

UNIVERSITÀ DEGLI STUDI DI CAGLIARI

---

FACOLTA' DI INGEGNERIA  
Corso di Laurea in Ingegneria Elettronica

**IDENTIFICAZIONE  
DEI PARAMETRI ELETTROMAGNETICI  
DI UN ORDIGNO INESPLOSO SEPOLTO**

Tesi di Laurea Triennale

**Relatore:**  
**Prof. Giuseppe Rodriguez**

**Candidato:**  
**Alessio Pani**

**Correlatore:**  
**Prof. Luisa Fermo**

**Anno Accademico 2014/2015**



Ai miei genitori e ai miei amici per il supporto datomi  
in questi anni e ai miei fantastici colleghi per aver  
condiviso gioie e dolori di questo lungo percorso.



# Indice

<b>Introduzione</b>	<b>6</b>
<b>1 Sistema fisico</b>	<b>7</b>
1.1 Teoria del rilevamento degli UXO . . . . .	7
1.2 Sfera conduttiva . . . . .	8
1.3 Fattore adimensionale . . . . .	10
<b>2 Risoluzione numerica di equazioni non lineari</b>	<b>12</b>
2.1 Equazioni non lineari . . . . .	12
2.2 Risoluzione tramite fzero . . . . .	13
2.2.1 Risultati con fzero . . . . .	13
2.3 Metodo di Newton . . . . .	15
2.3.1 Risultati con Newton . . . . .	16
2.4 Metodo Newton-Bisezione . . . . .	18
2.4.1 Risultati con Newton-Bisezione . . . . .	19
2.5 Confronto Newton-Bisezione . . . . .	20
<b>3 Metodo di Prony</b>	<b>22</b>
3.1 L'algoritmo . . . . .	22
3.1.1 Primo step . . . . .	23
3.1.2 Secondo step . . . . .	24
3.2 Esempio . . . . .	24
3.3 Confronto con la trasformata di Fourier . . . . .	32
<b>4 Analisi dei dati</b>	<b>33</b>
4.1 Tempo reale . . . . .	34
4.1.1 Esempio 1 . . . . .	35
4.1.2 Esempio 2 . . . . .	35
<b>Conclusioni</b>	<b>37</b>

*INDICE*

5

**Bibliografia**

**38**

# Introduzione

I metodi di Induzione Elettromagnetica sono efficaci nel definire i parametri caratteristici di un ordigno inesplosivo (UXO in inglese).

È noto però che la risposta degli UXO (così come di altri conduttori confinati) può essere parametrizzata nel dominio del tempo come una serie di curve esponenziali negative con i loro rispettivi coefficienti, o, nel dominio di Laplace, in una sommatoria di poli reali del primo ordine e i loro relativi residui.

Poiché i poli e i residui sono funzioni dei parametri caratteristici dell'ordigno (dimensioni, conduttività e permeabilità magnetica relativa), avendo un certo numero di coppie poli/residui è possibile ottenere tali parametri tramite metodi numerici. In primo luogo si affronta il problema diretto del calcolo dei dati a partire dai parametri caratteristici dell'ordigno, in primis nel calcolo delle radici di una equazione non lineare e poi nel calcolo della risposta impulsiva dell'obiettivo.

Successivamente si affronta il problema inverso: a partire dalla risposta impulsiva si vuole conoscere i parametri caratteristici dell'ordigno. Per fare questo la si analizza tramite il metodo di Prony che permette di ottenere una serie di esponenziali con i relativi coefficienti.

Dapprima verrà proposto un esempio esplicativo sul metodo di Prony nel quale verranno descritte le problematiche e i punti critici per una soddisfacente applicazione del metodo (come l'analisi del rango della matrice di Hankel in relazione al numero dei dati in ingresso). Successivamente verrà applicato alla risposta impulsiva nel caso dimensionale (cioè con un tempo reale), calcolata tramite l'inserimento dei parametri caratteristici di un ordigno in modo da ricostruire uno scenario verosimile per verificare che la serie dei coefficienti e degli esponenziale trovati dal metodo in questione effettivamente approssimano la funziona in modo soddisfacente.

I risultati ottenuti potranno in seguito essere utilizzati all'interno di un algoritmo di fitting (ad esempio il metodo dei minimi quadrati) in modo tale da riottenere i parametri caratteristici dell'ordigno inseriti in partenza.

# Capitolo 1

## Sistema fisico

### 1.1 Teoria del rilevamento degli UXO

La teoria del rilevamento degli UXO e la loro caratterizzazione è semplificata tramite alcune assunzioni che sono, per la maggior parte, pienamente giustificate dalla natura dell'oggetto [2]. Le assunzioni sono:

1. Gli UXO hanno una conduttività elettrica che è almeno 6 ordini di grandezza più grande del mezzo in cui si trovano. Per una buona approssimazione quindi noi possiamo assumere che il mezzo è non conduttivo ( $\sigma = 0$ );
2. Gli UXO sono genericamente prodotti dall'acciaio avente una permeabilità magnetica relativa di circa 200. Tipicamente il mezzo ha una bassa permeabilità ( $\sim 1$ ). Possiamo quindi assumere che la permeabilità del mezzo è  $\mu_0$ ;
3. Approssimazione "Quasi Statica": alle frequenze impiegate, l'UXO è elettricamente piccolo. In pratica le sue dimensioni sono piccole rispetto alla lunghezza d'onda  $\lambda_0$  del campo elettromagnetico primario;
4. L'UXO ha una geometria tri-dimensionale con dimensioni caratteristiche che sono piccole rispetto alla distanza con il sensore che genera il campo magnetico primario. Possiamo inoltre assumere che tale campo sia uniforme su tutto il volume dell'obiettivo e che il campo magnetico secondario generato dall'UXO sia approssimato dal comportamento di un dipolo magnetico con il relativo momento magnetico.

Nel caso in cui un corpo rispetti le prime tre assunzioni, tale corpo viene definito **conduttore confinato** [2].

Inoltre si può assumere che lo spettro del campo magnetico creato dalle correnti

indotte presenti nel corpo sopra definito possa essere rappresentato come una sommatoria di residui e poli semplici dislocati lungo la parte negativa dell'asse reale [5] e che, nel caso del dominio del tempo prende la forma:

$$h(t) = \sum_{m=1}^{\infty} B_m e^{S_m t} \quad (1.1)$$

mentre, nel dominio di Laplace la formula prende la forma:

$$H(s) = \sum_{m=1}^{\infty} \frac{B_m}{s - S_m} \quad (1.2)$$

dove:

- $h(t)$  : intensità del campo magnetico secondario
- $H(s)$  : trasformata di Laplace dell'intensità del campo magnetico secondario
- $t$  : tempo
- $s$  : variabile di Laplace
- $S_m = 1/\tau_m$  : m-esimo polo del campo magnetico secondario, con  $S_m$  : numero negativo
- $B_m$  : residuo del polo m-esimo

I valori di  $S_m$  e di  $B_m$  possono essere trovati analiticamente solo per alcune forme geometriche caratterizzati da un sistema di coordinate curvilinee ortogonali (come le sfere). Alex Kaufman [4] ad esempio ha sviluppato delle tabelle complete con le formule per le costanti di tempo principali (le più lunghe) di varie forme canoniche. In realtà queste formule ci fanno capire il comportamento a regime, ossia alla fine del transitorio del campo magnetico secondario e non il comportamento completo ed è per questo motivo che non possono essere utilizzate in questo lavoro.

## 1.2 Sfera conduttiva

La forma canonica che prendiamo in considerazione per analizzare il Campo secondario è quello della sfera. Già nel 1969 i ricercatori Wait e Spies svilupparono un modello della risposta nel dominio del tempo da parte di una sfera conduttiva [3]. Il campo secondario della sfera si comporta quindi come un dipolo puntiforme situato al centro della sfera.

Il comportamento matematico del dipolo quindi è

$$\vec{H}_s = \frac{1}{4\pi r^5} [3(\vec{r}\vec{r} - r^2 I) \cdot (\vec{M} \cdot \vec{H}_0)] \quad (1.3)$$

dove:

- $\vec{H}_0, \vec{H}_s$  : rispettivamente campo magnetico primario e secondario
- $\vec{M} = M \cdot \vec{H}_0$  : momento magnetico indotto
- $M$  : polarizzabilità magnetica
- $I$  : matrice identità
- $r = |\vec{r}|$  : vettore posizione diretto dal dipolo verso il punto campo  $P$
- $\vec{r}\vec{r} = \vec{r} \otimes \vec{r}$ , cioè il prodotto tensoriale<sup>1</sup>

Il sistema fisico può essere quindi rappresentato come in Figura 1.1.

