

UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA



Corso di Laurea Triennale in Ingegneria Elettronica

**Identificazione di sistemi dinamici non lineari reali con il metodo
basato sui polinomi di Bernstein**

**Real nonlinear dynamic systems identification with method based
on Bernstein polynomials**

Relatore:

Dott. TURCHETTI CLAUDIO

Laureando:

BAGNARELLI MICHAEL

Correlatore:

Dott.ssa FALASCHETTI LAURA

A.A. 2019/2020

INDICE

1 Introduzione	7
1.1 Obiettivo della tesi	7
1.2 Struttura della tesi	8
2 Identificazione di un sistema non lineare	9
2.1 Generalità	9
2.2 Modelli per identificazione di sistemi non lineari	11
2.2.1 Modelli basati sulla serie di Volterra	11
2.2.2 Modelli strutturati a blocchi	12
2.2.3 Modelli basati sulle reti neurali artificiali	13
2.2.4 Modello NARMAX	15
2.3 Approccio alternativo per l'identificazione di un sistema non lineare	17
2.3.1 Trasformare di un operatore non lineare in una funzione non lineare	17
2.3.2 Polinomio di Bernstein	21
2.3.3 Regressione del polinomio di Bernstein	23
3 Sperimentazione	25
3.1 Struttura Dataset	25
3.1.1 Eccitazione di tipo Sine-Sweep with a linear, negative rate .	28

3.1.2	Eccitazione di tipo Multisine with a full frequency grid . . .	29
3.1.3	Eccitazione di tipo Sine-Sweep with a random frequency grid	29
3.2	Analisi Dataset	30
3.2.1	Analisi del dataset Sine-Sweep with a linear, negative rate .	32
3.2.2	Analisi del dataset Multisine with a full frequency grid . . .	34
3.2.3	Analisi del dataset Multisine with a random frequency grid	34
3.3	Risultati Sperimentazione	35
3.3.1	Risultati del dataset Sine-Sweep with a linear, negative rate	37
3.3.2	Risultati del dataset Multisine with a full frequency grid . . .	38
3.3.3	Risultati del dataset Multisine with a random frequency grid	39
3.4	Osservazione dei risultati	40
4	Conclusioni	43
	Bibliografia	45
	Figure	47

CAPITOLO 1

INTRODUZIONE

1 Obiettivo della tesi

L'argomento principale di questa tesi è l'identificazione di sistemi dinamici non lineari reali, in particolare in questo elaborato verranno analizzati i vantaggi e gli svantaggi dei metodi che consentono di effettuare questa operazione. La conoscenza del sistema è una parte essenziale per comprendere i vari "fenomeni" che ci circondano: per questo negli ultimi anni sono state sviluppate diverse tecniche che hanno come obiettivo misurare e far comprendere la dinamica di questi sistemi. Tuttavia molte di queste tecniche richiedono un'analisi molto approfondita e complessa per individuare un modello adatto alla rappresentazione del sistema. Infatti l'individuazione del modello è una fase fondamentale in queste tecniche e in molti casi richiede un costo computazionale elevato. Questo elaborato ha l'obiettivo di presentare un metodo per l'identificazione di sistemi dinamici non lineari che non richiede l'individuazione di un modello come richiesto in altre tecniche. Questo approccio alternativo permette di trasformare il problema dinamico in un problema di regressione di una funzione non lineare, statica. Il grande vantaggio di questo approccio è quello che permette, dopo la trasformazione, di approssimare la funzione ottenuta con uno dei numerosi algoritmi di machine learning. In questo caso è stata scelta la rappresentazione polinomiale basata sui polinomi di Bernstein che permettono di ricavare i coefficienti del polinomio limitando il costo computazionale e riducendo i tempi di addestramento.

2 Struttura della tesi

La tesi è strutturata nel modo seguente:

- nella prima parte vengono descritti i metodi più diffusi per l'identificazione dei sistemi non lineari e ne verranno analizzati i vantaggi e gli svantaggi. Di seguito verrà introdotto dettagliatamente l'approccio che è stato utilizzato in questo elaborato, elencandone tutte le caratteristiche.
- nella parte centrale verrà descritto il dataset in dotazione per questa analisi. Verrà descritta l'elaborazione fatta sui dati per ottenere i risultati che verranno poi riportati nel corso dell'elaborato ed inoltre sarà riportata e descritta una parte del codice Matlab utilizzato nell'analisi.
- nell'ultima parte vengono riportati e discussi i risultati ottenuti

CAPITOLO 2

IDENTIFICAZIONE DI UN SISTEMA NON LINEARE

2.1. Generalità

L'identificazione di un sistema non lineare si riferisce al problema di costruire e misurare la relazione matematica che lega un ingresso u e ad un'uscita y di un sistema dinamico sconosciuto. Quindi, per trattare un qualunque sistema fisico con un approccio ingegneristico, è necessario disporre di un modello matematico del sistema, cioè di un insieme di relazioni matematiche che descrivano il suo funzionamento. In questo modo è possibile prevederne il comportamento sia qualitativamente che quantitativamente [3].

Ogni metodo che mira a identificare un sistema non lineare segue comunemente quattro step:

- **raccolta dei dati:** selezione dei dati in maniera appropriata. In questa fase viene svolta anche una prima elaborazione dei dati raccolti.
- **modellazione del sistema:** scelta della struttura del modello. Questa fase può essere eseguita seguendo due approcci contrapposti: il primo prevede la deduzione del modello a partire da leggi fisiche generali, mentre il secondo consiste nel modellare il sistema a partire dalle sole misurazioni dei suoi ingressi e delle sue uscite, senza una specifica conoscenza del fenomeno.

- **identificazione del modello:** applicazione degli algoritmi di identificazione al fine di determinare i parametri incogniti da cui il modello dipende.
- **validazione del modello:** il modello individuato viene sperimentato per valutarne la bontà. In questa fase ci si aspetta che esista una corrispondenza tra i dati effettivi del modello ed i risultati computazionali ottenuti.

Parlando dell'identificazione di sistemi fisici si deve tener conto anche del problema del rumore. Infatti in ogni sistema, oltre ai fenomeni che lo caratterizzano e che vogliamo studiare, sono presenti altri fenomeni che comportano una deviazione del comportamento rispetto a quello "teorico". Questi fenomeni sono detti disturbi o rumore. Le misure eseguite sui sistemi fisici sono sempre affette da rumore, nel senso che non corrispondono al comportamento ideale del sistema e questo complica i tentativi di identificazione. Da qui si comprende che la robustezza al rumore è una caratteristica importante per un qualunque metodo di identificazione.

Negli ultimi decenni sono stati proposti numerosi e differenti approcci per affrontare questo problema: l'identificazione dei sistemi non lineari si è sviluppata concentrandosi su classi specifiche di sistemi per le quali sono stati adottati diversi modelli di base, tra i più popolari ci sono:

- modelli basati sulla Serie di Volterra
- modelli strutturati blocchi
- reti neurali artificiali
- modelli NARMAX

2.2. Modelli per identificazione dei sistemi non lineari

2.2.1. Modelli basati sulla 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. Supponiamo di avere un sistema continuo e tempo-invariante con $x(t)$ come ingresso e $y(t)$ come uscita, il sistema può essere espanso in serie di Volterra come:

$$y(t) = h_0 + \sum_{n=1}^N H_n x(t) \quad (2.1)$$

dove

$$H_n x(t) = \int_a^b \dots \int_a^b h_n(\tau_1, \dots, \tau_n) \prod_{j=1}^n x(t - \tau_j) d\tau_j \quad (2.2)$$

e con $a, b \in \mathcal{R} \cup \{-\infty, \infty\}$ e $N \in \{0, 1, 2, \dots\} \cup \{+\infty\}$.

