di uscita disponibili. Considereremo nel seguito il legame tra l'ingresso di tipo "Sine-sweep" ed uno qualsiasi dei tre segnali d'accelerazione, ricordando che la procedura può essere ripetuta in maniera speculare per il test sui due segnali di accelerazione rimanenti.

Di seguito il codice che implementa le elaborazioni appena descritte:

```
1 acc_sens=1;
3 dataset="SineSw";
5 for i=2:7
7     data_path=strcat("./F16_original_data/F16Data_",dataset,"_Level",int2str(i),".mat");
9     load(data_path);
11    if mod(i,2)==0
13        X_row_train_orig=cat(2,X_row_train_orig,Force);
15        Y_row_train_orig=cat(2,Y_row_train_orig,Acceleration(acc_sens,:));
17    else
19        X_row_test_orig=cat(2,X_row_test_orig,Force);
21        Y_row_test_orig=cat(2,Y_row_test_orig,Acceleration(acc_sens,:));
23    end
25 end
```

la variabile "dataset" permette di selezionare uno dei due ingressi precedentemente descritti ai paragrafi 3.3.1, 3.3.2 mentre la variabile "sens_acc" può essere utilizzata per scegliere uno dei tre segnali di accelerazione. In accordo con quanto suggerito al paragrafo 3.3.1 i livelli Level2, Level4 e Level6 sono stati utilizzati per il training ed i tre livelli rimanenti (3,5,7) per il test. Nel listato appena mostrato, i vari array di ingresso ed uscita sono stati concatenati al fine di semplificare le successive elaborazioni.

In seguito alle precedenti manipolazioni si ottengono 2 coppie di array che rappresentano rispettivamente il training ed il test set.

Per applicare la regressione al sistema in esame è stato necessario suddividere i segnali di ingresso ed uscita in un certo numero di frame in quanto, per operare efficacemente, l'algoritmo necessita di un elevato numero di osservazioni. La lun-

ghezza di ogni frame incide sulle prestazioni del sistema ed è stata quindi variata durante la sperimentazione. I migliori risultati sono stati ottenuti suddividendo le due sequenze di train (input e output), composte entrambe da $N = 333690$ campioni, in $n = 5559$ frame di lunghezza $l = 200$ campioni e con un overlap di 140 campioni. Questa procedura di frammentazione del dataset presupporrebbe la stazionarietà del sistema che, in questo caso, può essere assunta accettando un piccolo errore di approssimazione nel fitting.

```

1  j=1;
3  for i=1:(num_columns-overlap):size(X_row_train_orig,2)
5      try
7          X_multiple_rows(j,:)=X_row_train_orig(i:(i+num_columns-1));
9          Y_multiple_rows(j,:)=Y_row_train_orig(i:(i+num_columns-1));
11         j=j+1;
13     catch error
15         break
17     end
19 end
21 X_train=X_multiple_rows;
23 Y_train=Y_multiple_rows;

```

In seguito alla ripartizione del dataset, si ricava una coppia di matrici di train (input-output) di dimensioni 5559x200. Il sistema che è stato considerato durante la sperimentazione è un sistema dinamico, quindi la funzione che lega gli ingressi e le uscite agisce direttamente anche sulla variabile tempo. Come spiegato ampiamente nel paragrafo 2.3, la regressione può essere applicata per stimare una funzione non lineare indipendente dal tempo per cui è stato necessario esprimere i segnali di ingresso ed uscita come coordinate rispetto agli autovettori delle rispettive matrici di covarianza. In seguito alla decomposizione, l'algoritmo è stato applicato per stimare la relazione tra le coordinate. Viene riportato in seguito il codice matlab relativo alle elaborazioni appena descritte:

```

Ut=cat(1,X_train,X_test);
2
Yt=cat(1,Y_train,Y_test);
4
Ruu=cov(Ut);
6
Ryy=cov(Yt);
8
[psi, lambda]=eig(Ruu);
10
[fi, gamma]=eig(Ryy);
12
for i=1:size(Ut,1)

```

```

14     X(i,:) = psi \ (Ut(i,:)');
16     K(i,:) = fi \ (Yt(i,:)');
18 end

```

Il codice implementa la scomposizione in accordo con le equazioni 2.7, 2.8, 2.9 ed 2.10. In seguito ad un'analisi degli autovalori della matrice di covarianza degli ingressi è stata effettuata una PCA riducendo il numero di componenti dei segnali di ingresso a $k = 14$. Anche in questo caso il numero di componenti è stato variato durante la sperimentazione ed i risultati migliori sono stati ottenuti per $k = 14$.