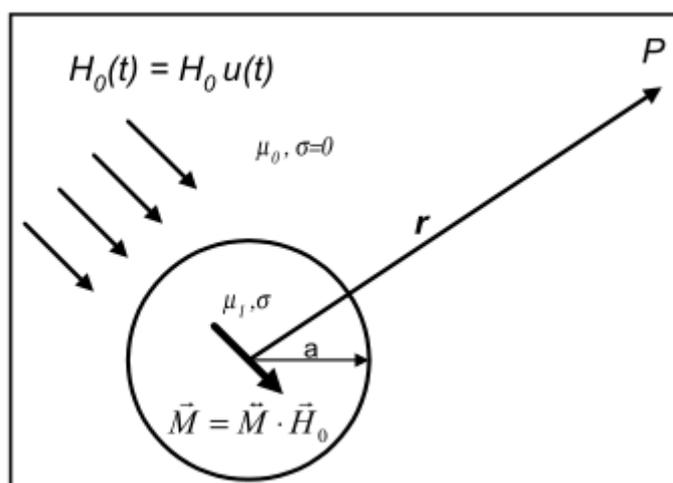


Figura 1.1: Sistema fisico oggetto di analisi

Abbiamo in pratica un classico sistema formato da due bobine, la prima che genera il campo magnetico primario  $H_0$  e il secondo che si occupa di ricevere (e quindi

<sup>1</sup>Per prodotto tensoriale (o diade) tra due vettori espressi in componenti cartesiane si intende una applicazione lineare che trasforma due vettori in una matrice [1]. Tale prodotto in realtà è un caso speciale del più generico prodotto di Kronecker (che tratta anche matrici) in cui coincide con il prodotto del primo vettore trasposto e il secondo espresso come vettore riga. In MATLAB il comando per quanto riguarda il prodotto tensoriale tra due vettori è `'*`, mentre per il prodotto di Kronecker si utilizza il comando `A=kron(M,N)`.

analizzare) la risposta  $H_s$  dell'oggetto sepolto. Si può notare dall'equazione (1.3) che il comportamento del campo secondario è completamente contenuto all'interno del termine più a destra dell'equazione, cioè quello relativo al momento magnetico indotto. La prima parte invece è un termine puramente geometrico che contiene sia la componente radiale del momento ( $\vec{r}\vec{r}$ ) sia il momento completo di dipolo essendo presente una matrice identità.

Poichè a noi interessa il comportamento, l'interesse è sul momento di dipolo magnetico ( $\vec{H}_0$  è noto) ed il suo comportamento nel transitorio può essere espresso dall'equazione (1.4).

$$h(t) = \left[ 6\mu_R \sum_{n=1}^{\infty} \frac{e^{-\frac{\delta_n^2 T}{\mu_R}}}{(\mu_R + 2)(\mu_R - 1) + \delta_n^2} - \frac{2(\mu_R - 1)}{\mu_R + 2} \right] u(T) \quad (1.4)$$

con

- $h(t) = m(t)/2\pi a^3 H_0$  : momento magnetico nel dominio del tempo normalizzato
- $\mu_R$  : permeabilità magnetica relativa ( $\mu_1 = \mu_R \mu_0$ )
- $T = t/\mu_0 \sigma a^2$  : tempo adimensionale
- $\delta_n$  : soluzione di  $\tan(\delta_n) = \frac{(\mu_R - 1)\delta_n}{\mu_R - 1 + \delta_n^2}$
- $u(T)$  : funzione di Heaviside, per  $t \geq 0$

La maggior parte dei sistemi che analizzano la risposta ad un campo elettromagnetico però misura la risposta impulsiva (cioè il comportamento nel caso in cui il sistema in oggetto sia sollecitato da un impulso come ingresso  $H_0$ ), che equivale alla derivata di (1.4) rispetto al tempo  $t$  cambiata di segno, cioè

$$-\frac{\delta h(t)}{\delta t} = \frac{-1}{\mu_0 a^2 \sigma} \frac{\delta h(t)}{\delta T} = \frac{1}{\mu_0 a^2 \sigma} \left[ 6 \sum_{n=1}^{\infty} \frac{\delta_n^2 e^{-\frac{\delta_n^2 T}{\mu_R}}}{(\mu_R + 2)(\mu_R - 1) + \delta_n^2} - \delta(T) \right] \quad (1.5)$$

Molti dispositivi come il NanoTEM della Zonge rappresenta in uscita proprio il grafico relativo alla (1.5).

### 1.3 Fattore adimensionale

Analizzando le ultime due equazioni del precedente paragrafo si può notare che i residui  $B_m$  e i poli  $S_m$  sono, nell'ordine

- $B_n = \frac{6\mu_R}{(\mu_R+2)(\mu_R-1)+\delta_n^2}$
- $B_0 = -\frac{2(\mu_R-1)}{\mu_R+2}$
- $S_n = -\frac{\delta_n^2 T}{\mu_R}$

per quanto riguarda il momento magnetico, mentre per la sua risposta impulsiva abbiamo

- $B_n = \frac{6}{\mu_0 a^2 \sigma} \frac{\delta_n^2}{(\mu_R+2)(\mu_R-1)+\delta_n^2}$
- $S_n = -\frac{\delta_n^2 T}{\mu_R}$

Si può notare che in entrambe le ultime due equazioni del paragrafo precedente si parla di tempo adimensionale e, notando il fattore fuori dalle parentesi quadre della risposta impulsiva, anche il residuo ha lo stesso fattore di normalizzazione ( $\mu_0 a^2 \sigma$ ).

Da dove deriva questo termine? Il motivo è semplice: per ottenere le formule (1.4) e (1.5) tra i vari passaggi [3] è presente a fattore comune  $\alpha = \beta\sqrt{S_n}$ , con  $\beta = \sqrt{\sigma\mu_o}a$ . Poichè in un passaggio per determinare delle singolarità di un integrando  $G(s)$  a denominatore occorre trovare i valori di  $s$  tale che  $G(S_m) = 0$  e sapendo che ovviamente i poli erano negativi sull'asse reale, si è semplificato con  $\beta\sqrt{S_n} = j\delta_n$ . In definitiva, il valore di  $\delta_n$  ha al suo interno sia informazioni sui poli sia sui parametri caratteristici dell'ordigno, in relazione alla sua permeabilità magnetica relativa, oltre al fatto che è un valore che ci permette di calcolare la risposta impulsiva.

Risulta quindi importante risolvere l'equazione non lineare che ci permette di conoscere tali valori, ma diventa di importanza fondamentale per affrontare il problema diretto, cioè quello del calcolo dei dati a partire dai parametri caratteristici dell'oggetto.

Si può inoltre notare come nei calcoli sopracitati, si passi da un  $\beta = \sqrt{\sigma\mu_o}a$  ad un  $\beta = \sqrt{\sigma\mu_1}a$  mettendo in evidenza il valore di  $\mu_R$ , semplicemente perchè si è tenuto conto della permeabilità magnetica relativa dell'oggetto sepolto.

# Capitolo 2

## Risoluzione numerica di equazioni non lineari

Per calcolare (1.4) o (1.5) è indispensabile risolvere l'equazione

$$\tan(\delta_n) = \frac{(\mu_R - 1)\delta_n}{\mu_R - 1 + \delta_n^2} \quad (2.1)$$

dove ricordiamo che  $\mu_R$  è la permabilità magnetica relativa.

Dopo una breve spiegazione di cosa sono le equazioni non lineari, verranno presentati i metodi con cui è stata trovata la soluzione e le problematiche riscontrate, dovute sia alla natura della funzione stessa, sia al fatto che la risoluzione viene svolta tramite MATLAB e quindi sopraggiungono i classici problemi di Calcolo Numerico, come gli arrotondamenti o i troncamenti dei valori.

### 2.1 Equazioni non lineari

Una equazione non lineare è una equazione che non è esprimibile come combinazione lineare delle incognite e di una costante.

Di norma queste equazioni possono anche avere una forma complicata, del tipo

$$e^y + \sin(y) = \tan(y^2)$$

In generale però non è possibile risolvere analiticamente una equazione non lineare ed è dunque necessario ricorrere all'approssimazione numerica delle sue radici (o zeri), ossia quei valori  $\alpha$  tale che  $f(\alpha) = 0$ .

La forma generale di un'equazione non lineare è la seguente

$$f(x) = 0. \quad (2.2)$$

L'approssimazione numerica di una radice  $\alpha$  della funzione  $f$  si basa in genere sull'uso di metodi iterativi [8] i quali costruiscono una successione di valori  $x_k$  tali che

$$\lim_{k \rightarrow \infty} x_k = \alpha.$$

**Definizione 1.** Poichè una successione di  $n$  punti ha, al passo  $k < n$ , una differenza rispetto al valore  $\alpha$ , si definisce l'**errore al passo  $k$** , come

$$e_k = x_k - \alpha.$$

L'errore  $e_k$  è un valore molto importante, di solito deciso a priori, e che in base al metodo utilizzato può fornire una condizione di arresto del ciclo di iterazione.

**Definizione 2.** Un metodo iterativo si definisce **convergente** se

$$\lim_{k \rightarrow \infty} |e_k| = 0.$$

Molto importante è l'ordine di convergenza che sostanzialmente indica la velocità con cui la successione di valori ottenuta converge al valore esatto.

Una analisi un po' più approfondita sull'ordine di convergenza verrà sviluppata nel Paragrafo 2.5, dove si confrontano i due metodi numerici utilizzati per il calcolo delle radici in relazione al numero di iterazione e quindi all'ordine di convergenza.

## 2.2 Risoluzione tramite `fzero`

Il primo approccio per la risoluzione di (2.2) è quello di usare una routine predefinita di MATLAB sviluppata proprio per questo scopo: risolvere equazioni non lineari in una variabile.

La routine richiede in ingresso ovviamente la funzione da analizzare (di norma precedentemente definita tramite un puntatore a funzione `@nomevariabile`) e il punto iniziale  $x_0$  per calcolare la radice (o le radici). In pratica il valore restituito da `fzero`,  $x$ , è un punto vicino a dove la funzione cambia di segno.

Per svolgere tale calcoli `fzero` si affida ad un metodo che ibrida tre algoritmi per la ricerca delle soluzioni di equazioni non lineari, ed esse sono il metodo della Secante, il metodo di Bisezione ed il metodo dell'Interpolazione quadratica inversa.

### 2.2.1 Risultati con `fzero`

Come prima prova si è deciso di calcolare le radici con  $\mu_R = 5$  ed il numero di soluzioni da trovare  $n = 5$ . Inoltre attraverso uno semplice studio di funzione si è

capito che ciascuna soluzione  $\alpha_n$  (poichè la funzione è periodica ci saranno infinite soluzioni che si ripetono con periodicità, in base al valore di  $\mu_R$  in ingresso) è compresa nell'intervallo  $[\pi, n\pi + \frac{\pi}{2}]$ : le radici che mi verranno date da **fzero** dovranno quindi essere comprese tra questi due estremi.