Le funzioni $h_n(\tau_1, \dots, \tau_n)$, h_0 sono dette kernel di Volterra dell' n ordine e possono essere viste come una generalizzazione della risposta impulsiva. Le serie di Volterra permettono di ricostruire anche l'effetto memoria in quanto approssimano la risposta di un sistema non lineare che dipende da tutto l'andamento temporale dell'ingresso [4].

Il più grande problema dei metodi che si basano sulla serie di Volterra è dovuto alla multidimensionalità dei kernel di Volterra che ne rendono complicato il calcolo computazionale.

In un primo momento, per superare la difficoltà computazionale, i metodi che si basano su questo approccio hanno utilizzato la serie di Volterra troncata, cioè prendendo in considerazione solo i primi due kernel, quello lineare e quello quadratico. Però queste approssimazioni non sono sempre accettabili per identificare un sistema reale; per questo motivo, successivamente, il metodo è stato esteso per includere anche i kernel di ordine superiore.

2.2.2. Modelli strutturati a blocchi

Per superare il problema di calcolo della serie di Volterra è stato proposto il modello di identificazione strutturato a blocchi. In questo tipo di modello, il sistema viene descritto da una particolare sequenza di blocchi in cascata. Ogni blocco rappresenta una particolare caratteristica del sistema e in base alla disposizione dei vari blocchi all'interno del sistema si possono avere diverse realizzazioni [6]. Un esempio di modello molto usato è quello di Hammerstein il quale è costituito da un blocco non lineare statico seguito da un elemento lineare dinamico (figura 2.1). Un secondo modello a blocchi è quello di Wiener il quale è opposto al modello di Hammerstein: in questo caso il blocco lineare dinamico è posto prima del blocco con la caratteristica non lineare statica (figura 2.2). L'ultimo esempio riportato di sistema a blocchi è il modello di Wiener-Hammerstein il quale è costituito da un blocco lineare dinamico inserito tra due blocchi non lineari statici (figura 2.3).

Questo tipo di modello approssima piuttosto bene i sistemi fisici non lineari e per questo viene utilizzato in molti campi dell'ingegneria. Tuttavia, un problema legato a tutti i sistemi a blocchi è quello che, in generale, esistono infiniti insiemi di parametri che forniscono la stessa relazione ingresso/uscita per il sistema. Infatti modificando i parametri in modo da moltiplicare le uscite dei vari blocchi per opportune costanti, si possono ottenere infiniti sistemi che, partendo da un certo ingresso forniscono la stessa uscita per il sistema complessivo, anche se le uscite dei singoli blocchi sono diverse.

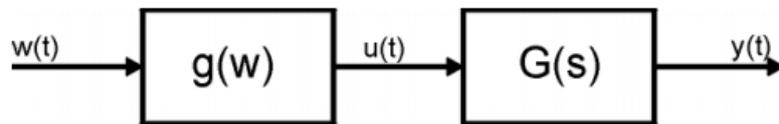


Figura 2.1: Modello a blocchi di Hammerstein

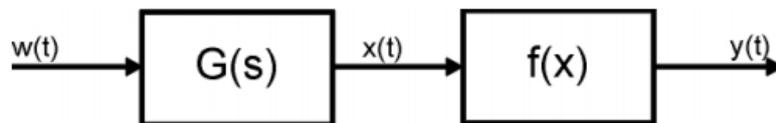


Figura 2.2: Modello a blocchi di Wiener

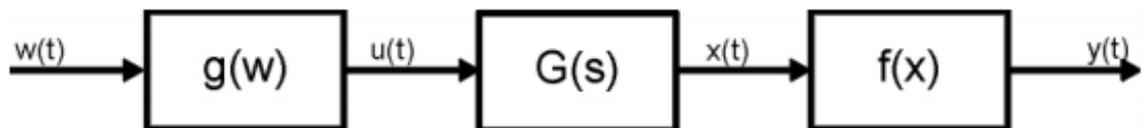


Figura 2.3: Modello a blocchi di Hammerstein-Wiener

2.2.3. Modelli basati sulle reti neurali artificiali

La rete neurale è un modello computazionale che cerca di imitare, vagamente, una rete neurale biologica in cui l'elaborazione delle informazioni viene svolta da un gran numero di semplici unità che, collegate tra loro, formano una rete più complessa [5]. Tale modello è costituito da un gruppo di interconnessioni formate da neuroni artificiali e da processi che utilizzano un approccio basato su varie connessioni che consentono di elaborare le informazioni ricevute. Nella maggior parte dei casi una rete neurale artificiale è un sistema adattivo che cambia la propria struttura in base alle informazioni esterne o interne che scorrono attraverso la rete stessa durante la fase di apprendimento.

Una rete neurale artificiale riceve segnali esterni su uno stato di nodi (unità di elaborazione) d'ingresso, ciascun dei quali è collegato con numerosi nodi interni, organizzati in più livelli. Ogni nodo elabora i segnali e trasmette il risultato ai nodi successivi. 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. La fase di apprendimento della rete neurale è quindi un processo fondamentale per la riuscita dell'identificazione del sistema. Esistono diversi paradigmi di apprendimento, ciascuno corrispondente ad un particolare compito astratto di apprendimento. I principali sono: l'apprendimento supervisionato e l'apprendimento non supervisionato.

- nell'apprendimento supervisionato, la rete viene addestrata operando sulla differenza tra l'output effettivo e l'output desiderato della rete, cercando di minimizzare il più possibile l'errore di previsione relativo all'insieme di addestramento. Se l'addestramento ha successo, la rete impara a riconoscere la relazione incognita che lega le variabili d'ingresso a quelle di uscita, ed è quindi in grado di fare previsioni anche laddove l'uscita non è nota a priori.
- nell'apprendimento non supervisionato, gli algoritmi di addestramento fanno esclusivamente riferimento ad un insieme di dati che include le sole variabili di ingresso. L'obiettivo di questi algoritmi è quello di raggruppare i dati di ingresso e di individuare pertanto degli opportuni cluster rappresentativi dei dati stessi, facendo uso tipicamente di metodi topologici o probabilistici.

Le reti neurali possono essere utilizzate per studiare sia i problemi statici, cioè quelli che includono il riconoscimento, la classificazione e l'approssimazione dei modelli, sia i problemi dinamici. Nonostante le reti neurali abbiano diversi vantaggi, come la loro semplicità, la facilità di addestramento e le eccellenti proprietà di approssimazione, possiedono anche l'inconveniente di essere, a volte, poco chiare. Infatti il gran numero di nodi rende complicata l'identificazione del modello. Una rete neurale è in grado di generare un risultato valido, o comunque con un'alta probabilità di essere accettabile, ma non è possibile spiegare il come e il perché tale risultato sia stato generato.

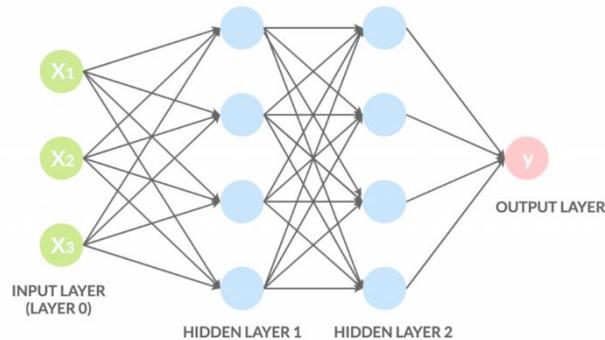


Figura 2.4: modello strutturale di una rete neurale artificiale

2.2.4. Metodo NARMAX

Nel dominio del tempo discreto un modello che ha molto successo per l'approccio ai sistemi non lineari è il modello NARMAX. In questo modello il sistema viene modellato in termini di espansione di un funzione non lineare di input, output ed errori di previsione ritardati. Il modello può essere quindi definito:

$$y(t) = F[y(k-1), y(k-2), \dots, y(k-n_y), u(k-d), u(k-d-1), \dots, u(k-d-n_u), e(k-1), e(k-2), \dots, e(k-n_e)] \quad (2.3)$$

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 ritardi massimi 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.

