

# Identificazione parametrica in un modello di meccanica respiratoria

Mastrofini Alessandro

alessandro.mastrofini@alumni.uniroma2.eu

## Abstract

Nel seguente report viene analizzato un modello lineare di meccanica polmonare. Il modello corrisponde ad un circuito RLC con parametri incogniti.

L'identificazione parametrica permette di risalire ai coefficienti di governo del sistema una volta stimato un modello matematico.

Diversi approcci permettono di minimizzare la differenza tra i parametri stimati e le misure reali.

Viene affrontato un approccio privo di differenziazione numerica, tramite algoritmo di Nelder-Mead e un approccio mediante il metodo di Gauss-Newton rilassato.

Entrambi i metodi portano alla stima corretta dei parametri ma alcune considerazioni permettono di ottimizzarne le prestazioni.

## 1 Introduzione

La meccanica respiratoria viene descritta in letteratura da numerosi modelli. Tra questi, un modello valido in determinate condizioni è rappresentato da un circuito RLC. Il modello permette di descrivere la meccanica respiratoria di un individuo sano.

A partire dalle misure dell'individuo è possibile stimare i parametri per rappresentarlo tramite il modello stimato. Questo processo, sotto il nome di ottimizzazione parametrica, permette di adattare il sistema generale al caso specifico.

Nella seguente analisi viene affrontata la tecnica di identificazione parametrica tramite minimizzazione della differenza tra la risposta misurata e quella stimata.

Differenti tecniche di ottimizzazione permettono di stimare i parametri che permettono di descrivere il sistema considerato.

Viene descritto sia il metodo di Nelder-Mead sia il metodo di Gauss-Newton, analizzandone punti in

comune e differenze. Vengono fatte anche alcune considerazioni sulla convergenza del metodo e sul corretto condizionamento del problema.

## 2 Background

L'identificazione di sistemi è un insieme di tecniche che si prefigge lo scopo di identificare i parametri e il sistema tale da descrivere un certo fenomeno. La prima grande distinzione vede la separazione di due tecniche: identificazione parametrica e non parametrica. La prima parte stimando un modello del fenomeno e punta ad identificare i parametri che lo rappresentano. L'identificazione non parametrica, invece, procede senza conoscere il modello del fenomeno, al più ne stima alcune caratteristiche, e punta ad identificare il sistema che meglio lo descrive.

In questo caso si parte da un modello ben noto, un modello di meccanica polmonare lineare descritto da un circuito RLC, e se ne identificano i parametri descrittivi.

## 2.1 Sistema RLC

Il circuito RLC (fig. 2) permette, seguendo l'analogia elettrica, di descrivere un modello approssimato della meccanica respiratoria. Il modello prevede una serie di resistenza, induttore e capacità. Sulla base dell'analogia tra la corrente elettrica e il flusso polmonare è possibile rappresentare, in un modello a parametri concentrati, le componenti di compliance e resistenza elastica dei polmoni [1].

Questo modello permette di affrontare i casi fisiologici in cui il flusso entrante nei polmoni è predominante rispetto al flusso bloccato (flusso che invece aumenta notevolmente in caso di patologie ostruttive). L'analogia vede l'associazione della differenza di pressione alla differenza di potenziale, la corrente al flusso d'aria e l'induttanza come un'inertanza, ovvero la differenza di pressione richiesta per causare una variazione unitaria nel tasso di variazione del flusso nel tempo.

In un caso paziente specifico sarebbero necessarie misure sperimentali così da ottenere il modello reale sul quale effettuare l'identificazione. In questo caso, non avendo a disposizione dati sperimentali, si utilizzano dei dati noti in letteratura [2].

- $R = 0.1 \left[ \frac{\text{cmH}_2\text{O s}}{\text{L}} \right]$
- $L = 0.01 \left[ \frac{\text{cmH}_2\text{O s}^2}{\text{L}} \right]$
- $C = 0.1 \left[ \frac{\text{L}}{\text{cmH}_2\text{O}} \right]$

In questo modello semplificato si considerano un'unica resistenza e un'unica compliance che vanno a rappresentare la complessiva e la capacità di accumulare aria del sistema respiratorio nel suo complesso. Quindi la resistenza  $R$  porta con sé il contributo resistivo delle vie aeree, dei tessuti polmonari e della parete toracica. La capacità  $C$  rappresenta il compliance dei tessuti e della parete toracica.