Per  $n = 1$  ad esempio, avrei una radice compresa tra 3.14 e 4.7124, per  $n = 2$  invece la radice sarà presente tra gli estremi 6.28 e 7.8540 e così via.

Analizzando i valori ottenuti però notiamo un risultato un po' particolare.

4.7124  
7.8540  
10.9956  
14.1372  
17.2788

Tabella 2.1: Radici fino a  $n = 5$  con **fzero**

Come si può notare dalla Tabella 2.1 le radici corrispondono esattamente **all'estremo destro dell'intervallo al variare di n**.

Il motivo di ciò è da ricercare nei problemi di approssimazione, nell'errore di calcolo e nella natura stessa della funzione: in pratica **fzero** considera l'estremo destro di qualunque  $n$ -esimo intervallo (che è un punto di discontinuità in cui si passa da  $+\infty$  a  $-\infty$ ) come un valore della funzione e quindi lo considera come radice.

La soluzione consiste nello spostarsi di un valore piccolo, rispetto all'intervallo e a sinistra rispetto all'estremo destro, in modo tale da fargli considerare il valore corretto. Poichè il valore medio dell'intervallo è di  $n\pi + \frac{\pi}{4}$  si può togliere il valore  $\frac{\pi}{4}$  al punto  $x_0$  di partenza. In questo caso le soluzioni sono quelle mostrate dalla Tabella 2.2.

3.8289  
6.7804  
9.7983  
12.8612  
15.9500

Tabella 2.2: Radici fino a  $n = 5$  con **fzero** con  $x_0 - \frac{\pi}{4}$

I risultati stavolta sono corretti e sono all'interno dell'intervallo.

Ora rimane un solo aspetto da valutare: il numero di soluzioni non è abbastanza per il lavoro che deve essere svolto, quindi è meglio verificare se all'aumentare di  $\mu_R$  si mantengono le stesse prestazioni. La seguente simulazione verrà eseguita con tre valori di  $\mu_R$  (5, 50 e 100) e un numero di soluzioni di 10. I risultati sono mostrati nella Tabella 2.3.

$\mu_R = 5$	$\mu_R = 50$	$\mu_R = 100$
3.8289	4.4055	4.4489
6.7804	7.5749	1.5708
9.7983	10.6934	1.5708
12.8612	13.7969	13.9278
15.9500	16.8948	17.0519
19.0543	19.9909	20.1722
22.1683	23.0868	23.2906
25.2887	26.1835	26.4079
28.4135	29.2817	41.9903
31.5416	29.8541	29.8451

Tabella 2.3: Radici con  $n = 10$ 

Si può notare che i valori mostrati per valori di  $\mu_R$  elevati sono totalmente errati. Addirittura per  $\mu_R = 100$ , il secondo e il terzo valore sono uguali e sono inferiori al primo valore (e di conseguenza dell'intervallo in cui dovrebbero trovarsi).

Per mitigare questo problema si potrebbe cercare un valore da sottrarre al punto in ingresso su *fzero* per ogni valore di  $\mu_R$ , cercando nel frattempo di ottimizzare il numero di risultati trovati. In realtà il gioco non vale la candela, poichè anche per valori di  $\mu_R$  molto elevati ( $\cong 200$ ), il numero massimo di risultati è di circa 5, e lo scopo è di ottenerne almeno 10 volte tanto.

Si deve quindi abbandonare l'idea di usare *fzero* e di puntare su un altro algoritmo. La scelta ricade sul metodo di Newton.

## 2.3 Metodo di Newton

Sia  $f(x)$  una funzione continua in un intervallo  $[a, b]$  e ivi dotata di un solo zero. Fissato un punto iniziale  $x_0$  posto in prossimità della radice  $\alpha$ , consideriamo la retta tangente alla funzione data nel punto di ascissa  $x_0$ . Tale metodo (definito anche metodo delle tangenti), consiste nel considerare come approssimazione successiva il punto  $x_1$ , intersezione della retta con l'asse delle ascisse e nell'iterare il procedimento considerando la retta tangente a  $f(x)$  nel punto di ascissa  $x_1$ .

La formulazione matematica del metodo (più specificatamente la sua relazione di ricorrenza) è la seguente:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad (2.3)$$

che è ovviamente valida se e solo se la derivata della funzione nel punto sia diversa da 0 in tutti i punti generati dalla successione.

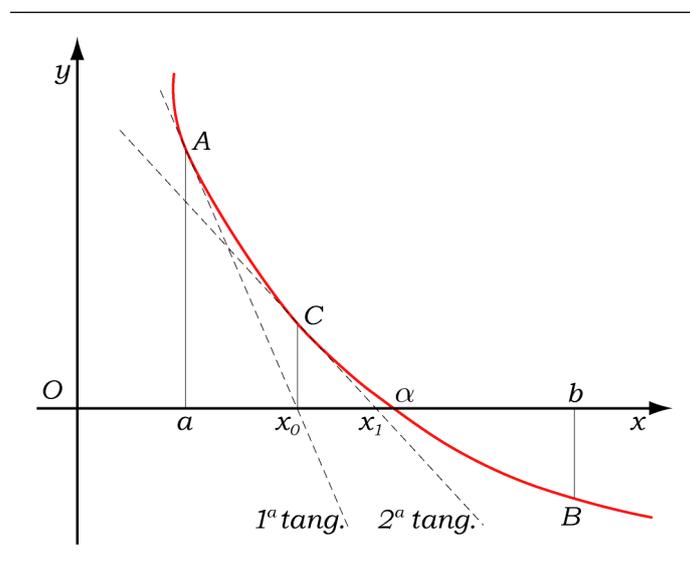


Figura 2.1: Spiegazione geometrica del metodo di Newton

Nelle prime definizioni all'inizio del capitolo ho parlato della *convergenza*. Per quanto riguarda questo metodo si dimostra che la convergenza è sempre verificata purchè il punto iniziale  $x_0$  venga scelto sufficientemente vicino alla radice  $\alpha$ , verificando il teorema della convergenza tramite l'analisi del rapporto tra l'errore al passo  $k+1$  e l'errore al passo  $k$ .

Da un semplice sguardo alla formula (2.3) si può altresì intuire quale sia la condizione per uscire dalla ricorsione: in caso di buona approssimazione della radice,  $\epsilon = x_{k+1} - x_k$  è un valore molto piccolo. Parlando in termini di MATLAB, quando l'errore  $\epsilon$  è dell'ordine di  $10^{-8}$  (valore arbitrario) o inferiore, l'algoritmo si arresta.

### 2.3.1 Risultati con Newton

Tramite una implementazione del Metodo di Newton all'interno di un ciclo per variare il valore di  $n$  (e quindi del punto) proviamo a trovare le radici (almeno 10 per iniziare), sempre con un valore di  $\mu_R=5$ . Tenendo però a mente il discorso sull'estremo destro dell'intervallo che è presente anche in questo caso, esso sarà ancora ridotto di un certo valore, piccolo a piacere.

Si parte da un valore molto piccolo per vedere il comportamento di Newton: l'estremo destro diventa così  $n\pi + \frac{\pi}{2} - \frac{\pi}{30}$ .

Le radici sono corrette, sono all'interno dell'intervallo, come si può notare dalla Tabella 2.4. Occorre ora testare all'aumentare di  $\mu_R$  se questo algoritmo mantiene

3.8289  
 6.7804  
 9.7983  
 12.8612  
 15.9500  
 19.0543  
 22.1683  
 25.2887  
 28.4135  
 31.5416

Tabella 2.4: Radici con Newton con  $n = 10$  e  $\mu_R = 5$

la sua efficacia. Come nel caso di *fzero* le prossime simulazioni saranno, sempre con  $n=10$ , per  $\mu_R=5, 50, 100$ .

$\mu_R = 5$	$\mu_R = 50$	$\mu_R = 100$
3.8289	4.4055	4.4489
6.7804	7.5749	7.6489
9.7983	10.6934	10.7966
12.8612	13.7969	13.9278
15.9500	16.8948	17.0519
19.0543	19.9909	20.1722
22.1683	23.0868	23.2906
25.2887	26.1835	26.4079
28.4135	29.2817	29.5245
31.5416	32.3816	32.6409

Tabella 2.5: Radici con Newton con  $n = 10$

I risultati sono corretti anche in questo caso (Tab. 2.5). In riferimento al testo [2] il massimo valore utilizzato da  $\mu_R$  è 200. In realtà il valore della permeabilità magnetica relativa di materiali ferromagnetici è dell'ordine di 2000 e oltre in base al campo magnetico applicato.

Il passo successivo è quindi verificare che Newton risolva correttamente l'equazione non lineare iniziale per valori più alti di  $\mu_R$ .

Posto quindi  $\mu_R = 200, 1000, 2000$  e con  $n = 10$ , i risultati sono mostrati nella Tabella 2.6.

Per i primi due valori di  $\mu_R$  Newton funziona decisamente bene, mentre per  $\mu_R = 2000$  gli ultimi tre valori sono **totalmente fuori dall'intervallo**. Ovviamente non conviene ripetere tutte le simulazioni sperando che un nuovo eventuale metodo

$\mu_R = 200$	$\mu_R = 1000$	$\mu_R = 2000$
4.4711	4.4889	4.4912
7.6868	7.7175	7.7214
10.8499	10.8932	10.8987
13.9963	14.0521	14.0592
17.1353	17.2036	17.2121
20.2703	20.3510	20.3611
23.4030	23.4960	23.5077
26.5342	26.6394	83.1986
29.6643	29.7818	230.7878
32.7939	32.9235	196.2466

Tabella 2.6: Radici con Newton con  $n = 10$ 

possa reggere anche a valori di  $\mu_R$  così alti, anche perchè in ogni caso fino ad un valore di circa 1000 Newton si comporta benissimo.