Il processo di identificazione del sistema mediante i modelli NARMAX si concentra su due obiettivi principali: il primo è quello di sviluppare un modello che approssimi il set di dati in modo da produrre minimi errori di previsione; il secondo è quello di trovare il modello più semplice che rappresenti il sistema.

Questo approccio si basa su alcuni passaggi fondamentali:

- rilevazione della struttura
- stima dei parametri
- validazione del modello
- previsione dell'output in un momento futuro
- analisi delle proprietà del sistema

Nella fase di rilevamento della struttura occorre determinare un numero consono di termini che siano in grado di rappresentare la struttura. Questa è una fase delicata in quanto una scelta eccessiva del numero di termini potrebbe causare alcuni problemi nel rilevamento del modello. Altra fase importante è quella della stima dei parametri: eccedere "ingenuamente" nella stima dei parametri potrebbe causare problemi computazionali.

Nel tempo discreto, il modello NARMAX rappresenta uno degli approcci che ha avuto maggior successo per l'identificazione di sistemi non lineari. Infatti tale modello, si è dimostrato molto efficiente anche in numerose applicazioni del mondo reale. Tuttavia una delle più grandi difficoltà nell'utilizzo di tale approccio è la selezione di un modello parsimonioso per quanto riguarda il numero di parametri e che riesca a rappresentare adeguatamente il sistema non lineare.

2.3. Approccio alternativo per l'identificazione di sistemi non lineari dinamici

2.3.1. Trasformare un operatore non lineare in una funzione non lineare

Si consideri un sistema dinamico non lineare, che trasforma un segnale di ingresso u in un segnale di uscita y , rappresentato dall'operatore T [1]:

$$y = T(u), \quad u \in \mathbb{R}^m, y \in \mathbb{R}^n \quad (2.4)$$

dove $u=(u(1), \dots, u(n))$, $y=(y(1), \dots, y(n))$ sono entrambi vettori reali dipendenti dal tempo discreto $t=1, \dots, n$.

In questa trattazione è stato considerato solo il sistema non lineare dinamico visto che è quello che ci interessa, quindi consideriamo che l'operatore T agisce sull'ingresso u e sul tempo t .

Supponiamo che l'ingresso u appartenga alla classe di segnali casuali con la relativa matrice di covarianza $R_{uu} = E\{uu^T\}$, che può essere decomposta come segue:

$$R_{uu} = \psi \Lambda \psi^T, \quad R_{uu} \in \mathbb{R}^{m \times n} \quad (2.5)$$

dove ψ è una matrice unitaria le cui colonne sono gli autovettori di R_{uu} e Λ è la matrice diagonale composta dagli autovalori della matrice. Vista la proprietà di ortogonalità della matrice ψ , il vettore generico u può essere riscritto come combinazione lineare delle colonne di ψ

$$u = \psi x \quad (2.6)$$

in cui x è definito come il vettore delle coordinate della base ψ .

$$x = \psi^T u, \quad x \in \mathbb{R}^m \quad (2.7)$$

Questo tipo di analisi può essere fatta allo stesso modo sull'uscita y

$$y = \phi k \quad (2.8)$$

$$R_{yy} = \phi \Gamma \phi^T \quad (2.9)$$

La matrice $\Gamma \in \mathbb{R}^{n \times n}$ è una matrice diagonale. Vale la pena notare che nelle rappresentazioni (2.6) e (2.8), le matrici ψ e ϕ dipendono dal tempo in quanto sono composte da autovettori, mentre x e k non dipendono dal tempo poiché rappresentano le coordinate nella base. Combinando la (2.4), (2.6) e la (2.8) consegue che:

1. L'operatore T stabilisce una corrispondenza, in generale non biunivoca, tra u e y
2. le matrici ψ e ϕ stabiliscono due corrispondenze biunivoche rispettivamente tra u e x , y e k .

Di conseguenza esisterà una corrispondenza tra x e k :

$$k = f(x) \quad (2.10)$$

dove la funzione $f(\cdot)$ è non lineare ed indipendente dal tempo.

La figura 2.5(a) fornisce una rappresentazione chiara sul significato della funzione $f(x)$ dove u , y rappresentano i segnali di ingresso e uscita nel dominio del tempo, mentre x e k nello spazio caratteristico. Si può notare che l'operatore T nel tempo si riduce in una funzione non lineare. Questa funzione riduce l'imput u nell'uscita y , quindi combinando la (2.7), (2.8) e (2.10) il risultato è

$$y = \phi f(\psi^T u) = T(u) \quad (2.11)$$

dove

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

Una rappresentazione della (2.11) è riportata nella figura 2.5(b).

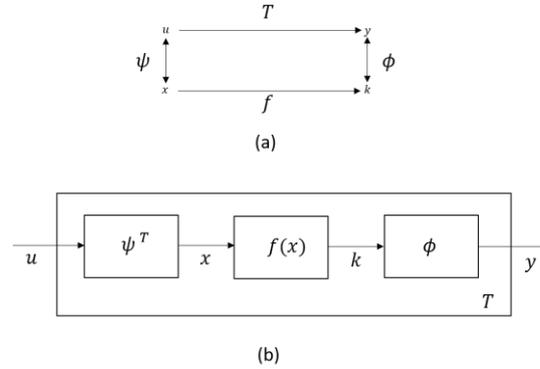


Figura 2.5: (a) Rappresentazione della corrispondenza tra u, y, x e k . (b) rappresentazione schematica della (2.11)

La stima della funzione $f(\cdot)$ in (2.10) può essere vista come un problema di regressione fornito su un set di training

$$\{\psi^T u^{(i)}, \phi^T y^{(i)}, i = 1, \dots, N\} \tag{2.13}$$

che può essere riscritto in forma matriciale

$$X = U\Psi, \quad K = Y\Phi \tag{2.14}$$

dove U, Y, X, K sono le matrici dei dati di u, y, x, k .

Utilizzando queste relazioni, la stima della (2.12) si riduce nella stima della funzione (2.10) dove vengono forniti i dati matriciali X e K rispettivamente di x e k . Per concludere, può essere utilizzata una PCA con l'obiettivo di ridurre la dimensione del problema. A tal fine, si assume che gli autovalori nella matrice Λ siano disposti in ordine decrescente $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ e gli autovalori in ψ si comportano allo stesso modo. Così ψ può essere partizionata come

$$\psi = (\psi_d \psi_n) \tag{2.15}$$

dove la matrice $\psi_d \in \mathbb{R}^{n \times d}$ contiene gli autovettori più significativi. La (2.6) può essere quindi riscritta:

$$u = \psi_d x_M + \psi_n x_n, \quad x_M \in \mathbb{R}^d \tag{2.16}$$

Infine, trascurando i termini residui $\psi_n x_n$, corrispondenti alle componenti meno significative, u può essere approssimata come:

$$u \cong \psi_d x_M \quad (2.17)$$

E questo produce

$$x_M \cong \psi_d^T u \quad (2.18)$$

dove è stato utilizzato il vettore x_M che ha una dimensione più piccola rispetto a x ($d \ll m$). Avendo sostituito u nel vettore x_M , l'identificazione del sistema non lineare si riduce nella regressione della funzione

$$\phi^T y \cong g(x_M) \quad (2.19)$$

Una volta effettuate le trasformazioni appena descritte, il problema di identificazione del sistema si riduce alla regressione di una funzione non lineare statica. La tecnica appena descritta è del tutto generale e può essere applicata a qualsiasi sistema non lineare, senza la necessità di individuare preventivamente il miglior modello della rappresentazione del fenomeno e ciò rappresenta il punto di forza di questo approccio.