L'obiettivo è la predizione della pressione alveolare  $p_A$  e della sua risposta dinamica a differenti forme d'onda di pressione applicare all'apertura delle vie aeree  $p_{ao}$ .

Applicando la prima legge di Kirchhoff al circuito in fig. 2 si ottiene:

$$p_{ao} = L \frac{di}{dt} + Ri + p_a \quad (1)$$

Dove vale il legame:

$$i = C \frac{dp_a}{dt} \quad (2)$$

Da cui si ricava la funzione di trasferimento:

$$H(s) = \frac{P_a(s)}{P_{ao}(s)} = \frac{1}{1 + sRC + s^2LC} \quad (3)$$

Questo mostra come  $p_a$  è interamente dipendente da  $p_{ao}$  e il sistema è un sistema con configurazione open-loop.

## 2.2 Identificazione dei parametri

Non avendo a disposizione dati paziente specifici sulla meccanica respiratoria i dati verranno generati tramite la procedura `rlc_fun.m` che restituisce la risposta del circuito RLC fornendogli in ingressi i parametri del sistema.

A questo può essere aggiunto anche del rumore.

## 2.3 Algoritmo di Nelder-Mead

Una volta assunto come modello del sistema la funzione di trasferimento di un circuito RLC, per stimare i parametri è necessario definire una funzione obiettivo. Tale funzione rappresenta la differenza tra la risposta misurata (vera) del sistema e quella stimata:

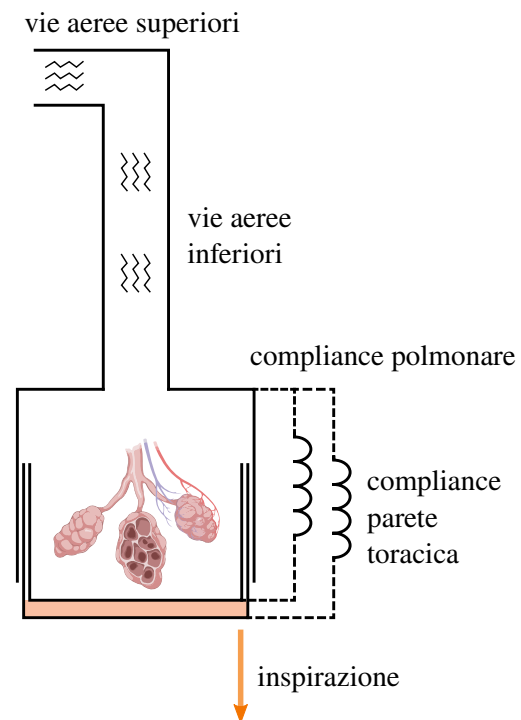


FIG. 1: Astrazione di polmoni, parete toracica e spazio pleurico

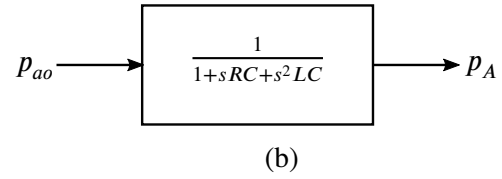
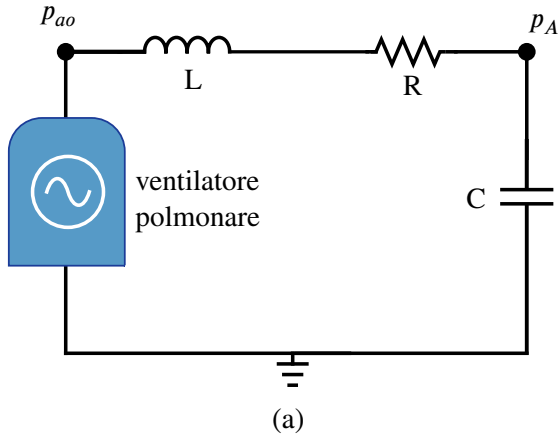


FIG. 2: Circuito RLC rappresentante la meccanica polmonare (a); sistema open loop rappresentante il circuito RLC (b)

$$\underline{e} = \underline{y}^{\text{pred}} - \underline{y}^{\text{mis}} \quad (4)$$

E la ricerca dei parametri corretti punta alla minimizzazione della funzione obiettivo definita come:

$$E = \frac{1}{2} \|\underline{e}\|^2 \quad (5)$$

Uno dei possibili modi per affrontare questo problema consiste nell'affidarsi alla routine di Matlab nota come `fminsearch()`. Tale procedura si occupa di restituire i parametri ottimali. In particolare, questa funzione sfrutta l'algoritmo di Nelder Mead [3]. Tale metodo non fa uso di derivate e si basa sul concetto di un semplice approssimando il punto di ottimo locale in  $n$  variabili.