L'idea è quindi di **utilizzare un altro Metodo insieme a Newton** che possa gestire il caso in cui si esca fuori dall'intervallo in modo così brusco. La scelta ricade sul metodo di Bisezione. I motivi di questa scelta e una analisi più approfondita della convergenza e della sua velocità saranno spiegati nel Paragrafo 2.5.

## 2.4 Metodo Newton-Bisezione

Il metodo di bisezione è essenzialmente la versione matematica del metodo di ricerca binario usato in Informatica, normalmente su array di dati già ordinati in senso crescente. Il principio è molto semplice: si rifà al metodo di ricerca di una parola in un dizionario, in cui si parte da metà dizionario e poi in base all'iniziale della parola da cercare decidiamo se continuare la ricerca in una delle due metà. Successivamente si riapplica la suddivisione alla metà considerata fino a quando non si trova la parola.

Il metodo di bisezione è simile ma studia l'andamento del segno della funzione in modo da avvicinarsi sempre di più alla radice.

In pratica data una funzione  $f(x)$  continua nell'intervallo  $[a,b]$ , consideriamo il punto medio  $c = \frac{a+b}{2}$  dell'intervallo. Se  $f(c)=0$ , la radice  $\alpha$  in quell'intervallo è proprio  $c$ , altrimenti la radice cade in uno dei due semi intervalli  $[a,c]$  e  $[c,b]$ . Per capire in quale intervallo si trova la radice è sufficiente verificare in quale intervallo la funzione cambia di segno, ossia analizzare del prodotto  $f(a) \cdot f(c)$ : se è negativo la radice si trova nel primo intervallo, in caso contrario nel secondo. La successiva iterazione del metodo consisterà nel ripetere questo procedimento dopo

aver sostituito l'intervallo  $[a, b]$  con uno dei due precedenti semi intervalli, in base al segno.

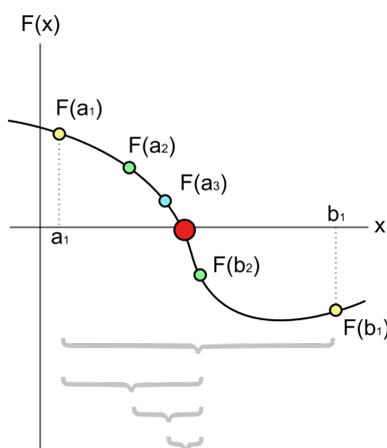


Figura 2.2: Metodo di bisezione

Se per Newton il criterio di arresto era quello relativo alla differenza tra due valori di  $x_k$  che doveva essere piccolo a piacere, per quanto riguarda invece il metodo di bisezione i criteri sono tre:

1. si annulla la funzione nel punto medio;
2. il semi intervallo  $[a_k, b_k]$  si rimpicciolisce al di sotto di una precisione assegnata;
3. si è raggiunto il numero massimo di iterazioni.

Ovviamente basta uno di questi tre criteri per uscire dalla iterazione. Inoltre, per ottimizzare i tempi di calcoli, i valori della funzione  $f$  vengono memorizzate in variabili temporanee in modo tale che ad ogni iterazione si debba valutare la funzione solamente in un nuovo punto.

### 2.4.1 Risultati con Newton-Bisezione

Il procedimento è analogo all'ultima parte del paragrafo del metodo di Newton, quindi numero di risultati da calcolare pari a 10 e valore di  $\mu_R$  di 200, 1000 e 2000. Il risultato viene mostrato nella Tabella 2.7 e come si può notare le soluzioni ora sono accettabili.

Rimane un ultimo test, ed è quello relativo a  $\mu_R$  sempre molto elevato ma con un numero di risultati pari a 50.

Come mostrato nella Tabella 2.8 anche in questo caso le radici sono corrette e fanno parte del loro intervallo di appartenenza.

L'algoritmo usato funziona in modo efficace per il sistema preso in esame.

$\mu_R = 200$	$\mu_R = 1000$	$\mu_R = 2000$
4.4711	4.4889	4.4912
7.6868	7.7175	7.7214
10.8499	10.8932	10.8987
13.9963	14.0521	14.0592
17.1353	17.2036	17.2121
20.2703	20.3510	20.3611
23.4030	23.4960	23.5077
26.5342	26.6394	26.6527
29.6643	29.7818	29.7967
32.7939	32.9235	32.9399

Tabella 2.7: Radici con il metodo ibrido con  $n = 10$ 

4.4912	36.0826	67.4957	98.9007	130.3034
7.7214	39.2248	70.6364	102.0410	133.4436
10.8987	42.3667	73.7770	105.1813	136.5838
14.0592	45.5084	76.9176	108.3216	139.7240
17.2121	48.6498	80.0581	111.4619	142.8642
20.3611	51.7911	83.1986	114.6022	146.0043
23.5077	54.9322	86.3391	117.7424	149.1445
26.6527	58.0732	89.4795	120.8827	152.2847
29.7967	61.2141	92.6199	124.0229	155.4248
32.9399	64.3550	95.7603	127.1631	158.5650

Tabella 2.8: Radici con il metodo ibrido con  $n = 50$  e  $\mu_R = 2000$ 

## 2.5 Confronto Newton-Bisezione

La prima domanda è sicuramente diretta al metodo di Newton: precedentemente ho scritto che questo metodo converge quasi sempre, a meno di funzioni poco regolari e se il punto iniziale è lontano dalla radice. Se per  $\mu_R$  relativamente piccoli la funzione (2.1) è regolare, ciò non è corretto per  $\mu_R$  elevati ed è per questo motivo per cui Newton non riesce a trovare correttamente le radici, anche variando di una quantità dipendente dal valore di  $\mu_R$ .

Per quanto riguarda il confronto, la differenza tra i due metodi può essere notata immediatamente attraverso l'analisi del numero di iterazioni per  $\mu_R=2000$ . Ricordandosi che per  $\mu_R=2000$  nella Tabella 2.6 gli ultimi 3 valori erano errati, con questo metodo ibrido interviene il metodo di bisezione riuscendo a calcolare correttamente le radici ma ad un prezzo: nella Tabella 2.9 si vede benissimo come i primi 7 valori calcolati con Newton abbiano un numero di iterazioni molto basso,

5
4
3
4
5
6
8
28
28
28

Tabella 2.9: Numero di iterazioni con il modello ibrido

addirittura inferiore a 10, mentre gli ultimi tre calcolati con il metodo di Bisezione il numero di passaggi sia di poco inferiore di 30.

Il motivo va ricercato nella diversa velocità di convergenza dei due metodi usati: nel primo caso infatti abbiamo una convergenza quadratica (di ordine 2), mentre nel secondo caso abbiamo una convergenza lineare (di ordine 1).

**Definizione 3.** Data una successione  $\{x_k\}$ , con  $x_k$  convergente ad  $\alpha$ , diremo che la successione ha ordine di convergenza  $p$  se esistono due numeri reali  $p \geq 1$  e  $c > 0$ , con  $c =$  costante asintotica di errore, tali che

$$\lim_{k \rightarrow \infty} \frac{|e_{k+1}|}{|e_k|^p} = c \tag{2.4}$$

dove  $c < 1$  se  $p=1$ .

Inoltre se  $p=1$  la convergenza si definisce “lineare”, se  $p=2$  la convergenza si definisce “quadratica”.

Si può dimostrare che, in caso di radici semplici (cioè radici con molteplicità 1), il metodo di Newton converge in modo quadratico, mentre nel caso di radici multiple (molteplicità maggiore di 1), il metodo converge in modo lineare. Poiché in questo caso siamo di fronte a radici semplici, l’ordine di convergenza del metodo di Newton è pari a 2 e quindi è quadratico. Invece per quanto riguarda il metodo di Bisezione la convergenza è di tipo lineare.

Analizzando bene l’equazione 2.4 si può ben capire il significato delle due convergenze: l’ordine infatti esprime il numero di cifre decimali che il metodo guadagna ad ogni iterazione rispetto alla soluzione esatta. Nella pratica, la velocità di convergenze è strettamente legata al suo ordine, infatti tanto è più grande l’ordine tanto maggiore è la velocità con cui la soluzione  $x_k$  converge alla soluzione esatta  $\alpha$ .

# Capitolo 3

## Metodo di Prony

Il metodo di Prony è un algoritmo sviluppato da Gaspard Riche de Prony nel 1795 [7]. Ad ogni modo le prime applicazioni, a causa della natura dell'algoritmo, si sono viste solamente con la presenza dei computer. Similmente alla trasformata di Fourier, il metodo di Prony estrae utili informazioni da un segnale campionato uniformemente nel tempo e sviluppa una serie di esponenziali complesse o sinusoidi. Questo permette la stima della frequenza, ampiezza, fase e i vari componenti del segnale.

### 3.1 L'algoritmo

Supponiamo di avere una funzione del tipo

$$f(t) = \sum_{j=1}^n c_j e^{-\lambda_j t} \quad (3.1)$$

dove  $c_j, \lambda_j \in \mathbb{C}$ , e di conoscere i valori esatti di tale funzione  $f$  in  $2N$  punti  $t_i$  equispaziati (con  $N \geq n$ ), ossia  $[f(t_0), f(t_1), f(t_2), \dots, f(t_{2N-1}), f(t_{2N})]^T$ .

Il nostro obiettivo è quello di ricostruire la somma e quindi di identificare i parametri  $\lambda_j$  e  $c_j$ , con

$$j = 1, 2, \dots, n.$$

Tale metodo si suddivide in due steps. Nel primo si identificano i parametri  $\lambda_j$  (in realtà una loro approssimazione) mediante il calcolo degli zeri di un polinomio (detto polinomio di Prony), nel secondo si identificano i coefficienti  $c_j$  mediante la risoluzione di un sistema lineare nel quale intervengono i  $\lambda_j$  approssimati.