3.5 Risultati della sperimentazione

I grafici e i risultati che verranno mostrati in seguito sono relativi alla sperimentazione sul dataset "Sine-sweep excitation with a linear, negative rate" e "Multisine excitation with a full frequency grid".

Nella seguente tabella riassuntiva sono riportati i risultati ottenuti al termine della sperimentazione:

Dataset	Accelerometro	MSE	RMSE	Max grado pol.	t. test (s)
Sine-sweep	1	0.067799	0.260383	73	1.407536
Sine-sweep	2	0.186809	0.432214	73	1.427074
Sine-sweep	3	0.234781	0.484542	73	0.923690
Multisine	1	0.114293	0.338072	51	0.620390
Multisine	2	0.228674	0.478199	51	0.621560
Multisine	3	0.297928	0.545828	51	0.629466

3.5.1 Hardware utilizzato durante la sperimentazione

I risultati dei test mostrati in tabella sono stati ottenuti effettuando la sperimentazione con un calcolatore con le seguenti specifiche:

- processore: AMD Ryzen 7 3700X, 3593MHz 8-Core, 16 Thread;
- ram: Viper Steel DDR4 3733MHz 16GB (2x8GB);
- scheda madre: ASUS TUF, Chipset: X570;
- scheda video: Nvidia Geforce gtx 1660 super;
- sistema operativo: Windows 10 Professional;

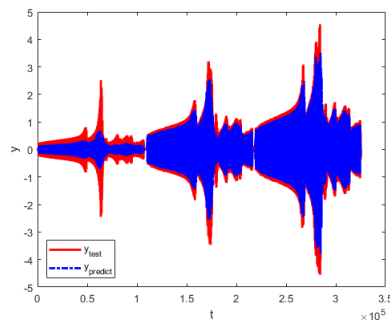


Figura 3.4: Particle Bernstein (features=14, order = 73) - dominio del tempo: output y in funzione del tempo discreto t ; $t_{test} = 1.407536$ s, numero campioni di test = 333396; dataset di riferimento: Sine-sweep; segnale di uscita: segnale misurato dall'accelerometro nella posizione in cui applicato lo shaker.

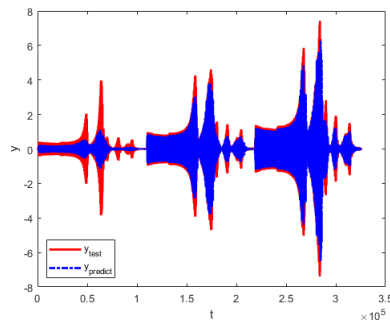


Figura 3.5: Particle Bernstein (features = 14, order = 73) - dominio del tempo: output y in funzione del tempo discreto t ; $t_{test} = 1.427074$ s, numero campioni di test = 333690; dataset di riferimento: Sine-sweep; segnale di uscita: segnale misurato dall'accelerometro posizionato sull'ala destra vicino all'interfaccia non lineare di interesse.

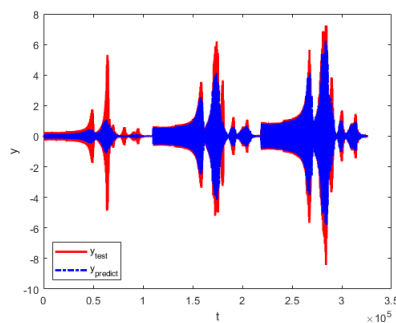


Figura 3.6: Particle Bernstein (features = 14, order = 73) - dominio del tempo: output y in funzione del tempo discreto t ; $t_{test} = 0.923690$ s, numero campioni di test = 333690; dataset di riferimento: 3.3.1; segnale di uscita: segnale misurato dall'accelerometro posizionato sul carico fittizio, vicino all'interfaccia