## 2.4 Metodo di Gauss-Newton

Un'alternativa è il metodo di Gauss-Newton. Il metodo, diversamente dal precedente, richiede di portare in conto anche le derivate.

La minimizzazione dell'eq. (5) è un problema non lineare ai minimi quadrati. La soluzione numerica sfrutta il metodo di Gauss Newton. Come il metodo di Newton, si impone la stazionarietà, condizione che si riflette sulla  $j^{\text{ma}}$  componente:

$$(\nabla E)_j = \sum_k e_k \cdot \frac{\partial e_k}{\partial p_j} \quad (6)$$

Ovvero tale differenziazione corrisponde a premoltiplicare per la matrice di sensibilità  $J$ :

$$\nabla E = \underline{J}^T \underline{e} \quad (7)$$

Utilizzando quindi l'algoritmo di Newton l'iterazione al passo  $k + 1$  cercherà i parametri ( $\underline{p}$ ) come:

$$\underline{p}^{(k+1)} = \underline{p}^{(k)} - [\nabla(\nabla E)]^{-1} (\nabla E) \Big|_{\underline{p}^k} \quad (8)$$

Per l'approssimazione introdotta da Gauss [5] è possibile trascurare le componenti quadratiche dovute alla derivata seconda:

$$\begin{aligned} (\nabla^2 E)_{ij} &= \frac{\partial^2 E}{\partial p_j \partial p_i} = \frac{\partial}{\partial p_i} \left( e_k \frac{\partial e_k}{\partial p_j} \right) \\ &= \frac{\partial e_k}{\partial p_i} \frac{\partial e_k}{\partial p_j} + e_k \frac{\partial^2 e_k}{\partial p_j \partial p_i} \end{aligned} \quad (9)$$

Ovvero considerare l'iterazione semplicemente come:

$$\underline{p}^{(k+1)} = \underline{p}^{(k)} + \underline{h}^{(k)} \quad (10)$$

Dove l'incremento sarà:

$$\underline{h}^{(k)} = - \left[ \underline{J}^T \underline{J} \right]^{-1} \cdot \underline{J}^T \underline{e} \Big|_{\underline{p}} (k) \quad (11)$$

## 3 Metodi

Dunque il problema di identificazione parametrica viene affrontato come un problema di minimizzazione che porta a trovare i parametri ottimali. Sono quindi riportate le due differenti analisi.

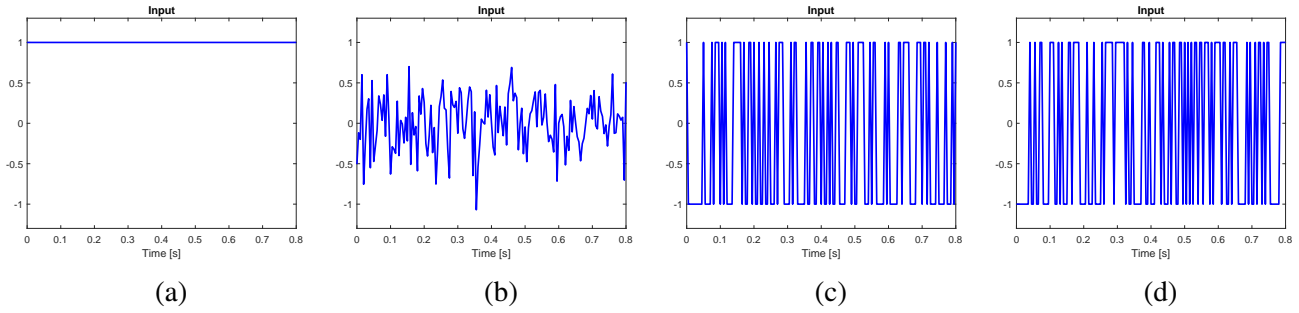


FIG. 3: Differenti ingressi utilizzati per stimolare il sistema RLC. Ingresso a gradino (a); ingresso di tipo binario random (b); segnale gaussiano random (c); segnale binario pseudorandomico (d). I segnali sono stati generati tramite il comando `idinput()` appositamente per l'identificazione di sistemi [4].