### 3.1.1 Primo step

Riscriviamo la (3.1) come

$$f(t) = \sum_{j=1}^n c_j z_j^t, \quad \text{con } z_j = e^{-\lambda_j} \quad (3.2)$$

e assumiamo di conoscere la (3.2) in  $2N$  numeri interi.

Quindi conosciamo

$$f(k) = \sum_{j=1}^n c_j z_j^k, \quad \text{con } k = 0, 1, \dots, 2N - 1. \quad (3.3)$$

La chiave del metodo è quella di interpretare (3.3) come la soluzione di una equazione alle differenze di ordine  $2N$ , del tipo

$$\sum_{k=0}^{n-1} p_k f(k+m) = -f(m+n), \quad \text{con } m = 0, 1, \dots, n-1, \quad (3.4)$$

dove i  $p_k$  sono i coefficienti del polinomio di Prony i cui zeri sono gli  $z_j$  che stiamo cercando. Più precisamente

$$P(z) = \prod_{j=1}^n (z - z_j) = \sum_{k=0}^n p_k z^k, \quad \text{con } p_n \equiv 1. \quad (3.5)$$

Per ottenere i coefficienti del polinomio dobbiamo semplicemente risolvere il sistema (3.4), cioè il sistema così rappresentato

$$\begin{bmatrix} f(0) & f(1) & f(2) & \dots & f(n-1) \\ f(1) & f(2) & f(3) & \dots & f(n) \\ f(2) & f(3) & f(4) & \dots & f(n+1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f(n-2) & f(n-1) & f(n) & \dots & f(2n-3) \\ f(n-1) & f(n) & f(n+1) & \dots & f(2n-2) \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_{n-2} \\ p_{n-1} \end{bmatrix} = \begin{bmatrix} -f(n) \\ -f(n+1) \\ -f(n+2) \\ \vdots \\ -f(2n-2) \\ -f(2n-1) \end{bmatrix}$$

Osserviamo che la matrice dei coefficienti ha le antidiagonali costanti: questo tipo di matrice si definisce **matrice di Hankel**, la cui espressione matematica è

$$f_{ij} = f_{i-1, j+1}.$$

Risolto il sistema e quindi ottenuto il vettore dei coefficienti  $p = [p_0, p_1, \dots, p_n]^T$ ,

il passo successivo è quello di calcolare i suoi zeri. A questo scopo costruiamo la matrice compagna  $C$  così definita

$$C = \begin{bmatrix} 0 & \dots & \dots & 0 & -p_0 \\ 1 & \dots & \dots & 0 & -p_1 \\ 0 & 1 & 0 & \dots & -p_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & -p_{n-1} \end{bmatrix} \quad (3.6)$$

e calcoliamo i suoi autovalori, i quali corrisponderanno proprio agli  $z_j$  cercati. Per ricavare i valori di  $\tilde{\lambda}_j$  occorre semplicemente calcolare il logaritmo degli  $z_j$  e cambiarlo di segno,

$$\tilde{\lambda}_j = -\log(z_j).$$

### 3.1.2 Secondo step

Per risolvere il problema restano solamente i valori di  $c_j$  che corrispondono alla soluzione di questo sistema lineare

$$\begin{bmatrix} z_1^0 & z_2^0 & \dots & z_n^0 \\ z_1^1 & z_2^1 & \dots & z_n^1 \\ z_1^2 & z_2^2 & \dots & z_n^2 \\ \vdots & \vdots & \vdots & \ddots \\ z_1^n & z_2^n & \dots & z_n^n \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ \vdots \\ f(n) \end{bmatrix} \quad (3.7)$$

Da notare inoltre che anche in questo caso la matrice dei coefficienti è particolare e viene definita **matrice di Vandermonde**, caratterizzata dall'aver gli elementi in progressione geometrica. L'espressione matematica in questo caso è data da

$$z_{ij} = z_j^{i-1}.$$

## 3.2 Esempio

Per realizzare un esempio generiamo un generico segnale del tipo (3.1), con

$$\mathbf{c} = \begin{bmatrix} 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{bmatrix}, \quad \boldsymbol{\lambda} = \begin{bmatrix} 10 \\ 9 \\ 8 \\ 7 \\ 6 \end{bmatrix}$$

Per calcolare la funzione, determiniamo i nostri  $2N$  punti equispaziati ponendo

$$t = 1, 2, \dots, 10,$$

rispettando quindi la condizione  $N \geq n$ , essendo  $N=n=5$ .

In questo modo la nostra funzione  $f(t)$  assume i seguenti valori

$$f(t) = \begin{bmatrix} 0.0060 \\ 8.2161 \cdot 10^{-6} \\ 1.6868 \cdot 10^{-8} \\ 3.9173 \cdot 10^{-11} \\ 9.4850 \cdot 10^{-14} \\ 2.3311 \cdot 10^{-16} \\ 5.7600 \cdot 10^{-19} \\ 1.4261 \cdot 10^{-21} \\ 3.5335 \cdot 10^{-24} \\ 8.7573 \cdot 10^{-27} \end{bmatrix}$$

Il prossimo passo è quello di trovare i coefficienti del polinomio di Prony, tramite il sistema

$$H\mathbf{p}=\mathbf{b},$$

dove

$$H = \begin{bmatrix} 0.0060 & 8.2161 \cdot 10^{-6} & 1.6868 \cdot 10^{-8} & 3.9173 \cdot 10^{-11} & 9.4850 \cdot 10^{-14} \\ 8.2161 \cdot 10^{-6} & 1.6868 \cdot 10^{-8} & 3.9173 \cdot 10^{-11} & 9.4850 \cdot 10^{-14} & 2.3311 \cdot 10^{-16} \\ 1.6868 \cdot 10^{-8} & 3.9173 \cdot 10^{-11} & 9.4850 \cdot 10^{-14} & 2.3311 \cdot 10^{-16} & 5.7600 \cdot 10^{-19} \\ 3.9173 \cdot 10^{-11} & 9.4850 \cdot 10^{-14} & 2.3311 \cdot 10^{-16} & 5.7600 \cdot 10^{-19} & 1.4261 \cdot 10^{-21} \\ 9.4850 \cdot 10^{-14} & 2.3311 \cdot 10^{-16} & 5.7600 \cdot 10^{-19} & 1.4261 \cdot 10^{-21} & 3.5335 \cdot 10^{-24} \end{bmatrix}$$

$$\mathbf{p} = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -2.3311 \cdot 10^{-16} \\ 5.7600 \cdot 10^{-19} \\ 1.4261 \cdot 10^{-21} \\ 3.5335 \cdot 10^{-24} \\ 8.7573 \cdot 10^{-27} \end{bmatrix}$$

Da notare come la matrice dei coefficienti  $H$  è sempre una matrice di Hankel.

I valori dei coefficienti di Prony sono

$$\mathbf{p} = \begin{bmatrix} -4.2484 \cdot 10^{-18} \\ 1.4704 \cdot 10^{-13} \\ -1.3527 \cdot 10^{-9} \\ 4.0324 \cdot 10^{-6} \\ -3.8949 \cdot 10^{-3} \end{bmatrix}$$

In questo specifico esempio la matrice  $H$  ha un condizionamento pari a  $2.6 \cdot 10^{28}$ , cioè è in buona approssimazione una matrice singolare, cioè una matrice con determinante nullo. Inoltre alcune linee della matrice  $H$  risultano linearmente dipendenti e quindi l'intero sistema risulta essere mal condizionato.

Il passo successivo è quello di calcolare gli zeri del polinomio avente come coefficienti proprio i valori di  $p$  che abbiamo appena trovato. Per fare ciò costruiamo la matrice compagna così definita

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 4.2484 \cdot 10^{-18} \\ 1 & 0 & 0 & 0 & -1.4704 \cdot 10^{-13} \\ 0 & 1 & 0 & 0 & 1.3527 \cdot 10^{-9} \\ 0 & 0 & 1 & 0 & -4.0324 \cdot 10^{-6} \\ 0 & 0 & 0 & 1 & -3.8949 \cdot 10^{-3} \end{bmatrix}$$

Da notare che questo algoritmo è esattamente lo stesso che viene utilizzato all'interno della routine di MATLAB, `roots`.

Gli autovalori di questa matrice, calcolati con MATLAB tramite il comando  $Z = \text{eig}(C)$ , corrispondono ai nostri  $z_j$  e, una volta calcolati il loro logaritmo cambiato di segno otteniamo i nostri  $\tilde{\lambda}_j$ . Abbiamo quindi

$$\mathbf{z} = \begin{bmatrix} 4.5400 \cdot 10^{-5} \\ 1.2341 \cdot 10^{-4} \\ 3.3546 \cdot 10^{-4} \\ 9.1188 \cdot 10^{-4} \\ 2.4788 \cdot 10^{-3} \end{bmatrix}, \quad \tilde{\boldsymbol{\lambda}} = \begin{bmatrix} 10 \\ 9 \\ 8 \\ 7 \\ 6 \end{bmatrix}$$

Come per i valori di  $f(t)$ , anche questi valori trovati ovviamente non sono precisi ma hanno un loro errore rispetto ai valori esatti. La norma 2 e la norma infinito in questo esempio sono dell'ordine di  $10^{-8}$ .