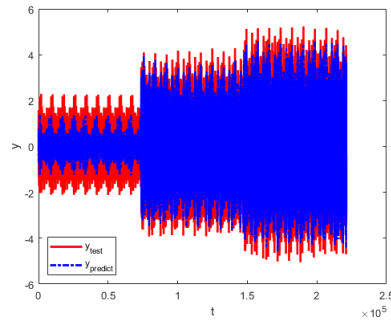


Figura 3.7: Particle Bernstein (features = 20, order = 51) - dominio del tempo: output y in funzione del tempo discreto t ; $t_{test} = 0.620390$ s, numero campioni di test = 221184; dataset di riferimento: Multisine; segnale di uscita: segnale misurato dall'accelerometro nella posizione in cui applicato lo shaker.

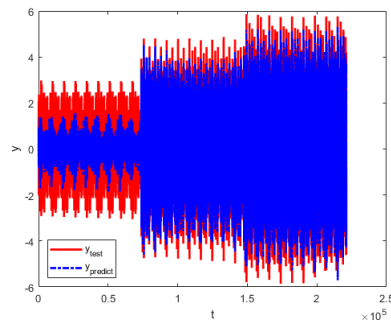


Figura 3.8: Particle Bernstein (features=20, order = 51) - dominio del tempo: output y in funzione del tempo discreto t ; $t_{test} = 0.621560$ s, numero campioni di test = 221184; dataset di riferimento: Multisine; segnale di uscita: segnale misurato dall'accelerometro posizionato sull'ala destra vicino all'interfaccia non lineare di interesse.

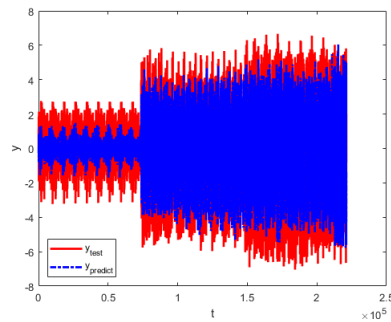


Figura 3.9: Particle Bernstein (features=20, order = 51) - dominio del tempo: output y in funzione del tempo discreto t ; $t_{test} = 0.629466$ s, numero campioni di test = 221184; dataset di riferimento: Multisine; segnale di uscita: segnale misurato dall'accelerometro posizionato sul carico fittizio, vicino all'interfaccia.

Capitolo 4

Conclusioni e possibili sviluppi futuri

4.1 Conclusioni

In base agli scopi prefissati ed in relazione ai risultati ottenuti al termine della sperimentazione è possibile concludere che:

- seguendo l'approccio descritto nel paragrafo 2.3 è stato possibile identificare il sistema non lineare in esame superando le difficoltà legate alla scelta di un modello specifico di rappresentazione dei dati;
- per sfruttare adeguatamente questa nuova tecnica è stato necessario effettuare un'indagine approfondita del dataset in quanto, per ottenere una buona stima, questo metodo richiede un'accurata pre-elaborazione dei dati;
- i risultati ottenuti sono migliori per il dataset "Sine-sweep" e ciò è giustificato dal maggior numero di campioni disponibili (si ricorda che il dataset "Sine-sweep" è composto da $N = 333690$ campioni, mentre il "Multisine" da $M = 221184$ campioni);
- sistematicamente, le curve predette durante la sperimentazione presentano un disadattamento di ampiezza in regime di piccole oscillazioni del sistema (ovvero là dove le non linearità sono meno pronunciate).
- i migliori risultati sono stati ottenuti con l'ingresso di tipo "Sine_sweep" nonostante il sistema presenti un gran numero di risonanze quando sottoposto ad ingressi di questo tipo.

4.2 Possibili sviluppi futuri

Possibili sviluppi futuri potrebbero prevedere la sperimentazione sul dataset illustrato nel paragrafo 3.3.3, non utilizzato in questo lavoro in quanto, essendo strutturato in maniera differente, avrebbe richiesto un'elaborazione alternativa. Come specificato precedentemente, le prestazioni dell'algoritmo dipendono fortemente dalle pre-elaborazioni svolte sul dataset. In particolare l'accuratezza (MSE) nel fitting dipende da:

- lunghezza dei frame;
- overlap;
- numero di componenti;
- grado del polinomio.