2.3.2. Polinomio di Bernstein

I polinomi di Bernstein appartengono ad una classe di polinomi che grazie alle loro proprietà permettono di applicare la regressione su una funzione ottenendo risultati soddisfacenti.

Considerando una generica funzione f dipendente dalla variabile x , il polinomio di Bernstein $P(x)$ di grado n è dato dalla formula:

$$P(x) = \sum_{k=0}^n c_k B_k^n \tag{2.20}$$

dove $B_i^n(\cdot)$ gli sono elementi della base dei polinomi di Bernstein, definiti da:

$$B_i^n(x) = \binom{n}{i} x^i (1-x)^{n-i} \tag{2.21}$$

con $x \in [0,1]$.

Una prima proprietà di questi polinomi è quella per la quale i coefficienti possono essere ottenuti valutando la funzione f , che si vuole approssimare, in punti appartenenti ad una griglia fissa $\{\frac{k}{m}, k: 0,1 \dots, m\}$

$$B_m(x) = \sum_{k=0}^n f\left(\frac{k}{m}\right) b_k^m(x) \tag{2.22}$$

Una proprietà del polinomio di Bernstein $B_m(x)$ è quella che converge uniformemente per $m \rightarrow \infty$. Questa proprietà è fondamentale in quanto ci permette di determinare i coefficienti del polinomio in modo più semplice rispetto ad altre basi note [7].

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 piena e ciò corrisponde alla necessità di dover alzare il grado m del polinomio, aumentando di fatto il costo computazionale. Un altro possibile inconveniente è quello per cui le variabili m e k sono a valori interi così come i coefficienti binomiali. Queste condizioni possono essere rilassate, assumendo

che k appartenga ai numeri reali e ridefinendo questo valore con ξ . A questo punto è possibile definire un nuovo set di funzioni chiamato particle-Bernstein:

$$C_{\xi}^m(x) = \alpha_{\xi}^m x^{\xi} (1-x)^{m-\xi} = \alpha_{\xi}^m k_{\xi}^m(x) \quad (2.23)$$

con $\xi \in \mathbb{R}^1$ e $\xi \in [0, m]$.

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

$$\int_0^1 C_{\xi}^m(x) dx = 1 \quad (2.24)$$

Alcuni esempi di funzioni definiti da (2.23) per $m = 20$ e per differenti valori di ξ sono riportate nella figura 2.6.

Una proprietà importante di questi polinomi è che scegliendo opportunamente i coefficienti α_{ξ}^m , i polinomi sottendono tutti la stessa area ed esibiscono il proprio valore massimo in $x = \frac{\xi}{m}$. Inoltre, scegliendo un 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 $x = \frac{\xi}{m}$.

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

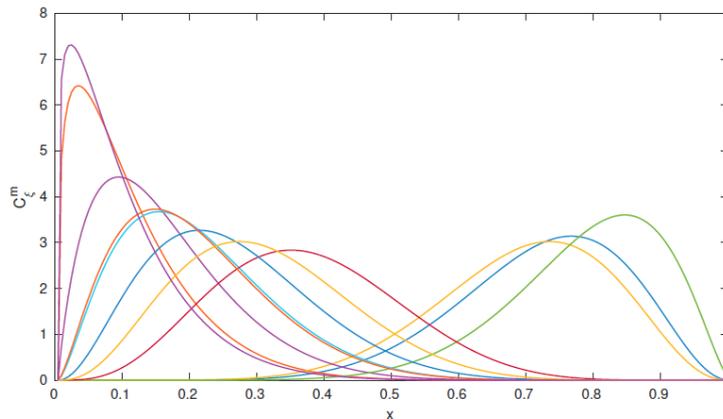


Figura 2.6: Polinomi di particle-Bernstein

2.3.3. Regressione del polinomio di Bernstein

Nel paragrafo precedente è stato mostrato come i polinomi di Bernstein possiedono la proprietà che ci permette di definire i coefficienti come i valori della funzione da approssimare nella griglia di punti fissi, così che vengano evitate perdite di tempo durante la fase training. Questo approccio può essere usato per risolvere la regressione della funzione $f(\cdot)$ nella (2.19). Seguendo questo metodo la funzione stimata $f_m(z)$ di $f(\cdot)$ in un generico punto z di test è data da:

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

Dove il set $\{x^{(j)}, f(x^{(j)}), j = 1, \dots, N\}$ rappresentano il set di dati per il training che possono essere ottenuti dai dati di ingresso e uscita $\{u^{(i)}, y^{(i)}, i = 1, \dots, N\}$ attraverso la relazione (2.14) e x_t, ξ_t sono le coordinate dei vettori $x = (x_1, \dots, x_d)$ e $\xi = z/m = (\xi_1, \dots, \xi_d)$. Nella (2.26) C_{ξ}^m è il polinomio di grado m definito come:

$$\prod_{t=1}^d x_t^{\xi_t} (1 - x)^{m - \xi_t}. \quad (2.27)$$

CAPITOLO 3

SPERIMENTAZIONE

3.1. Struttura Dataset

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

Il dataset su cui è stata svolta la sperimentazione contiene i dati sperimentali acquisiti su un velivolo F-16 a scala naturale in occasione della Siemens LMS Ground Vibration testing Master Class, tenutasi nel settembre 2014 nella base militare di Saffraanberg (Sint-Truiden, Belgio) [2].

Lo scopo principale del test svolto dalla Siemens è quello di ottenere dati sperimentali sulle vibrazioni dell'intera struttura dell'aeromobile per valutarne e migliorarne i modelli di dinamica strutturale. I test di vibrazione al suolo su un aeromobile sono spesso considerati uno dei passaggi più importanti in quanto vengono utilizzati per analizzare le dinamiche strutturali che caratterizzano il velivolo senza effettuare costosissimi test di volo; inoltre durante i test è possibile verificare anche il comportamento della struttura, in particolare dei flutter, ottenendo così una maggior sicurezza per i successivi test 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: Struttura completa di un F16

Durante la campagna di test, due carichi fittizi sono stati montati sulle punte delle ali per simulare le proprietà di massa e d'inerzia dei dispositivi reali che tipicamente equipaggiano un F-16 in volo (figura 3.2(a)).

La struttura del velivolo è stata dotata di 145 sensori di accelerazione. Uno shaker è stato posto sotto l'ala destra con lo scopo di generare il segnale di ingresso (figura 3.2(b)). La posizione dello shaker non è casuale: deve essere fatta in modo ottimale per poter misurare meglio possibile il fenomeno di interesse. In questo caso lo shaker è stato posizionato in direzione z vicino alla punta dell'ala, permettendo così di eccitare meglio la struttura.

La sorgente dominante di non-linearità della struttura dinamica si origina dalle interfacce di montaggio dei carichi fittizi. Queste interfacce sono costituite da elementi di collegamento a forma di T sul lato del carico utile e sono fatte scorrere attraverso una guida fissa sul lato dell'ala (figura 3.2(c)).



(a)



(b)



(c)

Figura 3.2: (a) Carico fittizio montato sulla punta dell'ala destra; (b) shaker fissato sotto l'ala destra; (c) collegamento posteriore dell'interfaccia di montaggio, a destra del carico.

Una prima analisi mostra che la fonte predominante della distorsione non lineare si verifica sulla connessione posteriore dell'interfaccia del carico sull'ala destra e quindi questo punto è il fulcro di questo studio.

Le misure sono state acquisite ad 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 derivante dallo shaker e misurata attraverso un'impedenza di testa applicata nel punto dell'eccitazione.

Come segnali di uscita sono disponibili tre segnali di accelerazione, i quali sono stati misurati:

- nella posizione dell'eccitazione
- sull'ala destra vicino all'interfaccia non lineare di interesse
- sul carico fittizio nella stessa interfaccia

I segnali di ingresso applicati sono divisi in tre categorie:

- 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 Hz – 15 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.