Rimangono da calcolare i valori dei  $c_j$  approssimati che corrispondono alla soluzione del seguente sistema lineare

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ z_1^1 & z_2^1 & z_3^1 & z_4^1 & z_5^1 \\ z_1^2 & z_2^2 & z_3^2 & z_4^2 & z_5^2 \\ z_1^3 & z_2^3 & z_3^3 & z_4^3 & z_5^3 \\ z_1^4 & z_2^4 & z_3^4 & z_4^4 & z_5^4 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} 0.0060 \\ 8.2161 \cdot 10^{-6} \\ 1.6868 \cdot 10^{-8} \\ 3.9173 \cdot 10^{-11} \\ 9.4850 \cdot 10^{-14} \end{bmatrix}$$

La soluzione del sistema risulta

$$\tilde{\mathbf{c}} = \begin{bmatrix} 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

ma, come nel caso di  $\tilde{\lambda}$ , il vettore  $\tilde{\mathbf{c}}$  non è preciso rispetto ai valori reali e infatti ha una norma 2 e infinito dell'ordine di  $10^{-7}$ .

E' interessante notare cosa succede alla matrice H aumentando il numero di istanti t. Nell'esempio noi conosciamo già il numero n dei coefficienti e, giusto in sede di esempio, abbiamo posto  $N = 5$ . Il metodo quindi viene applicato ad un numero di misurazioni pari a  $2N = 10$ . In realtà noi non conosciamo il numero dei coefficienti n e di conseguenza non conosciamo il numero di misurazioni tali da ottenere un risultato soddisfacente. Cosa succede quindi aumentando N?

Nel seguente esempio, sempre con gli stessi valori di  $\mathbf{c}$  e  $\lambda$  del primo esempio, poniamo

$$N = [5, 25, 50].$$

Se calcoliamo il condizionamento di H nei tre diversi casi otteniamo, rispettivamente,  $2.56 \cdot 10^{28}$ ,  $4 \cdot 10^{140}$  e  $3 \cdot 10^{270}$ . In pratica aumentando il numero di misurazioni il sistema porta ad una soluzione che può essere errata poichè la matrice corrispondente dei coefficienti è mal condizionata.

L'effetto del condizionamento elevato si può verificare facilmente nelle Figure 3.1, 3.2 e 3.3, in cui sono rappresentate le  $f(t)$  reali e i pallini che rappresentano i coefficienti calcolati tramite il metodo di Prony. Nei primi due casi i risultati sono

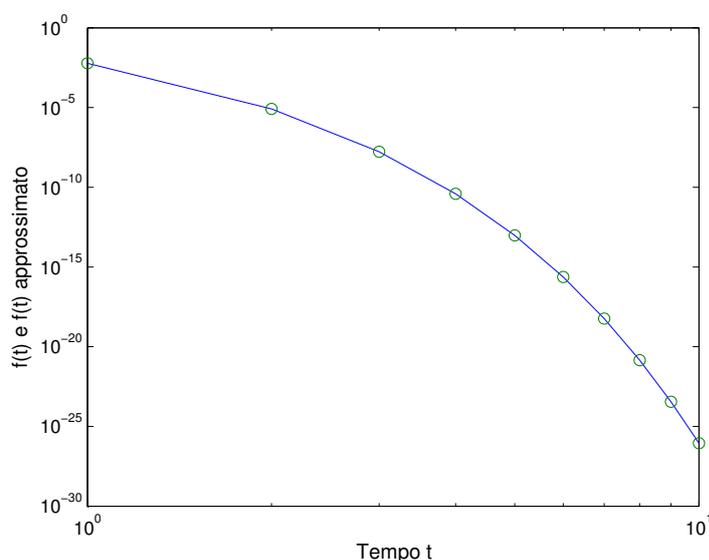
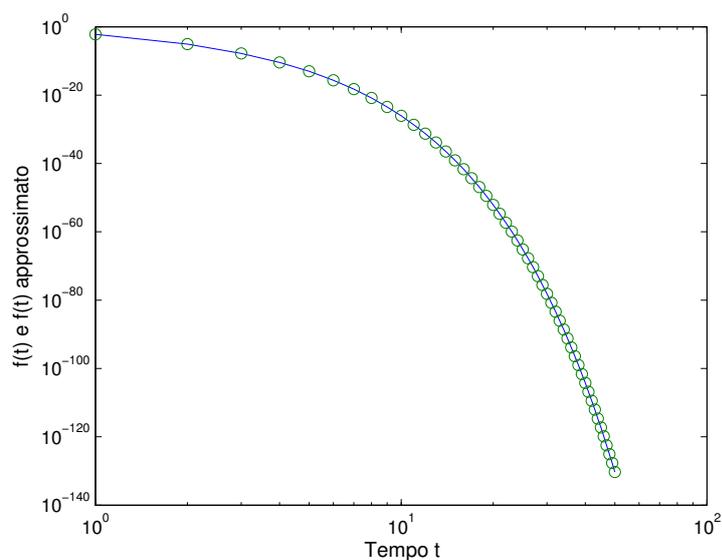
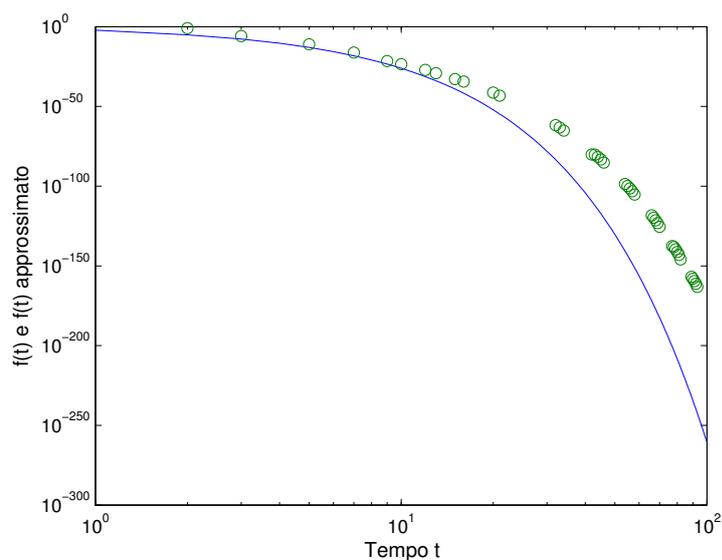


Figura 3.1: Prony applicato all'esempio con  $N=5$

accettabili, mentre nell'ultimo caso solo i primi 4 punti corrispondono a valori reali della curva, tutti quelli successivi sono totalmente errati.

Figura 3.2: Prony applicato all'esempio con  $N=25$ Figura 3.3: Prony applicato all'esempio con  $N=50$ 

E' doveroso notare che nonostante l'aumento delle misurazioni, il rango corrispondente della matrice  $H$  rimane ad un valore costante pari a 3. In pratica aggiungendo un numero di misurazioni di molto superiori ad  $n$  noi effettivamente aggiungiamo delle informazioni che non servono e che non vengono utilizzati ai fini

del calcolo dei coefficienti.

Il punto del discorso si palesa nel calcolo dei valori singolari di una matrice (mediante la fattorizzazione *SVD*, Singular Value Decomposition, in MATLAB utilizzabile tramite la routine `svd` e che restituisce un vettore contenente i valori singolari in ordine decrescente), cioè un metodo particolare per la fattorizzazione di una matrice e che viene utilizzato per il calcolo del rango (la routine di MATLAB `rank` utilizza effettivamente la *SVD* per calcolare il rango e che corrisponde al numero di valori singolari non nulli presenti nel vettore restituito).

Poichè la seguente verifica è stata effettuata grazie a MATLAB, il valore esattamente nullo non esiste all'interno di questi vettori. Si è deciso di analizzare ciascun valore presente nel vettore e di confrontarlo con un  $\epsilon$  (valore che dipende dalla precisione in virgola mobile del calcolatore) e, nel caso in cui quel dato elemento risulta essere inferiore ad  $\epsilon$ , di portarlo ad un valore di circa  $\epsilon^2$  per far risaltare il gradino che si viene a formare nel grafico. Nel nostro caso il valore di  $\epsilon$  ottenuto tramite la routine `eps` di MATLAB è di  $2 \cdot 10^{-16}$ .

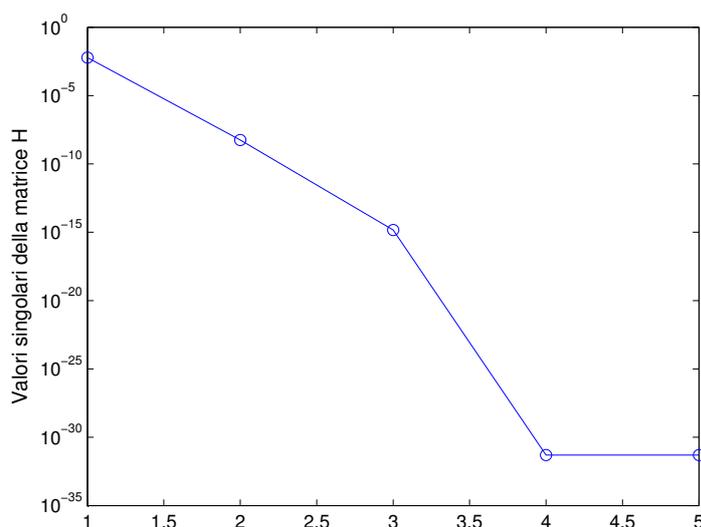
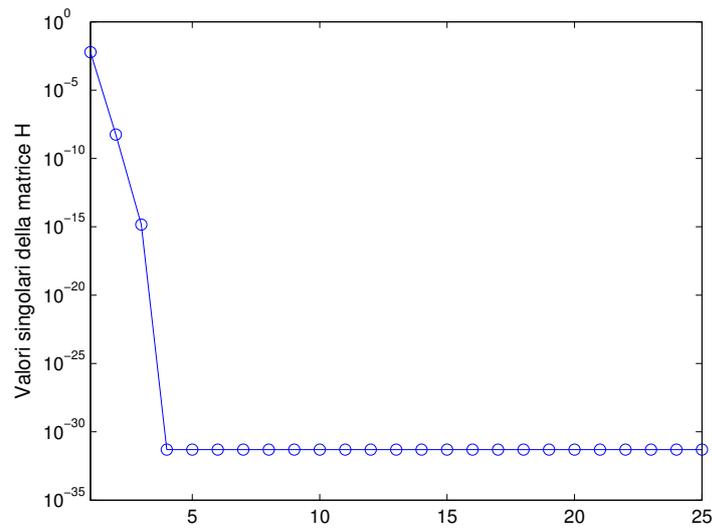
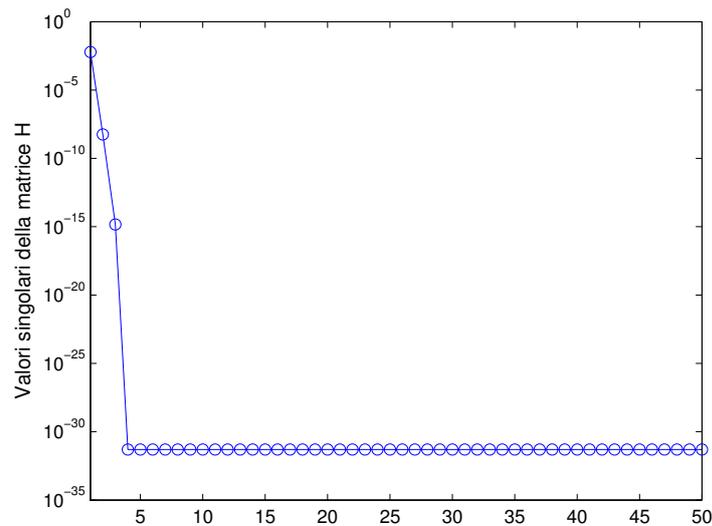