Durante la sperimentazione è stata osservata una relazione inversa tra il numero di componenti dell'ingresso (estratte in seguito alla PCA) ed il massimo grado del polinomio ottenibile. L'MSE dipende da entrambi i parametri, quindi se da una parte aumentare il numero di componenti contribuisce ad una ricostruzione più accurata del segnale, dall'altra ciò costringe ad abbassare il grado del polinomio che, come spiegato nel paragrafo 2.5, deve essere il più alto possibile. Durante la sperimentazione è stato cercato quindi il miglior compromesso tra questi due parametri.

In Fig. 4.1 e Fig. 4.2 vengono rispettivamente mostrati gli andamenti dell'MSE (quello corrispondente al grado massimo del polinomio) al variare del numero di componenti per il dataset di tipo "Sine_sweep" e "Multisine".

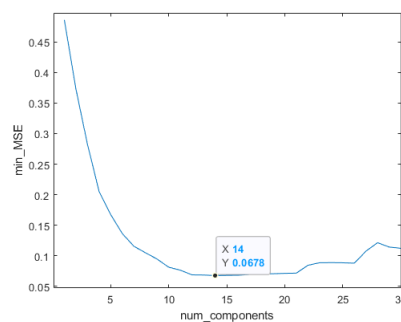


Figura 4.1: Minimo MSE al variare del numero di features. Dataset:Sine-sweep; ; Lunghezza frame=200; Overlap=140, numero ottimo di componenti = 14 -> grado massimo corrispondente del polinomio = 73; sensore accelerazione : sensore posizionato nel punto di applicazione dello shaker

Un'importante osservazione da fare è che, cambiando la sequenza di uscita, ovvero

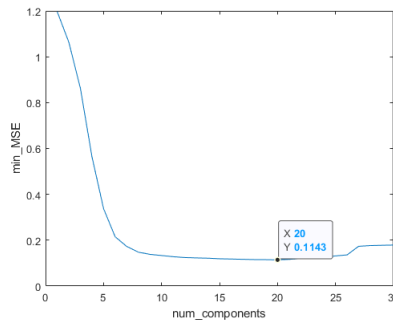


Figura 4.2: Minimo MSE al variare del numero di features. Dataset: Multisine; Lunghezza frame = 200; Overlap=150; grado massimo del polinomio = 51, sensore accelerazione : sensore accelerazione : sensore posizionato nel punto di applicazione dello shaker

scegliendo il secondo o il terzo accelerometro, le curve hanno lo stesso andamento e presentano il minimo valore del MSE in corrispondenza dello stesso numero di componenti e con lo stesso grado massimo del polinomio.

Questo risultato sembrerebbe suggerire che il compromesso ottimo tra numero di componenti e grado del polinomio non dipenda dalla sequenza di uscita ma esclusivamente dalla struttura dei segnali di ingresso.

I grafici sono stati ottenuti fissando la lunghezza dei frame e l'overlap. Sviluppi successivi potrebbero includere un'analisi parametrica al variare della lunghezza dei frame e dell'overlap che, in questo caso, sono stati selezionati effettuando un'analisi a campione.

Infine, un'ulteriore indagine potrebbe essere svolta al fine di individuare la causa del disadattamento di ampiezza in regime di piccole oscillazioni. Come mostrato infatti nelle curve ottenute al termine della sperimentazione (da Fig. 3.4 a Fig. 3.9, le zone in cui l'ampiezza del segnale di uscita è più contenuta corrispondono alle zone di peggiore previsione e determinano un aumento del MSE.

Bibliografia

- [1] Nonlinear System Identification Benchmarks. <http://www.nonlinearbenchmark.org/>. [Online; accessed 08-July-2020].
- [2] Serie di Volterra. https://it.wikipedia.org/wiki/Serie_di_Volterra. [Online; accessed 09-July-2020].
- [3] Nonlinear system identification. https://en.wikipedia.org/wiki/Nonlinear_system_identification, Last edit: 16-May-2020. [Online; accessed 08-July-2020].
- [4] L. Falaschetti C. Turchetti G. Biagetti, P. Crippa. "Machine learning regression based on particle bernstein polynomials for nonlinear system identification". *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Sept. 2017.
- [5] L. Falaschetti C. Turchetti G. Biagetti, P. Crippa. "A Machine Learning Approach to the Identification of Dynamical Nonlinear Systems". *2019 IEEE 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5, Sept. 2019.
- [6] Jean-Philippe Noël and Maarten Schoukens. F-16 aircraft benchmark based on ground vibration test data. In *2017 Workshop on Nonlinear System Identification Benchmarks*, volume 22, page 30, 2017.