3.1.1. 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 Hz/s (sweep down). L'intervallo di frequenze va da un massimo di 15 Hz ad un minimo di 2Hz. Si è preferito utilizzare l'eccitazione sinusoidale sweep rispetto alla sinusoidale tradizionale poiché, anche se l'effetto fornito è simile, l'eccitazione sweep fornisce un tempo transitorio più breve tra ogni spettro sinusoidale rispetto a quella tradizionale.

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. Le tre eccitazioni più grandi sono state fornite per innescare e stimare le oscillazioni a regime non lineare. Questi livelli corrispondono rispettivamente a 28.8 N, 67.0 N e 95.6 N. I livelli restanti, che equivalgono rispettivamente a 19.2 N, 57.6 N e 86.0 N possono essere usati per testare il modello usando i livelli descritti precedentemente. Da questo momento, si farà riferimento ai dati derivanti da questo tipo di eccitazione con l'abbreviazione "Sine-sweep".

3.1.2. 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 tra 2 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. La parte del segnale transitoria è presente solo nel primo periodo della misura. Il primo segnale presente nel dataset ha un valore RMS di 12.4 N e può essere considerato in regime di oscillazione lineare. I tre ingressi non lineari usati per stimare il modello hanno livelli RMS rispettivamente pari a 36.8 N, 73.6 N e 97.8 N e sono accoppiati con i loro corrispondenti dati di test che corrispondono rispettivamente ai livelli RMS 24.6 N, 61.4 N e 85.7 N. Per brevità nella stesura, si farà riferimento a questo dataset con il termine "FullMSine".

3.1.3. Eccitazione di tipo Multisine with a random frequency grid

Questa categoria di ingressi è stata ottenuta considerando solo eccitazioni con frequenze dispari. Inoltre, all'interno di ogni gruppo composto da 4 armoniche dispari successive, una frequenza è stata casualmente eliminata ed è utilizzata come rilevamento delle non linearità dispari. In questo set, sono state utilizzate bande di frequenza appartenenti al range di frequenze 1-60 Hz. Per ogni livello, sono stati registrati 3 periodi del segnale utilizzando dieci realizzazioni di input per ciascun livello. Il numero di punti per periodo è 16384. È possibile notare che solo gli ultimi due periodi di ogni realizzazione sono in stadio stazionario. I dati sono stati campionati ad una frequenza di 200 Hz. Poiché sono disponibili numerose realizzazioni, il numero di livelli di eccitazioni testati sono ridotti a tre, cioè 12.2, 49.0 e 97.1 N (in RMS). Questi tre livelli comportano oscillazioni non lineari. Nello studio in esame, viene suggerito, per ogni livello, di utilizzare nove realizzazioni per la stima e considerare solo l'ultima realizzazione come test per il modello. D'ora in poi, per rendere l'elaborato più leggibile si utilizzerà il termine "SpecialOdd" per identificare questo set di dati.

3.2. Analisi Dataset

Per l'analisi e la sperimentazione dell'algoritmo di regressione sono stati presi in considerazione tutti i dataset disponibili rappresentanti i dati ottenuti dai tre tipi di ingresso descritti in precedenza. Ogni singolo dataset è stato studiato ed analizzato, facendo diverse elaborazioni dei dati, per renderlo compatibile con l'algoritmo di regressione basato sul polinomio di Bernstein.

Vista la composizione diversa dei dataset, l'elaborazione dei dati di ogni singolo tipo di ingresso sarà diversa ma tutte presenteranno delle parti in comune:

- Selezione dei parametri per l'analisi (scelta del sensore, suddivisione dei dati, overlap e numero di componenti)
- Acquisizione dei dati dal dataset. I dati vengono salvati in quattro matrici che rappresentano: la matrice dei dati di ingresso per il training; la matrice delle uscite per il training; le matrici di ingresso e uscita per il test
- Elaborazione dei dati
- Calcolo delle componenti dei segnali rispetto alla matrice di covarianza. In seguito ad un'analisi degli autovalori è stata applicata una PCA riducendo così il numero di componenti di ingresso.
- Normalizzazione dei dati di ingresso ed applicazione dell'algoritmo basato sui polinomi di Bernstein.

Il codice che segue mostra la parte in cui vengono selezionati i parametri utili per l'elaborazione e le importazioni delle matrici contenente i dati per il training e per il test

```
num_columns=200;

overlap=190;

features=20;

acc_sens=1;

for i=2:7

data_path=strcat("./F16_data/F16Data_",dataset,"_Level",int2str(i),".mat");
load(data_path);
if mod(i,2)==0
    X_row_train_orig=cat(2,X_row_train_orig,Force);
    Y_row_train_orig=cat(2,Y_row_train_orig,Acceleration(acc_sens,:));
else
    X_row_test_orig=cat(2,X_row_test_orig,Force);
    Y_row_test_orig=cat(2,Y_row_test_orig,Acceleration(acc_sens,:));
end
end
```

Per applicare la regressione del sistema in studio occorre suddividere i segnali di ingresso e di uscita in un certo numero di frame in quanto, per operare efficientemente, l'algoritmo necessita di un elevato numero di osservazioni. Per questo motivo sono state introdotte le variabili *num_columns* e *overlap*: la variabile *num_columns* determina il numero di campioni che vengono utilizzati per ogni frame mentre *overlap* sono i campioni che si "sovrappongono" tra un frame e il successivo. Il ciclo riportato nel codice consente di acquisire i dati dai level e salvarli all'interno delle matrici. Questa parte di codice si riferisce al dataset "Sine-Sweep" ed è identica anche per il dataset "FullMSine". Per l'ultimo dataset, l'importazione avviene allo stesso modo ma vengono importati prima tutti i dati relativi al training e poi quelli relativi al test. Il parametro *features* permette di selezionare il numero di componenti rispetto agli autovalori, come analizzato nel paragrafo 2.3.1. Il numero delle componenti influenza la qualità dell'adattamento del segnale (MSE): infatti, un aumento di questo numero impone una riduzione del grado del polinomio di Bernstein il quale deve essere il più alto possibile per avere più punti per la rappresentazione. Nel codice è stata inserita anche la variabile *acc_sens* che permette di selezionare l'uscita che ci interessa senza dover importare anche i dati dell'altre uscite.

Invece il codice riportato di seguito rappresenta l'approccio descritto nel paragrafo 2.3.1. Come è stato spiegato precedentemente, 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 autovalori delle rispettive matrici di covarianza. Dopo la scomposizione e all'analisi degli autovalori della matrice di covarianza degli ingressi, viene effettuata una PCA con il fine di ridurre il numero di componenti dei segnali di ingresso.

```
Ut=cat(1,X_train,X_test);  
  
Yt=cat(1,Y_train,Y_test);  
  
Ruu=cov(Ut);  
  
Ryy=cov(Yt);  
  
[psi, lambda]=eig(Ruu);  
  
[fi, gamma]=eig(Ryy);  
  
for i=1:size(Ut,1)  
    X(i,:)=psi\ (Ut(i,:))';  
    K(i,:)=fi\ (Yt(i,:))';  
end  
  
X=X(:,size(X,2)-features+1:end);
```