	Vero	Iniziale	Stimato	Errore relativo
$R$	0.100	0.150	0.119	0.197
$L$	0.010	0.008	0.012	0.206
$C$	0.100	0.280	0.083	0.169

TAB. 1: Risultati dell'ottimizzazione con `fminsearch()` considerando 3 parametri.

### 3.1 Minimizzazione

Nel primo metodo si utilizza la routine di Matlab `fminsearch()`. La risposta vera del sistema viene calcolata tramite la routine `rlc_fun.m` che si occupa, una volta fornitegli i coefficienti  $R$ ,  $L$ ,  $C$  e l'ingresso, la risposta del sistema dinamica. Una volta generati i dati corrispondenti alla risposta vera del sistema è possibile procedere all'ottimizzazione.

Si richiama allora la procedura `fminsearch()` fornendogli la funzione obiettivo, il punto iniziale e le variabili necessarie per il calcolo della funzione obiettivo (ingresso e discretizzazione temporale). La funzione obiettivo è presente all'interno della routine `onj_fun.m` e si occupa di fare la differenza tra l'uscita vera e quella predetta. L'uscita predetta viene calcolata ad ogni iterazione sfruttando la procedura di `rlc_fun.m` andando a calcolare la risposta del sistema RLC con i parametri aggiornati di volta in volta mediante l'algoritmo di Nelder-Mead.

Effettuando un primo test, con rumore gaussiano, è possibile stimare i parametri.

Inoltre, è possibile estrarre alcune informazioni dal processo di ottimizzazione. In particolare, si vede come nonostante la risposta del sistema viene stimata correttamente (fig. 4), l'errore sui parametri è piuttosto alto (Tab. 1).

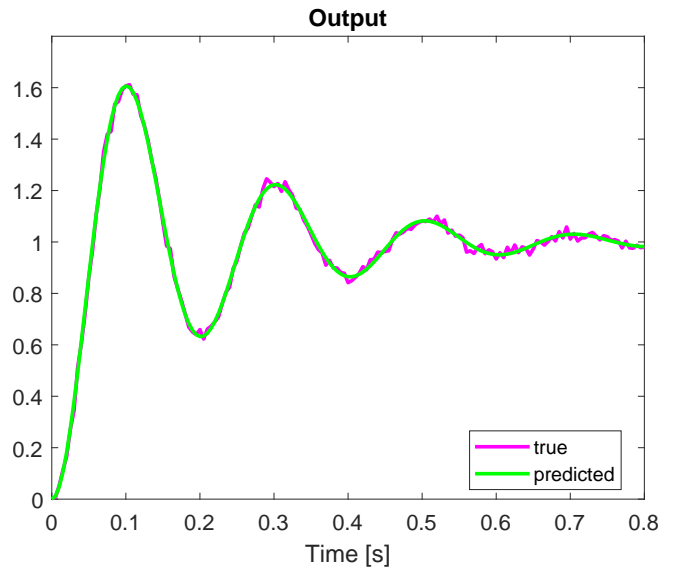


FIG. 4: Confronto tra la risposta vera (con aggiunta di rumore gaussiano) e la risposta stimata dopo aver ottimizzato i parametri (caso iniziale con 3 parametri). La risposta viene stimata correttamente e la procedura di stima va ad effettuare una sorta di filtraggio del rumore.

### 3.2 Non identificabilità strutturale

Il problema, per come è stato posto, presenta una formulazione sbagliata. Infatti, andando a calcolare la matrice di sensibilità si vedono alcuni problemi. La matrice di sensibilità non è altro che l'insieme delle derivate della misura  $n^{ma}$  rispetto al  $j^{mo}$  parametro. Calcolando quindi numericamente lo Jacobiano è possibile fare alcune considerazioni. Per calcolarlo è sufficiente perturbare rispetto al  $j^{mo}$  parametro e calcolarne la derivata come rapporto incrementale tra la risposta del sistema con ingresso perturbato e lo stato non perturbato rispetto la perturbazione:

$$J_i = \frac{y_{pert} - y_{ref}}{|pert_i|} \quad (12)$$

I risultati numerici mostrano come la matrice ha

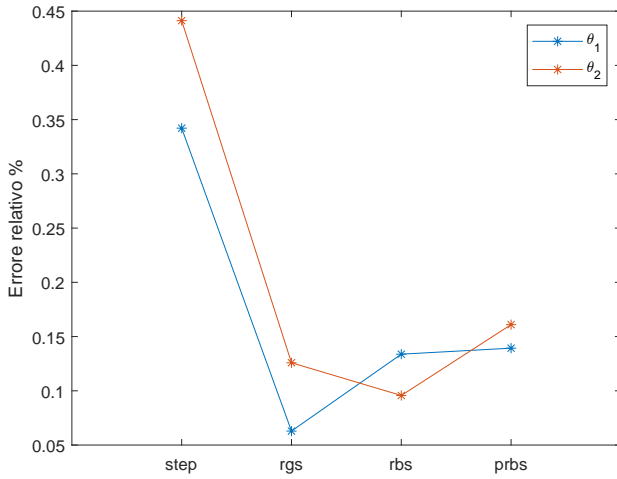


FIG. 5: Errore sui due parametri con i differenti segnali di ingresso utilizzati. Tra differenti ripetizioni dell'algoritmo l'errore potrebbe variare (sia a causa del metodo che della randomicità degli ingressi)

un numero di condizionamento molto alto  $1.85610^7$  ma sembrerebbe avere rango massimo. Tuttavia, andando ad indagare i valori singolari si vede come uno dei tre è in realtà molto piccolo, abbastanza da poter essere considerato numericamente nullo. Nonostante il software percepisce un numero diverso da zero questo è in realtà nullo ed indica come nel problema c'è una dipendenza tra i parametri.

Questo è legato al fatto che nella risposta di trasferimento in realtà i parametri sono soltanto due, dati dal prodotto dei tre valori numerici legati ai componenti circuitali:

$$\begin{aligned}\theta_1 &= LC \\ \theta_2 &= RC\end{aligned}\quad (13)$$

### 3.3 Parametri ottimi

Considerando quindi il problema nella sua versione ben condizionata è possibile procedere a stimare i parametri  $\theta_1$  e  $\theta_2$ . In particolare è possibile analizzare il risultato ottenuto considerando diversi ingressi, mostrati in fig. 3.

Per farlo viene riorganizzato il codice in modo da racchiudere la generazione dei dati e l'ottimizzazione in due procedure che possono essere richiamate di volta in volta con i differenti ingressi. Con tutti gli ingressi testati l'errore rimane sotto il 2% pur mantenendo una certa variabilità tra le differenti iterazioni.

### 3.4 Gauss Newton

Un approccio differente prevede l'utilizzo dell'algoritmo di Gauss Newton. Si cerca la minimizzazione della funzione obiettivo per il tramite dell'iterazione in eq. (5).

È necessario calcolare, per procedere lungo le iterazioni, lo Jacobiano. Questo viene calcolato tramite la routine `jacobian_fun.m`. Tale routine prende in ingresso il vettore dei parametri, l'ingresso e il vettore dei tempi e si occupa di calcolare la derivata numerica delle misure (risposta stimata) rispetto al parametro. Per farlo va a perturbare la  $j^{\text{ma}}$  componente della risposta facendo riferimento al  $j^{\text{mo}}$  parametro e ne calcola il rapporto incrementale. La procedura è analogo a quanto espresso per la matrice di sensibilità nell'eq. (12).

È possibile sfruttare gli stessi ingressi del caso precedente (fig. 3). I risultanti seguenti fanno riferimento ad un ingresso tipo gradino.

Per aiutare il metodo a convergere si aggiunge un parametro di rilassamento ( $\alpha$ ) che permette di modulare l'incremento tra un'iterazione e la successiva:

$$\underline{p}^{(k+1)} = \underline{p}^{(k)} + \alpha^{(k)} \underline{h}^{(k)} \quad (14)$$

Introducendo invece parametro di regolarizzazione  $\lambda > 0$  si ottiene sempre l'inversione di una matrice non singolare [5] e si evitano problemi di cattivo condizionamento.

$$\left[ \underline{J}^T \underline{J} \right]^{-1} \rightarrow \left[ \underline{J}^T \underline{J} + \lambda^{(k)} \underline{I} \right]^{-1} \quad (15)$$

Il problema ricade sulla scelta del parametro ottimo, se troppo grande può rovinare la velocità di convergenza del metodo mentre se troppo vicino allo zero può essere inefficiente e portare verso le singularità dello Jacobiano.