Elenco delle figure

2.1	Sistemi Hammerstein: la funzione $g(w)$ costituisce un blocco non lineare statico e precede la $G(s)$, che rappresenta una rete lineare dinamica nel dominio di Laplace.	9
2.2	Sistemi Wiener: la funzione $G(s)$ costituisce un blocco lineare dinamico nel dominio di Laplace e precede la funzione $f(x)$ che rappresenta le non linearità statiche del sistema	10
2.3	Sistemi Hammerstein-Wiener: la funzione $G(s)$ è una funzione lineare dinamica, preceduta e seguita da due blocchi non lineari statici $g(w)$ ed $f(x)$	10
2.4	Sistemi Wiener-Hammerstein: la funzione $f(u)$ è una funzione non lineare statica, seguita e preceduta da due blocchi dinamici lineari rappresentati rispettivamente da $G_1(s)$ e da $G_2(s)$ nel dominio di Laplace	11
2.5	Semplice modello strutturale di una rete neurale artificiale	11
3.1	F-16; Test di vibrazione a terra presso base militare di Saffraanenber.	17
3.2	Carico fittizio montato sull'ala destra dell'F16	18
3.3	Shaker applicato all'ala destra	18
3.4	Particle Bernstein (features=14, order = 73) - dominio del tempo: output y in funzione del tempo discreto t; ttest = 1.407536 s, numero campioni di test = 333396; dataset di riferimento: Sine-sweep; segnale di uscita: segnale misurato dall'accelerometro nella posizione in cui applicato lo shaker.	24
3.5	Particle Bernstein (features = 14, order = 73) - dominio del tempo: output y in funzione del tempo discreto t; ttest = 1.427074 s, numero campioni di test = 333690; dataset di riferimento: Sine-sweep; segnale di uscita: segnale misurato dall'accelerometro posizionato sull'ala destra vicino all'interfaccia non lineare di interesse.	24
3.6	Particle Bernstein (features = 14, order = 73) - dominio del tempo: output y in funzione del tempo discreto t; ttest = 0.923690 s, numero campioni di test = 333690; dataset di riferimento: 3.3.1; segnale di uscita: segnale misurato dall'accelerometro posizionato sul carico fittizio, vicino all'interfaccia	24

- 3.7 Particle Bernstein (features = 20, order = 51) - dominio del tempo: output y in funzione del tempo discreto t; ttest = 0.620390 s, numero campioni di test = 221184; dataset di riferimento: Multisine ; segnale di uscita: segnale misurato dall'accelerometro nella posizione in cui applicato lo shaker. 25
- 3.8 Particle Bernstein (features=20, order = 51) - dominio del tempo: output y in funzione del tempo discreto t; ttest = 0.621560 s, numero campioni di test = 221184; dataset di riferimento: Multisine; segnale di uscita: segnale misurato dall'accelerometro posizionato sull'ala destra vicino all'interfaccia non lineare di interesse. 25
- 3.9 Particle Bernstein (features=20, order = 51) - dominio del tempo: output y in funzione del tempo discreto t; ttest = 0.629466 s, numero campioni di test = 221184; dataset di riferimento: Multisine; segnale di uscita: segnale misurato dall'accelerometro posizionato sul carico fittizio, vicino all'interfaccia. 25
- 4.1 Minimo MSE al variare del numero di features. Dataset:Sine-sweep; ; Lunghezza frame=200; Overlap=140, numero ottimo di componenti = 14 -> grado massimo corrispondente del polinomio = 73; sensore accelerazione : sensore posizionato nel punto di applicazione dello shaker 28
- 4.2 Minimo MSE al variare del numero di features. Dataset: Multisine; Lunghezza frame = 200; Overlap=150; grado massimo del polinomio = 51, sensore accelerazione : sensore accelerazione : sensore posizionato nel punto di applicazione dello shaker 29