Figura 3.4: Valori singolari con  $N = 5$

Come si può notare effettivamente in tutti e tre casi nelle Figure 3.4, 3.5 e 3.6 i valori accettabili del vettore sono tre, esattamente come il rango della matrice  $H$  nei tre casi.

Ultima nota per questo esempio sono i valori della  $f(t)$ , ad esempio per  $N=10$ , con  $t$  espresso in secondi, a Pag. 24. Come si può notare metà dei valori presenti

Figura 3.5: Valori singolari con  $N = 25$ Figura 3.6: Valori singolari con  $N = 50$ 

nel vettore  $f(t)$  sono così piccoli da poter essere tranquillamente considerati come nulli, non portando quindi alcuna informazione utile ai fini dell'esercizio. Infatti come suggerisce la SVD, il rango della matrice è tre e non cinque come il numero dei coefficienti, e ciò suggerisce che per la risoluzione del sistema debba essere

utilizzato un opportuno metodo di regolarizzazione.

Il motivo di ciò è che per questo modello la misura del tempo in secondi non è adatta ed è necessario modificarlo, da  $Z_j^t$  a  $Z_j^{ht}$ , con  $h$  scelto arbitrariamente. Ricordando che il metodo di Prony funziona solo con gli interi,  $t$  deve rimanere un intero e per modificare la scala (millisecondi, nanosecondi etc... a seconda delle esigenze) occorre semplicemente modificare  $h$  (es. in  $10^{-3}$ ).

Il modello attuale quindi diventa

$$f(t) = \sum_{j=1}^n c_j z_j^{ht}, \quad \text{con } z_j = e^{-\lambda_j}. \quad (3.8)$$

Il valore della funzione  $f(t)$  per  $N=5$  e  $h=10^{-1}$  diventano

$$f(t) = \begin{bmatrix} 6.3556 \\ 2.7379 \\ 1.2001 \\ 0.5355 \\ 0.2433 \\ 0.1125 \\ 0.0529 \\ 0.0253 \\ 0.0123 \\ 0.0060 \end{bmatrix}$$

e il condizionamento della matrice  $H$  passa da  $2.6 \cdot 10^{28}$  a  $3 \cdot 10^{11}$ .

$$\tilde{\mathbf{c}} = \begin{bmatrix} 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{bmatrix}, \quad \tilde{\boldsymbol{\lambda}} = \begin{bmatrix} 1.0 \\ 0.9 \\ 0.8 \\ 0.7 \\ 0.6 \end{bmatrix}$$

Da notare che i valori di  $\tilde{\boldsymbol{\lambda}}$  sono proprio scalati dell' $h$  che abbiamo scelto all'inizio. Conoscendo infatti il valore di  $h$ , scelto arbitrariamente (ma con cognizione di causa, in modo da non portare il focus nel segnale in degli istanti di tempo in cui la  $f(t)$  rimane pressochè costante), possiamo riottenere i  $\tilde{\boldsymbol{\lambda}}$  reali semplicemente dividendo il vettore per  $h$ .

Infine, con la presenza di  $h$  e con il miglioramento non solo dei valori di  $f(t)$  ma anche del condizionamento (che passa da  $3 \cdot 10^{270}$  a  $7 \cdot 10^{40}$ ), anche per  $N = 50$  la funzione viene approssimata in modo soddisfacente da Prony (come si può notare dalla Figura 3.7) e i valori di  $\tilde{\boldsymbol{\lambda}}$  e di  $\tilde{\mathbf{c}}$  sono quelli corretti. Infine il rango  $\rho = n$ , cioè pari a 5, in tutti e tre i casi con  $h = 10^{-1}$ .

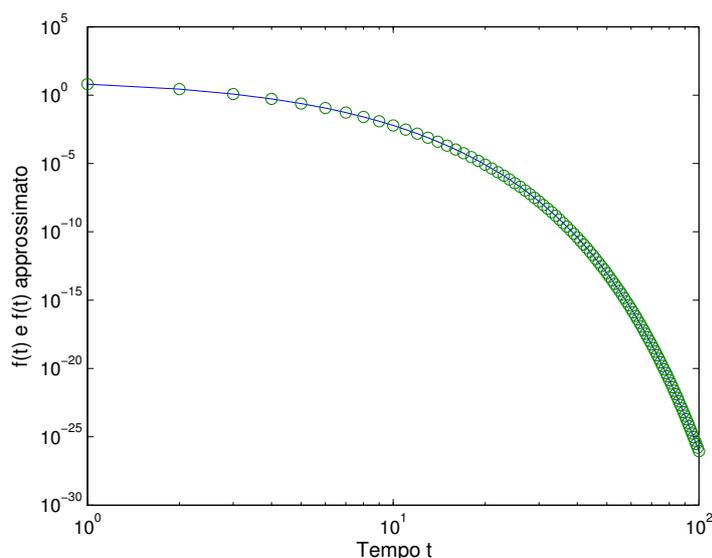


Figura 3.7: Prony applicato all'esempio con  $N = 50$  e  $h = 10^{-1}$

### 3.3 Confronto con la trasformata di Fourier

Rispetto a Fourier, il metodo di Prony ha un particolare vantaggio che risulta indispensabile per la risoluzione del problema indiretto affrontato in questa tesi, cioè il calcolo dei parametri degli esponenziali (residuo e polo) a partire da un segnale rappresentante la risposta impulsiva del target [9].

Il metodo di Prony è, infatti, di tipo parametrico e permette, oltre all'individuazione dell'ampiezza, fase e frequenza, il calcolo dei coefficienti di smorzamento di un segnale. Di conseguenza tale metodo ci permette di analizzare al meglio un segnale espresso come serie di esponenziali smorzati che è proprio il tipo di funzione che viene analizzata in questo lavoro.

Di contro però le sue prestazioni su un segnale contenente del rumore non sono affidabili. Più nel dettaglio il calcolo ad esempio dei parametri di smorzamento sono solitamente sovrastimati in modo significativo ed è per questo che i dati analizzati con il metodo di Prony standard devono essere **esatti**. Proprio per questo motivo il metodo di Prony è stato revisionato un numero considerevole di volte in modo tale da migliorare le prestazioni in caso di segnali affetti da rumore.

# Capitolo 4

## Analisi dei dati

Dopo aver affrontato il problema della risoluzione dell'equazione non lineare presente nel secondo capitolo, abbiamo tutti i valori per poter generare dei dati esatti relativi alla risposta impulsiva del target.

Il segnale da generare è quindi la Formula (1.5), in relazione al tempo adimensionale  $T (=t/\mu_0 a^2 \sigma)$ .

Supponiamo quindi di disporre di un numero  $2N$  di punti in cui si conosce la funzione pari a 100 e la generiamo per i valori di  $\mu_R = [ 2 \ 5 \ 10 \ 20 \ 50 \ 100 \ 200 ]$ . Dalla Figura 4.1 risulta con tutte le curve intrecciate. Man mano che  $\mu_R$  aumenta, il punto in cui la curva raggiunge lo 0 si sposta verso destra.

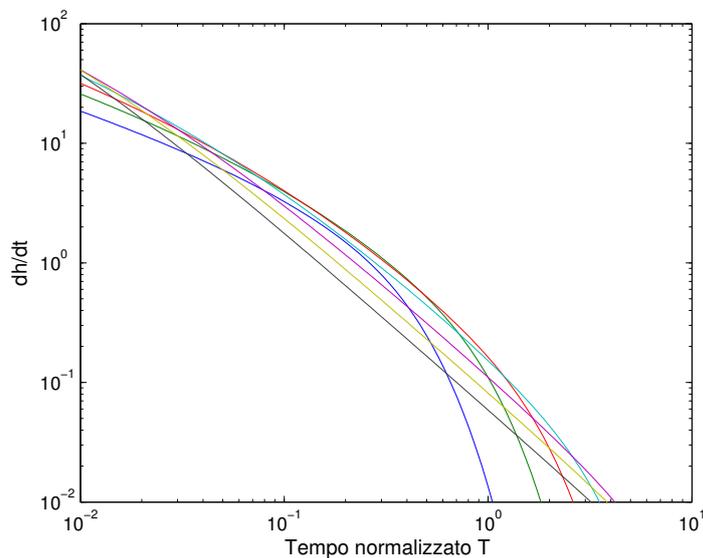


Figura 4.1: Risposta impulsiva del target con  $\mu_R = [ 2 \ 5 \ 10 \ 20 \ 50 \ 100 \ 200 ]$

In effetti dalla Figura 4.1 non si notano eventuali comportamenti particolari al variare di  $\mu_R$ . Per questioni di visibilità moltiplichiamo la (1.5) per  $\mu_R$ .

