

**ALMA MATER STUDIORUM - UNIVERSITA' DI BOLOGNA**

SCUOLA D'INGEGNERIA E  
ARCHITETTURA

-Sede di Forlì-

CORSO DI LAUREA  
IN INGEGNERIA AEROSPAZIALE

Classe L-9

ELABORATO FINALE DI LAUREA

In Avionica e Strumentazione Spaziale

Analisi e simulazione di un sistema per la  
misura sperimentale del dipolo magnetico  
residuo di micro-satelliti

CANDIDATO  
Alessandro D'Anniballe

RELATORE  
Prof. Paolo Tortora

CORRELATORE  
Dr. Ing. Marco Zannoni

Anno Accademico 2013/2014

Sessione II



*Alla mia famiglia ed al mio paese*

*“Per aspera sic itur ad astra”*



# INDICE

<b>Elenco figure</b> .....	2
<b>Capitolo 1 Sommario</b> .....	3
<b>Capitolo 2 Introduzione</b> .....	4
2.1 Disturbi d'assetto e dipolo magnetico residuo.....	4
2.2 Il micro-satellite ESEO.....	6
2.3 Gabbia di Helmholtz.....	8
<b>Capitolo 3 Studio effettuato</b> .....	9
3.1 Introduzione.....	9
3.2 Magnetometri.....	9
3.3 Determinazione del dipolo residuo.....	11
3.3.1 Modello matematico.....	11
3.3.2 Script implementato.....	12
3.3.3 Posizioni dei magnetometri.....	17
3.3.4 Rumori considerati ed errori valutati.....	18
3.3.5 Dipendenza dal numero di misure effettuate.....	27
3.4 Valutazione dei risultati ottenuti.....	29
<b>Capitolo 4 Conclusioni</b> .....	30
<b>Ringraziamenti</b> .....	32
<b>Bibliografia</b> .....	33
<b>Appendice A Script e functions</b> .....	34
A.1 Algoritmo_ESEO.m.....	34
A.2 fmincon_norm.m.....	43
A.3 progress_bar.m.....	43
A.4 b_field_dipole.m.....	44
A.5 remove_outliers.m.....	44
<b>Appendice B : Datasheet magnetometro AP539</b> .....	46

## Elenco figure

- 2.1 Coppie di disturbo su di un satellite in orbita attorno alla Terra
- 2.2 Dipolo magnetico
- 2.3 Schema di stima del RMM
- 2.4 La sonda ESEO
- 2.5 Payloads di ESEO
- 2.6 Bobine di Helmholtz
- 2.7 Gabbia di Helmholtz
- 3.1 Magnetometro AP539
- 3.2 Linee di campo magnetico
- 3.2 Intensità del dipolo residuo
- 3.4 Configurazioni di magnetometri con rispettive distanze (passi) tra di essi
- 3.5 Errori di stima in funzione del rumore dei magnetometri
- 3.6 Errori di stima in funzione delle posizioni dei magnetometri
- 3.7 Errori di orientazione ed intensità per una configurazione 1 più distante dal satellite
- 3.8 Rotazione del sistema assi magnetometro rispetto al SdR del satellite
- 3.9 Rotazione asse/angolo di Eulero
- 3.10 Errori di stima in funzione dell'orientazione dei magnetometri
- 3.11 Errori di stima in funzione dei bias di calibrazione
- 3.12 Errori di stima in funzione dei fattori di scala
- 3.13 Errori di stima in funzione del rumore intrinseco per la configurazione 4, effettuando 5 misure

# CAPITOLO 1

## Sommario

*La dinamica dell'assetto di un satellite artificiale rappresenta uno degli aspetti più delicati della missione che esso stesso andrà a svolgere in orbita attorno ad un qualche corpo celeste, quale appunto il pianeta Terra.*

*Il seguente lavoro di tesi si propone di analizzare la causa di una delle principali componenti di disturbo dell'assetto appena menzionato, preponderante per satelliti dalle piccole dimensioni, fornendo la spiegazione, validata attraverso una simulazione, della messa a punto di un metodo sperimentale per la valutazione della stessa. La componente in questione è la coppia di disturbo magnetica, ed è generata dall'interazione tra il campo magnetico terrestre ed il cosiddetto 'dipolo magnetico residuo' del satellite stesso, ossia quel campo magnetico che esso, in modalità operativa e non, risulta generare a causa del materiale ferromagnetico presente al suo interno, e delle correnti elettriche circolanti nei vari cavi conduttori. Ci si è dunque occupati dell'analisi e messa a punto di un metodo che possa consentire sperimentalmente di rilevare l'entità del dipolo residuo.*