3.2.1 Analisi del dataset Sine-sweep with a linear, negative rate

I dati all'interno del dataset sono strutturati come sei coppie di matrici ingresso-uscita. Ogni singola coppia era inserita all'interno di un file proprio denominato F16Data_SineSwLevelN dove LevelN è il numero corrispondente a una coppia di matrici. Per ogni coppia, le matrici hanno dimensioni rispettivamente 1x108477 e 3x108477. L'unica eccezione è quella di Level4 il quale è composto da due matrici di dimensioni rispettivamente di 1x116736 e 3x116736. Le tre matrici di uscita corrispondono ai tre punti in cui sono

posizionati i tre sensori per la misura dell'accelerazione. L'identificazione del sistema non lineare è stata svolta considerando singolarmente i tre segnali di uscita. Sono stati usati i level2, level4 e level6 come dati per il training mentre i tre livelli rimanenti (3,5,7) sono stati usati per il test così come riportato nel paragrafo 3.1.1. I segnali di ingresso e di uscita sono stati suddivisi in un certo numero di frame così da poter applicare la regressione al sistema in esame. La lunghezza di ogni frame viene variata, mediante i parametri descritti in precedenza, durante la sperimentazione per ricercare la migliore rappresentazione in termini di prestazioni del sistema. Il codice seguente riproduce quanto descritto in questo paragrafo e in particolare viene mostrata la divisione in frame dei segnali.

```
j=1;
for i=1:(num_columns-overlap):size(X_row_train_orig,2)
    try
        X_multiple_rows(j,:)=X_row_train_orig(i:(i+num_columns-1));
        Y_multiple_rows(j,:)=Y_row_train_orig(i:(i+num_columns-1));
        j=j+1;
    catch error
        break
    end
end

X_train=X_multiple_rows;
Y_train=Y_multiple_rows;
num_rows_train=size(X_train,1);

X_multiple_rows=[];
Y_multiple_rows=[];

for i=0:size(X_row_test_orig,2)
    try

X_multiple_rows(i+1,:)=X_row_test_orig(1,((i.*num_columns)+1):(i+1).*num_columns);

Y_multiple_rows(i+1,:)=Y_row_test_orig(1,((i.*num_columns)+1):(i+1).*num_columns);
        catch error
            break
        end
    end

X_test=X_multiple_rows;

Y_test=Y_multiple_rows;
```

3.2.2 Analisi del dataset Multisine with a full frequency grid

In questo dataset i dati sono strutturati in maniera molto simile a quelli del dataset precedente. Quindi i dati sono strutturati come sei coppie di matrici di ingresso-uscita denominate F16Data_FullMSine_LevelN dove N rappresenta il numero della coppia. La dimensione della matrici di ingresso per ogni coppia è 1×73728 mentre quelle di uscita 3×73728 . Le tre matrici righe dell'uscita rappresentano i tre tipi di sensore ed anche in questo caso la trattazione è stata svolta considerando i tre segnali di uscita singolarmente. L'analisi applicata per questo dataset, come detto in precedenza, è del tutto analoga a quella svolta in precedenza per il Sine-Sweep. Quindi sono stati presi i level 2,4,6 come livelli per il training e i loro corrispondenti livelli 3,5,7 come livelli di test dell'algoritmo. Anche in questo caso i segnali di ingresso e uscita sono stati suddivisi in frame visto che, per operare efficientemente, l'algoritmo necessita di un elevato numero di osservazioni. Il codice che rappresenta le operazioni descritte è lo stesso utilizzato per il dataset "Sine-Sweep".

3.2.3 Analisi del dataset Multisine with a full frequency grid

Per il dataset "Multisine with a random frequency grid" è stata richiesta un'analisi differente rispetto agli altri due dataset riportati in precedenza. In questo caso la composizione delle matrici destinate al training è diversa da quella per il test. Facendo riferimento al paragrafo 3.1.3, si ha che ogni livello di training è composto da coppie di matrici ingresso-uscita di dimensione pari a 9×49152 e $3 \times 9 \times 49152$, visto che per ogni livello sono state registrate nove realizzazioni. Per il testing è stata fornita solo una realizzazione, di conseguenza le matrici avranno dimensione pari a 1×49152 per quella di ingresso e $3 \times 1 \times 49152$ per quella di uscita. Le coppie di matrici dei level 1,2,3 sono stati usati per il training dei dati. Vista la dimensione della matrice di ingresso, come primo procedimento è stata creata una nuova matrice di dimensione 1×1327104 contenente tutti i dati per il training di ingresso. Dopo questo primo procedimento, l'analisi continua allo stesso identico modo degli altri due dataset utilizzando i dati contenuti nei level 4,5,6 per il test.

La seguente tabella fornisce un riassunto sulla composizione dei dataset, descritta nei paragrafi precedenti.

DATASET	Numero level considerati (level training/level test)	Numero periodi per ogni level	Punti per periodo	Punti per il training	Punti per il test
Sine-Sweep	6 (3/3)	1	108477 116736 (level4)	333690	325431
FullMSine	6 (3/3)	9	8196	221292	221292
SpecialOdd	6 (3/3)	10	16384	1327104	49152

Tabella 3.1: Composizione dei dataset con il dettaglio dei punti

3.3. Risultati Sperimentazione

In questo paragrafo sono stati riportati i risultati ottenuti dalla sperimentazione dei dataset. Per ogni dataset sono stati riportati i migliori risultati per tutti i segnali di uscita a disposizione.

Nella tabella 3.2 sono riassunti i valori dei parametri dell'algoritmo per ottenere il risultato migliore possibile in termini di efficienza e di tempo di esecuzione dell'operazione.

Per verificare la qualità dei risultati derivanti dai test effettuati variando i parametri, sono stati misurati anche i valori di MSE (errore quadratico medio), RMSE e il tempo di esecuzione dell'algoritmo. Questi valori sono stati riportati all'interno della tabella 3.3.

Occorre porre l'attenzione sul tempo di esecuzione dell'algoritmo il quale dipende fortemente anche dall'hardware utilizzato per la sperimentazione.

	Frame	Suddivisione dei dati	Overlap	Componenti	Grado del polinomio di Bernstein
Sine-Sweep	5559	200	140	14	73
Multisine with full frequency grid	11054	120	100	20	52
Multisine with a random frequency grid	11057	200	120	14	73

Tabella 3.2: valori dei parametri utilizzati per la sperimentazione

	Segnale di uscita	MSE	RMSE	Tempo test (s)
Sine-Sweep	1	0.06799	0.260383	138.59
	2	0.186809	0.432214	87.05
	3	0.234781	0.484542	97.27
FullMSine	1	0.156572	0.395691	220.69
	2	0.354642	0.595518	205.031
	3	0.405038	0.6364426	223.29
SpecialOdd	1	0.234710	0.4844888	88.147
	2	0.197791	0.444737	75.469
	3	0.157110	0.296710	76.286

Tabella 3.3: valori ottenuti della sperimentazione

3.3.1. Risultati del dataset Sine-sweep with a linear, negative rate

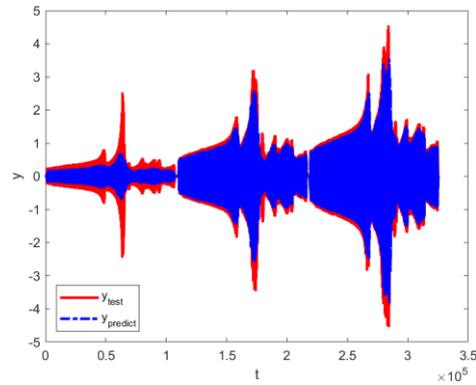


Figura 3.3: Rappresentazione dei risultati del dataset Sine-Sweep per il segnale misurato dal sensore posizionato sullo shaker

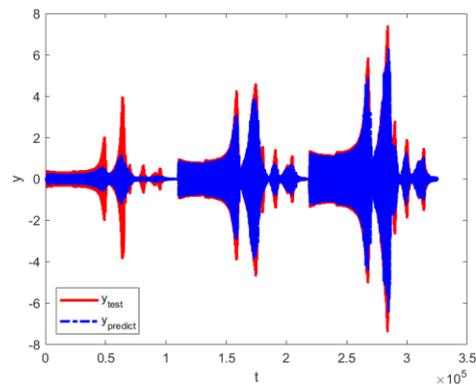


Figura 3.4: Rappresentazione dei risultati del dataset Sine-Sweep per il segnale misurato dal sensore sull'ala destra vicino all'interfaccia non lineare