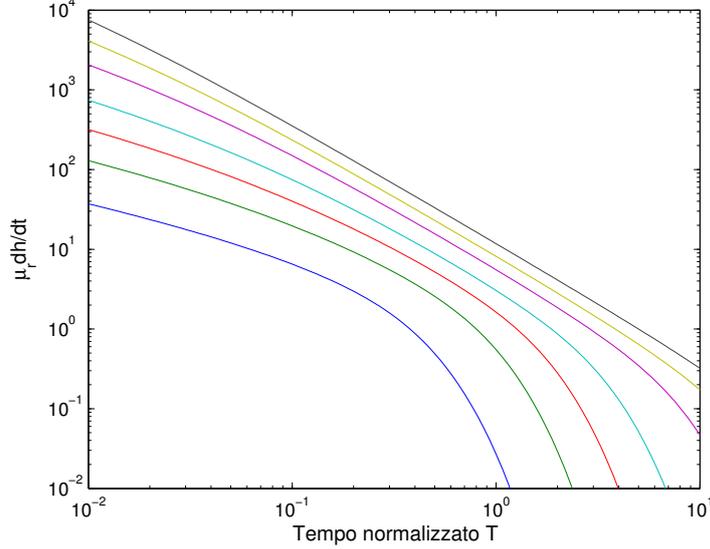


Figura 4.2: Risposta impulsiva  $\cdot \mu_R$  del target con  $\mu_R = [ 2 \ 5 \ 10 \ 20 \ 50 \ 100 \ 200 ]$

Stavolta dalla Figura 4.2 salta subito all'attenzione una cosa: per valori di  $\mu_R$  elevati il comportamento della risposta impulsiva su scala logaritmica **tende ad essere lineare** (proprio la curva più a destra corrisponde ad un valore di  $\mu_R=200$ ). In pratica per notare subito la presenza di materiale altamente conduttivo all'interno del sottosuolo si può analizzare in primis il comportamento della curva.

## 4.1 Tempo reale

In realtà nelle applicazioni reali, gli strumenti che si occupano di rilevare le risposte impulsive (come il già citato NanoTEM della Zonge), restituiscono la funzione richiesta nel dominio del tempo. Il motivo è semplice, non conoscendo i parametri caratteristici dell'ordigno lo strumento non può scalare opportunamente nè  $\mathbf{c}$  e nè  $\boldsymbol{\lambda}$ . Bisogna quindi analizzare la sua risposta impulsiva non scalata. A differenza della 1.5, noi dobbiamo considerare

$$-\frac{\delta h(t)}{\delta t} = \frac{2\pi a}{\mu_R \sigma} \left[ 6 \sum_{n=1}^{\infty} \frac{\delta_n^2 e^{-\frac{\delta_n^2 t}{\mu_R \mu_0 \sigma a^2}}}{(\mu_R + 2)(\mu_R - 1) + \delta_n^2} - \delta(t) \right]. \quad (4.1)$$

Possiamo quindi generare un paio di esempi con i parametri scelti arbitrariamente, tenendo sempre conto del tempo scalato in modo che sia coerente con il processo fisico che stiamo analizzando (dell'ordine dei microsecondi, quindi  $h = 10^{-6}$ ).

### 4.1.1 Esempio 1

Nel primo esempio consideriamo un ordigno in ferrite avente  $\sigma = 5 \cdot 10^6 \frac{S}{m}$ , poniamo  $a = 10\text{cm} = 0.1\text{m}$  e  $\mu_R = 290$ .

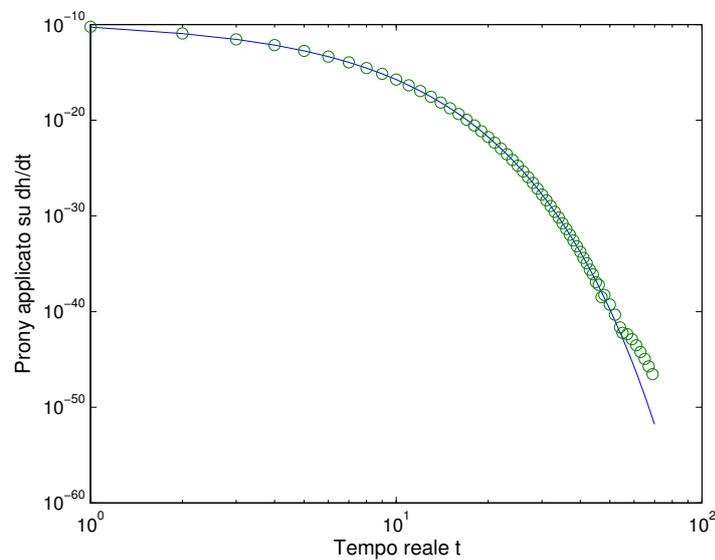


Figura 4.3: Prony applicato alla risposta impulsiva di un ordigno in ferrite

### 4.1.2 Esempio 2

Nel secondo e ultimo esempio consideriamo invece un ordigno in acciaio avente  $\sigma = 9.5 \cdot 10^6 \frac{S}{m}$ , poniamo  $a = 10\text{cm} = 0.1\text{m}$  e  $\mu_R = 200$ .

Come si può notare la curva viene seguita in modo soddisfacente dal metodo in esame.

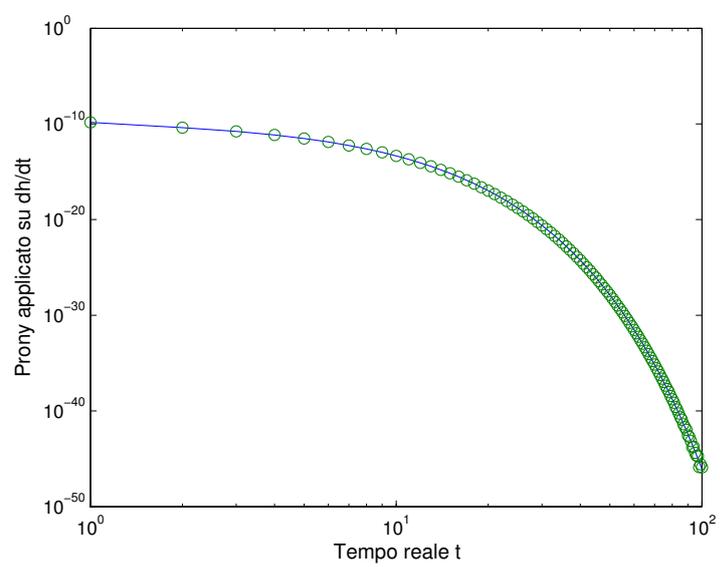


Figura 4.4: Prony applicato alla risposta impulsiva di un ordigno in acciaio

# Conclusioni

Con l'applicazione del metodo di Prony è stato possibile identificare una serie di esponenziali con i relativi coefficienti che sono in grado di approssimare in modo soddisfacente la risposta impulsiva di un ordigno inesplosivo sepolto.

Con la conoscenza di questi parametri, è possibile, tramite un ulteriore metodo di fitting (ad esempio il metodo dei minimi quadrati), ricavare i parametri dell'ordigno.

I punti critici del metodo di Prony e quindi ciò che permette al metodo stesso di essere affidabile sono sia la scala dei tempi (in base al segnale) ma soprattutto il condizionamento della matrice di Hankel. Nel corso degli anni sono stati sviluppati altri metodi che si basano su Prony come il metodo Matrix-Pencil (fascio di matrici) nel quale fondamentalmente non si calcolano più i coefficienti del polinomio di Prony ma si calcolano gli autovalori del fascio di matrici [6], generalmente definito come

$$A - \lambda B,$$

con A e B matrici quadrate complesse.

Di norma comunque, questa classe di metodi sono ormai lo standard per ricavare i parametri di un segnale.

# Bibliografia

- [1] Jean Van Bladel. *Electromagnetic Fields - Second Edition*. Donald G. Dudley, Series, 2007, pp. 1052–1059.
- [2] Scott Urquhart K.L. Zonge D.D. Snyder Scott MacInnes. *UXO classification using characteristic modes of the broadband electromagnetic induction response*. 2000.
- [3] Kenneth P. Spies James R. Wait. *Quasi-static transient response of a conducting permeable sphere*. 1969.
- [4] Alex Kaufman. *A Paradox in Geoelectromagnetism and its Resolution, Demonstrating the Equivalence of Frequency and Transient Domain Methods*. 1989.
- [5] Alex Kaufman. *Frequency and Transient Responses of Electromagnetic Fields Created by Current in Confined Conductors*. 1989.
- [6] C. van der Mee L. Fermo e S. Seatzu. *Matrix-pencil method for estimating parameters of monomial-exponential sums*. 2015.
- [7] Gaspard Riche de Prony. *Essai expérimental et analytique: sur les lois de la dilatabilité de fluides élastique et sur celles de la force expansive de la vapeur de l'alkool, à différentes températures*. Journal de l'École Polytechnique Floréal et Plairial, an III, Volume 1, 1795, pp. 24–76.
- [8] Giuseppe Rodriguez. *Algoritmi numerici*. Pitagora Editrice, 2008, Cap. 7.
- [9] Satnam Singh. *Application of prony analysis to characterize pulsed corona reactor measurements*. University of Wyoming, 2003, pp. 17–18.