*Il lavoro di simulazione è stato svolto prendendo in considerazione le dimensioni e le possibili caratteristiche del dipolo residuo del micro-satellite ESEO (European Student Earth Orbiter), sviluppato da studenti di diverse università europee ed ora in fase di progetto dettagliato (fase C) presso i laboratori dell'azienda ALMASpace S.r.l. di Forlì.*

*Il metodo in esame consiste nel rilevare il campo magnetico generato dal satellite, posto all'interno di un sistema tridimensionale di bobine di Helmholtz per avere una zona libera da campi magnetici esterni. Il rilevamento del dipolo avviene per mezzo di un magnetometro a tre assi, e dalla suddetta misura si può pervenire alla conoscenza delle componenti del dipolo stesso, quali posizione, orientamento ed intensità; siccome però la misura del magnetometro non è ideale, ma risulta affetta da errori, per una più corretta caratterizzazione del dipolo è necessario utilizzare un numero maggiore di magnetometri (oppure, il che è lo stesso, un unico magnetometro spostato mano a mano) in punti diversi attorno al satellite in modo da avere più misure di campo magnetico e poter così sfruttare una procedura numerica di ottimizzazione per risalire alle componenti del dipolo.*

*Questa intera parte di calcolo è stata realizzata in MatLab<sup>®</sup>, simulando quindi le misure ottenute dai magnetometri, 'sporcandole' con i predetti errori, ed utilizzando le funzioni di minimizzazione lsqnonlin ed fmincon per verificare la funzionalità del sistema; si sono infatti analizzati i grafici rappresentanti i livelli di errore commessi dall' algoritmo di stima sulle varie componenti del dipolo, per le tipologie di errore dei magnetometri menzionate in precedenza.*

*Si è così cercato di suggerire una configurazione ottimale di magnetometri in grado di fornire una stima caratterizzata da un buon compromesso tra : numero di magnetometri da utilizzare non troppo elevato ed errore derivante accettabile.*

# CAPITOLO 2

## Introduzione

### 2.1 Disturbi d'assetto e dipolo magnetico residuo

In orbita bassa terrestre, un satellite artificiale si trova sottoposto a forze di varia natura che tendono a generare sullo stesso coppie di momento, le quali risultano in una modifica, nella maggior parte delle situazioni indesiderata, del suo assetto.

La corretta orientazione di payloads come i dispositivi per effettuare fotografie della Terra è infatti un requisito di imprescindibile importanza per la buona riuscita della missione.

Questi disturbi esterni sono identificabili principalmente in 4 tipologie :

- gradiente gravitazionale
- magnetismo interno al satellite
- pressione di radiazione solare
- resistenza aerodinamica sbilanciata

e sono funzione del profilo di missione, del tipo di orbita e della geometria del satellite [1].

L'immagine mostra l'andamento tipico dell'intensità della coppia di disturbo relativa ai casi prima elencati, in funzione dell'altitudine dell'orbita :

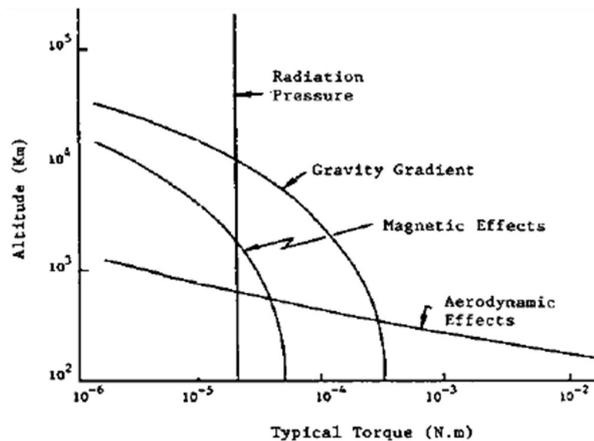
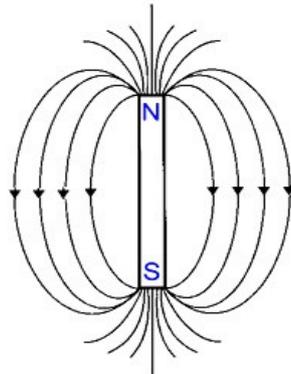


Fig. 2.1 Coppie di disturbo su di un satellite in orbita attorno alla Terra

Quella dovuta ad effetti magnetici può risultare essere la principale coppia di disturbo per piccoli satelliti, soprattutto per nano-satelliti con determinate configurazioni, per cui risulta necessaria un'opportuna valutazione della sua entità e della sua origine.