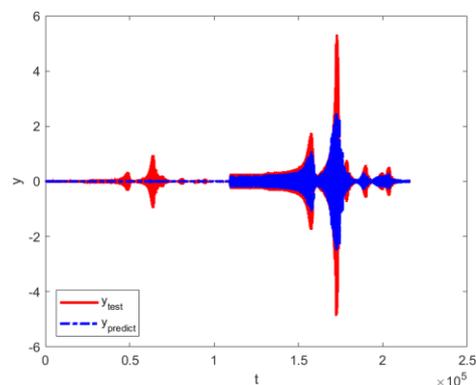


Figura 3.5: Rappresentazione dei risultati del dataset Sine-Sweep per il segnale misurato dal sensore posizionato sul carico fittizio, vicino all'interfaccia

3.3.2 Risultati del dataset Multisine with a full frequency grid

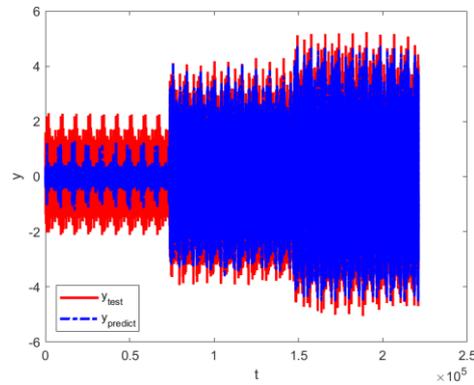


Figura 3.6: Rappresentazione dei risultati del dataset FullMSine per il segnale misurato dal sensore posizionato sullo shaker

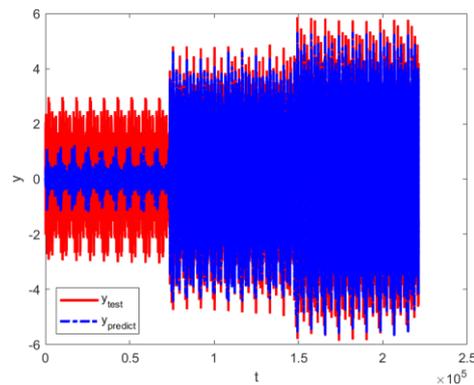


Figura 3.7: Rappresentazione dei risultati del dataset FullMSine per il segnale misurato dal sensore sull'ala destra vicino all'interfaccia non lineare

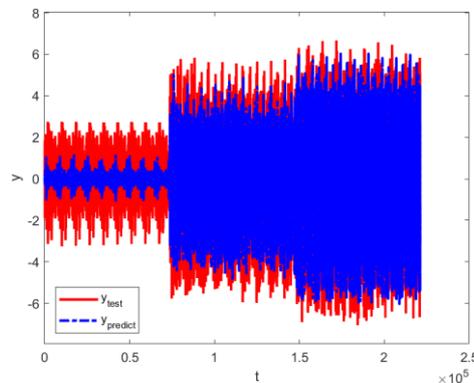


Figura 3.8: Rappresentazione dei risultati del dataset FullMSine per il segnale misurato dal sensore posizionato sul carico fittizio, vicino all'interfaccia

3.3.3 Analisi del dataset Multisine with random frequency grid

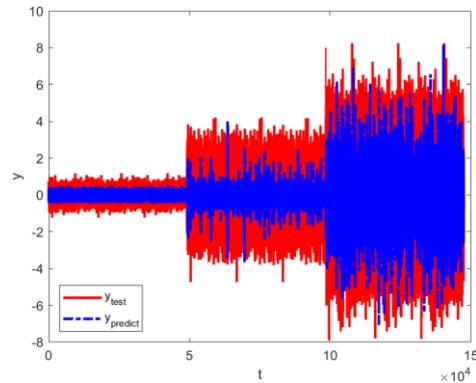


Figura 3.9: Rappresentazione dei risultati del dataset SpecialOdd per il segnale misurato dal sensore posizionato sullo shaker

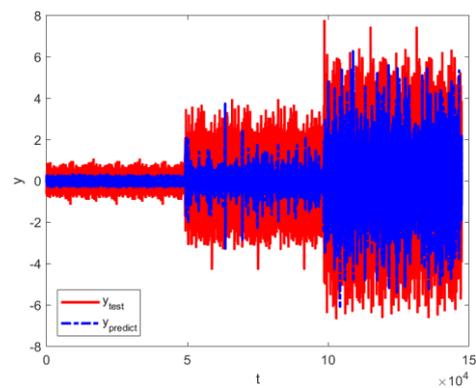


Figura 3.10: Rappresentazione dei risultati dataset SpecialOdd per il segnale misurato dal sensore sull'ala destra vicino all'interfaccia non lineare

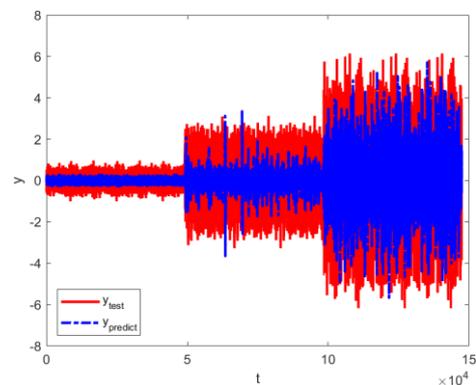


Figura 3.11: Rappresentazione dei risultati del dataset SpecialOdd per il segnale misurato dal sensore posizionato sul carico fittizio, vicino all'interfaccia

3.4. Osservazione dei risultati

Osservando attentamente i risultati ottenuti dalla sperimentazione dei dati, è stato notato un disadattamento tra il segnale predetto, cioè quello derivante dall'algoritmo di regressione del polinomio di Bernstein, e il segnale di test che deriva direttamente dai dati del dataset (figura 3.12). Questo disadattamento si verifica solo nel dataset "Sine-sweep" mentre negli altri due dataset questo inconveniente non si verifica. È possibile vedere il confronto tra i dataset nelle figure 3.12 e 3.13.

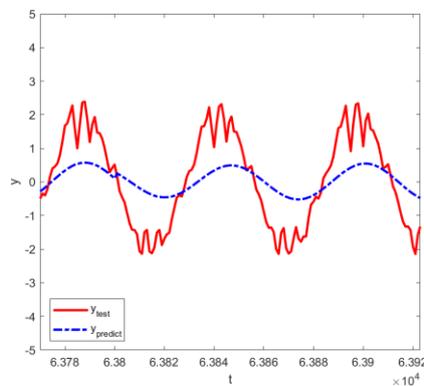


Figura 3.12: Particolare del disadattamento del segnale predetto con il segnale di test atteso nel dataset "Sine-Sweep"

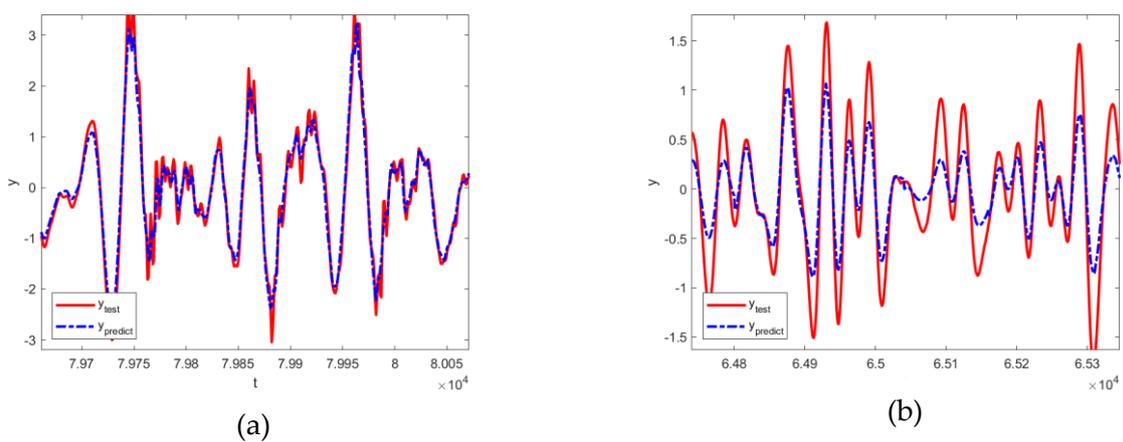


Figura 3.13: Particolare dei segnali derivati dalla sperimentazione dei dataset (a) FullMSine e (b) SpecialOdd

Per questo motivo si è deciso di analizzare più nel dettaglio tutto il dataset “Sine-sweep”. Con quest’ultima analisi si è potuto riscontrare che l’andamento non desiderato, si verifica solo nella prima parte del segnale. Infatti, analizzando tutto lo spettro, si osserva che il segnale predetto rispecchia quasi perfettamente il segnale di test nei punti restanti (figura 3.14).