Fissando il numero di iterazioni massimo è possibile confrontare come varia la convergenza (tempo di convergenza) al variare dei parametri di rilassamento e di regolarizzazione (fig. 6). Si considera il tempo di convergenza come il numero di passi necessari ad avere una soluzione valida, soluzioni tali da non dare il risultato entro il numero di passi fissati vengono considerate non convergenti.

Per il parametro di regolarizzazione non si ottengono miglioramenti in quanto il problema è già ben condizionato. Per il parametro di rilassamento invece si vede fortemente la sua influenza sulla velocità di convergenza del metodo.

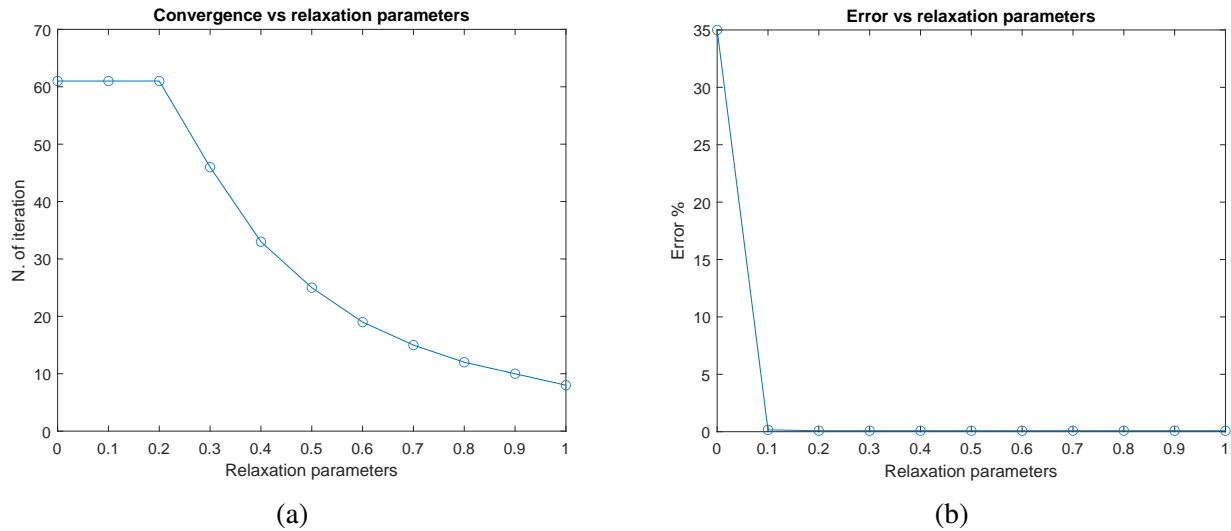


FIG. 6: Effetto del parametro di rilassamento sul metodo di Gauss-Newton. In (a) si vede come più il parametro è piccolo più il metodo converge lentamente.

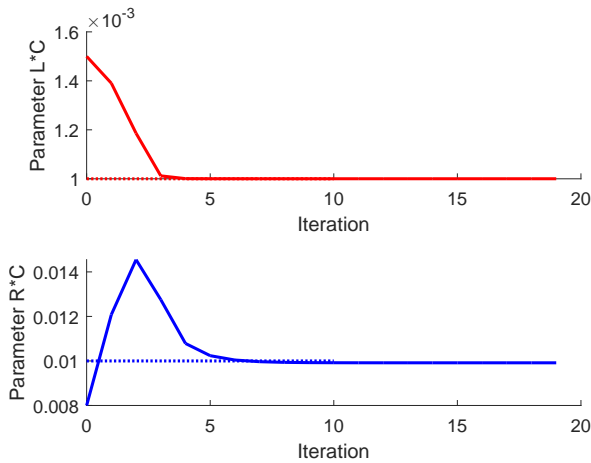


FIG. 7: Metodo di Gauss Newton arrivato a convergenza con 19 iterazioni,  $\alpha = 0.6$ ,  $\lambda = 0.1$ . In tratto continui i valori aggiornati durante le iterazioni, linea tratteggiata per i valori veri (obiettivo di minimizzazione).