Questa coppia di disturbo è dovuta all'interazione tra il campo magnetico terrestre ed un indesiderato magnetismo residuo del satellite, che può essere modellato come un singolo dipolo magnetico, e quindi schematizzato da un vettore, il quale avrà una certa posizione, orientazione ed intensità all'interno del satellite stesso, ed è rappresentabile come nella figura seguente :



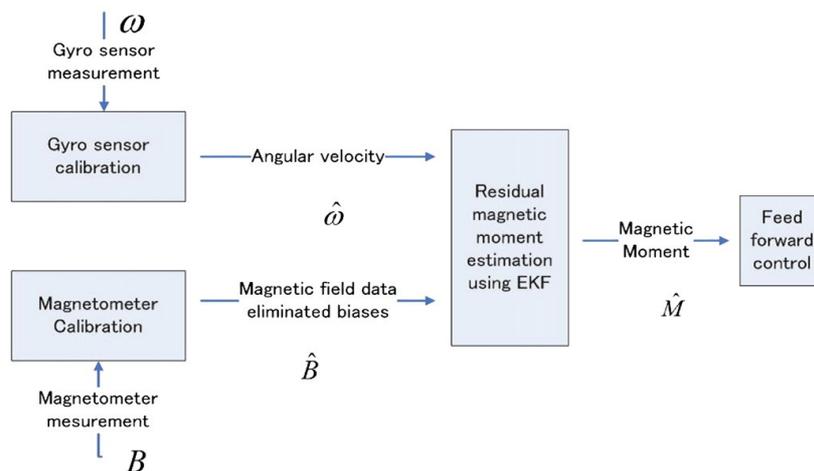
**Fig. 2.2 Dipolo magnetico**

La coppia di momento magnetico che si genera dall'interazione dei due campi magnetici è calcolabile come :

$$T_m = \vec{m} \times \vec{B}$$

dove  $B$  è il vettore del campo magnetico terrestre locale, misurato in Tesla (SI) o Gauss (CGS), ed  $m$  è il vettore del dipolo residuo del satellite, misurato in  $A \cdot m^2$  (SI). Tipici valori sono  $B = 3 \cdot 10^{-5}$  T,  $m = 0,1 A \cdot m^2$ , e  $T_m = 3 \cdot 10^{-6}$  N·m.

Il momento magnetico residuo (RMM) può essere misurato in orbita tramite un 'filtro di Kalman esteso' (Extended Kalman Filter), il quale può essere visto come un'implementazione ricorsiva del metodo dei minimi quadrati di Gauss non lineare [2].



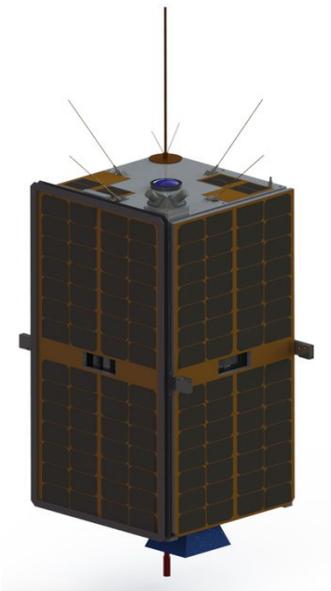
**Fig. 2.3 Schema di stima del RMM [3]**

Il sistema richiede in ingresso informazioni sul campo geomagnetico e sulla velocità angolare del satellite, le quali sono ottenute dai magnetometro e dai sensori giroscopici rispettivamente. Essendo queste misure affette da errori di bias e fattori di scala, si ha bisogno di una stima ai minimi quadrati per stimare con precisione il RMM, passaggio che viene realizzato dall'EKF appunto.

## 2.2 Il micro-satellite ESEO

Il satellite utilizzato come riferimento in questo lavoro di tesi è stato il micro-satellite European Student Earth Orbiter (ESEO) dell'European Space Agency (ESA), un progetto educativo per studenti provenienti da varie università europee, ed ora in fase di progetto avanzato in Italia, nell'azienda ALMA Space, sita in Forlì (FC).