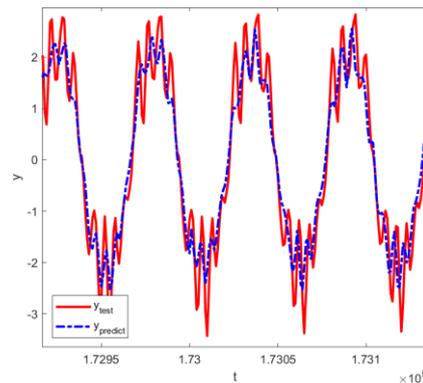


Figura 3.14: Particolare dei segnale in un altro punto derivante dal dataset “Sine-Sweep”

Approfondendo ancora di più l’analisi del dataset, si evidenzia che il problema del disadattamento è dovuto solo ai dati presenti all’interno delle matrici che compongono il level2 del dataset, come mostrato nella figura 3.15. Prendendo in considerazione solo questa parte del dataset e applicando di nuovo l’algoritmo di regressione del polinomio di Bernstein con nuovi parametri più stringenti, si è riuscito a limitare il problema solo in un minima parte del segnale (circa 2000 punti) senza però eliminarlo del tutto. Ipotizzando i vari motivi per i quali si verifica questo problema ci potrebbe essere la mancanza di un numero sufficiente di dati utili all’algoritmo per poter rappresentare al meglio il segnale ottenuto. Questo particolare andamento del segnale potrebbe derivare anche dalla componente di rumore presente durante i test della struttura e che sembrerebbe essere più rilevante all’interno del primo dataset.

Nonostante questo disadattamento, i risultati ottenuti dalla sperimentazione del dataset “Sine-Sweep” possono essere ritenuti accettabili in termini di accuratezza del fitting del segnale, se vengono paragonati con i risultati ottenuti dagli altri due dataset (tabella 3.3).

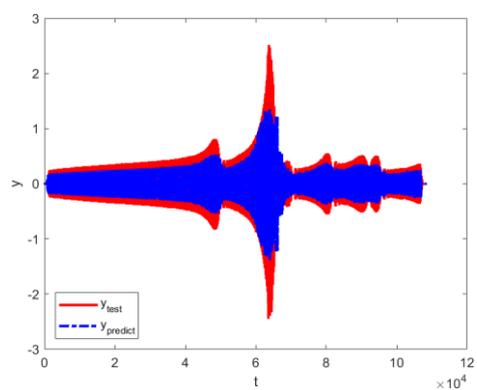


Figura 3.15: Segnale ottenuto solo dai punti del level2 del dataset
“Sine-Sweep”

CAPITOLO 4

CONCLUSIONI

Uno dei principali obiettivi nell'identificazione di sistemi dinamici non lineari è la selezione del modello che rappresenta adeguatamente il sistema e che richiede un numero ridotto di parametri. Questo elaborato ha mostrato come l'approccio presentato sia un'ottima tecnica per l'identificazione di un sistema non lineare, in quanto non richiede un modello adatto che rappresenti i segnali ingresso/uscita.

Tuttavia, per sfruttare adeguatamente questa tecnica, prima è necessario fare un'indagine approfondita del dataset, analizzando i dati a disposizione e renderli compatibili con l'approccio proposto. L'analisi iniziale è anche utile per comprendere il perché a volte i risultati non rispecchiano quelli attesi.

Bibliografia

- [1] 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.
- [2] 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."
- [3] Nonlinear System Identification.
https://en.wikipedia.org/wiki/Nonlinear_system_identification
- [4] Serie di Volterra. https://it.wikipedia.org/wiki/Serie_di_Volterra
- [5] Rete neurale artificiale.
https://it.wikipedia.org/wiki/Rete_neurale_artificiale
- [6] Sistemi strutturati a blocchi.
https://etd.adm.unipi.it/theses/available/etd-04052005-114205/unrestricted/003_CAP1.PDF
- [7] Polinomio Bernstein.
https://it.wikipedia.org/wiki/Polinomio_di_Bernstein
- [8] G. Biagetti, D. Coccia, P. Crippa, L. Falaschetti, C. Turchetti. "An acquisition system of in-house parameters from wireless sensors for the identification of an environmental model". 2018. 22nd International Conference on Knowledge-Based and Intellisent Information & Engineering System, pag 1906-1907.

Figure

2.1	Modello a blocchi di Hammerstein	13
2.2	Modello a blocchi di Wiener	13
2.3	Modello a blocchi di Hammerstein-Wiener	13
2.4	Modello strutturale di una rete neurale artificiale	15
2.5(a)	Rappresentazione della corrispondenza tra u, y, x e k	19
2.5(b)	Rappresentazione schematica della funzione (2.11)	19
2.6	Polinomi di particle-Bernstein	22
3.1	Struttura completa di un F16	26
3.2(a)	Carico fittizio montato sulla punta dell'ala	26
3.2(b)	Shaker fissato sotto l'ala destra	26
3.2(c)	Collegamento posteriore dell'interfaccia di montaggio, a destra del carico	27
3.3	Rappresentazione dei risultati del dataset Sine-Sweep per il segnale misurato dal sensore posizionato sullo shaker	37
3.4	Rappresentazione dei risultati del dataset Sine-Sweep per il segnale misurato dal sensore sull'ala destra vicino all'interfaccia non lineare	37
3.5	Rappresentazione dei risultati del dataset Sine-Sweep per il segnale misurato dal sensore posizionato sul carico fittizio, vicino all'interfaccia	37
3.6	Rappresentazione dei risultati del dataset FullMSine per il segnale misurato dal sensore posizionato sullo shaker	38
3.7	Rappresentazione dei risultati del dataset FullMSine per il segnale misurato dal sensore sull'ala destra vicino all'interfaccia non lineare	38

3.8	Rappresentazione dei risultati del dataset FullMSine per il segnale misurato dal sensore posizionato sul carico fittizio, vicino all'interfaccia ..	38
3.9	Rappresentazione dei risultati del dataset SpecialOdd per il segnale misurato da sensore posizionato sullo shaker ..	39
3.10	Rappresentazione dei risultati dataset SpecialOdd per il segnale misurato dal sensore sull'ala destra vicino all'interfaccia non lineare ..	39
3.11	Rappresentazione dei risultati del dataset SpecialOdd per il segnale misurato dal sensore posizionato sul carico fittizio, vicino all'interfaccia ..	39
3.12	Particolare del disadattamento del segnale predetto con il segnale di test atteso nel dataset "Sine-Sweep" ..	40
3.13(a)	Particolare del segnale derivato dalla sperimentazione dei dataset FullMSine ..	40
3.13(b)	Particolare del segnale derivato dalla sperimentazione dei dataset Multisine with a random frequency grid ..	40
3.14	Particolare dei segnale in un altro punto derivante dal dataset "Sine-sweep" ..	41
3.15	Segnale ottenuto solo dai punti del level2 del dataset "Sine-Sweep"	42