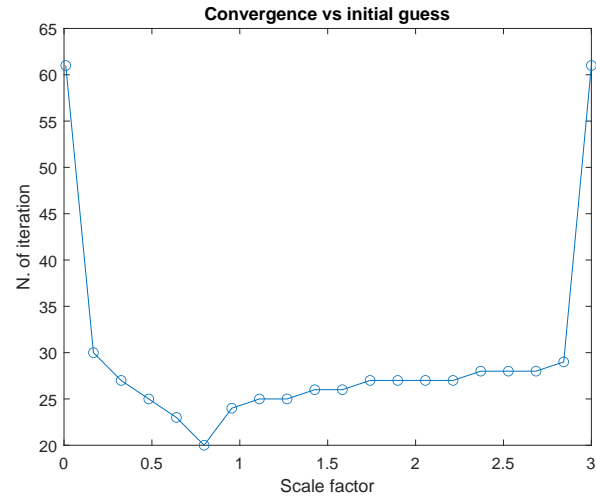


FIG. 8: Effetto della variazione del parametro iniziale sulla convergenza del metodo. Allontanandosi troppo dalla soluzione iniziale (scale factor=1) il metodo non converge.

Una notevole importanza è ricoperta anche dal punto iniziale. Le tecniche utilizzate finora considerano il punto iniziale come una quantità ottenuta scaldando i parametri reali con i seguenti fattori:

$$\begin{aligned}\theta_1^0 &= 1.5 \cdot \theta_1 \\ \theta_2^0 &= 0.8 \cdot \theta_2\end{aligned}\quad (16)$$

È possibile scalare ulteriormente questi valori e osservarne l'effetto. In particolare, scalando entrambi i parametri è evidente come allontanandosi troppo dalla soluzione vera il metodo non converge. I parametri vengono scalati di  $0.01 \div 3\times$  rispetto il valore iniziale precedentemente considerato.

## 4 Conclusioni

Entrambi i metodi sono validi e permettono di raggiungere il risultato stimando i parametri.

È importante considerare il problema nella forma corretta, considerando i parametri effettivamente linearmente indipendenti e non i differenti elementi circuitali.

Il metodo di Nelder-Mead permette di evitare la derivazione ma porta ad errori medi più elevati.

Il metodo di Gauss-Newton si mostra più preciso anche se può risultare computazionalmente più costoso.

Essendo entrambi metodi non lineari, risulta



molto importante scegliere il punto iniziale altrimenti il metodo potrebbe non convergere.

Inoltre, per il metodo di Gauss Newton risulta fondamentale selezionare correttamente i parametri di rilassamento e regolarizzazione per evitare una divergenza del metodo o un costo computazionale più elevato di quanto necessario.

## Disponibilità dei dati

Il materiale è disponibile alla repository online del progetto: <https://github.com/mastroalex/respiratory-mechanics>

## Codice

- Il codice per l'ottimizzazione con Nelder-Mead a tre parametri è presente nel file `par_sys_id.m`
- Il codice per l'ottimizzazione con Nelder-Mead a due parametri è presente nel file `par_sys_id_two_param.m`
- Il codice per l'ottimizzazione con Gauss-Newton è presente nel file `par_sys_id_gauss_newton.m`

## Riferimenti

- [1] Pardis Ghafarian, Hamidreza Jamaati, and Seyed Mohammadreza Hashemian. "A Review on Human Respiratory Modeling". en. In: (), p. 9.
- [2] Michael C. K. Khoo. *Physiological control systems: analysis, simulation, and estimation*. en. Second edition. IEEE Press series in biomedical engineering. Piscataway, NJ : Hoboken, New Jersey: IEEE Press ; Wiley, 2018. ISBN: 978-1-119-05533-4.
- [3] Jeffrey C. Lagarias et al. "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions". en. In: *SIAM Journal on Optimization* 9.1 (Jan. 1998), pp. 112–147. ISSN: 1052-6234, 1095-7189. DOI: 10.1137 / S1052623496303470. URL: <http://epubs.siam.org/doi/10.1137/S1052623496303470> (visited on 04/01/2022).
- [4] MathWorks Matlab. *Generate input signals to support system identification*. URL: <https://it.mathworks.com/help/ident/ref/idinput.html>.
- [5] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. 2nd ed. Springer series in operations research. OCLC: ocm68629100. New York: Springer, 2006. ISBN: 978-0-387-30303-1.