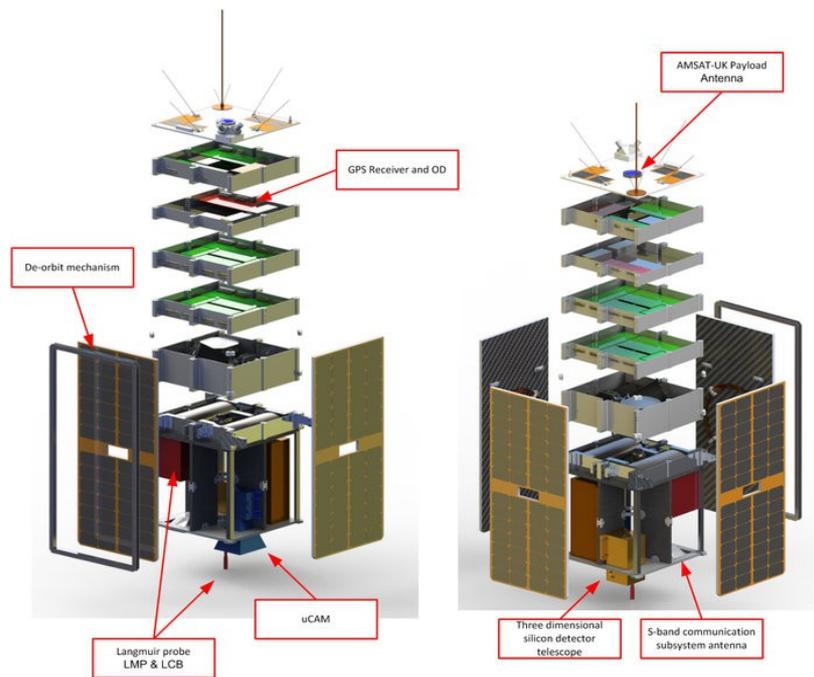
ESEO, mostrato in Fig.2.2, è la terza missione dell'Education Satellite Programme di ESA, e si basa dall'esperienza guadagnata con missioni quali SSETI Express (lanciata nel 2005) e YES2 (lanciata nel 2005) . Il lancio in Low Earth Orbit (LEO) è previsto nel 2015-2016.



**Fig. 2.4** La sonda ESEO

Gli obiettivi della missione saranno quelli di :

- Acquisire fotografie della Terra e/o di altri corpi celesti per scopi educativi; ESEO sarà infatti dotato di una microcamera sviluppata dall'istituto spaziale danese DTU Space.
- Fornire misure dei livelli di radiazione in orbita bassa terrestre; una sonda Langmuir e un dosimetro Tritel a tre assi misureranno la dose assorbita di radiazioni, la dose equivalente e gli spettri LET della radiazione cosmica). Questi strumenti sono stati sviluppati dall'Università di Budapest, Ungheria.
- Acquisire esperienza tecnologica per eventuali future missioni educative ESA; ESEO testerà un sistema di comunicazione in banda S costruito dal Politecnico di Wroclaw, un ricevitore GPS per la determinazione dell'orbita sviluppato dall'Università di Bologna, un esperimento software AODCS coordinato dalla Delft University ed un payload di de-orbiting costruito dalla Cranfield University.



**Fig. 2.5 Payloads di ESEO**

Il satellite sarà di forma parallelepipedica, con altezza 63cm e base quadrata di 33x33cm, e peserà all'incirca 40kg.

La durata prevista della missione è di 6 mesi, con la possibilità di una sua successiva estensione. L'orbita sarà eliosincrona, il che significa che ESEO attraverserà l'equatore ogni orbita a circa la stessa ora solare locale, e la sua altitudine massima non sarà superiore a 600 km. Alla fine della sua missione, dispiegherà la vela di resistenza fornita dalla Cranfield University, UK. Ciò farà sì che ESEO rientri e bruci in atmosfera terrestre, rispettando così il codice europeo di condotta dei detriti spaziali [4].

Dal momento che uno degli obiettivi di ESEO è proprio quello di acquisire fotografie del nostro pianeta, si capisce quale sia l'importanza di poter mantenere stabile il suo assetto in orbita, e di qui la necessità di conoscere in che modo le caratteristiche del satellite stesso possano influire sull'evoluzione del suo comportamento attorno al centro di massa.

### 2.3 Gabbia di Helmholtz

Per testare le proprietà magnetiche di un satellite si necessita di un ambiente 'magneticamente pulito', ossia con campo magnetico del valore più basso possibile, in modo da poter rilevare esclusivamente le caratteristiche magnetiche del satellite stesso.

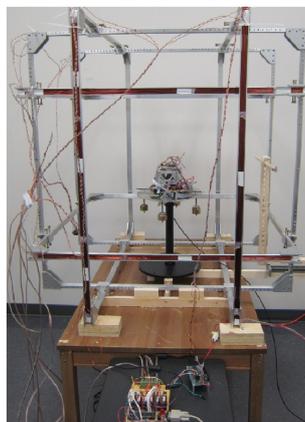
Una coppia di 'bobine di Helmholtz', ossia avvolgimenti identici di filo conduttore circolari o quadrati, attraversati dalla stessa corrente continua (nella stessa direzione) e posizionati simmetricamente l'uno di fronte all'altro, è in grado di generare nel proprio centro un campo magnetico caratterizzato da un'elevata uniformità spaziale; nel caso di bobine circolari, esse devono essere poste ad una distanza reciproca pari al raggio delle stesse [5].



**Fig. 2.6 Bobine di Helmholtz**

Al fine di poter ruotare il vettore campo magnetico, sono necessarie tre coppie di bobine posizionate solidalmente con gli assi di un sistema di riferimento cartesiano, in modo che ogni coppia generi un campo magnetico variabile in modulo e verso ma fisso nella direzione di riferimento per quella coppia (x,y o z) [6].

La condizione descritta ad inizio paragrafo può quindi essere ottenuta all'interno di un simulatore tridimensionale di campo magnetico costituito dalle predette coppie di bobine, che prende comunemente il nome di 'gabbia di Helmholtz' (Helmholtz cage); azionato nel modo opportuno, questo simulatore sarà in grado di generare un campo magnetico di valore pressoché uguale ma opposto al campo magnetico terrestre rilevato nella zona in cui si sta effettuando la sperimentazione, riuscendo così ad avere al suo interno uno spazio esente da campi magnetici esterni. La figura seguente mostra un esempio realizzativo della gabbia di Helmholtz :



**Fig. 2.7 Gabbia di Helmholtz**

## CAPITOLO 3

### Studio effettuato

#### 3.1 Introduzione

Definita la configurazione satellite-gabbia di Helmholtz si può passare ad analizzare in cosa consiste il lavoro da effettuare per arrivare alla conoscenza del dipolo magnetico residuo del satellite.

Una volta fornita una spiegazione sui magnetometri, strumenti da utilizzare per effettuare la rilevazione, e sugli errori che possono caratterizzare le loro misure, sarà esposta la procedura effettuata nella simulazione.

#### 3.2 Magnetometri

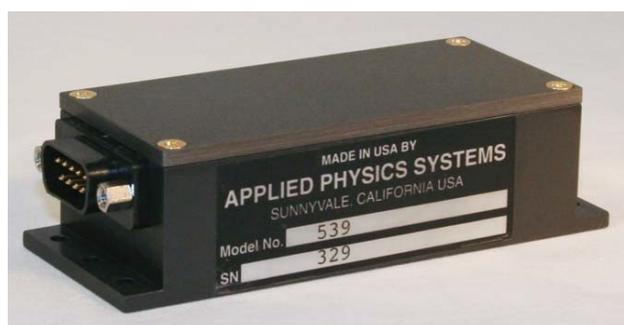
Un magnetometro è uno strumento che serve a misurare la direzione e l'intensità di un campo magnetico, ed in particolare del campo magnetico terrestre [7].

Esso può essere sostanzialmente di due tipologie :

- scalare - in grado di misurare esclusivamente il modulo del campo magnetico
- vettoriale - in grado di misurare la componente del campo magnetico lungo una o più particolari direzioni dello spazio

La lettura effettuata dal magnetometro può inoltre essere analogica o digitale.

I possibili magnetometri che saranno utilizzati su ESEO sono degli AP539, vettoriali a 3 assi, digitali, caratterizzati da un rumore intrinseco di circa 0.3nT, con scheda tecnica completa riportata in appendice B :



**Fig. 3.1** Magnetometro AP539

Nella simulazione sono stati considerati magnetometri di questo tipo, anche se sono poi stati utilizzati più livelli di rumore e di coefficienti di calibrazione, ma comunque nell'intorno di quello caratteristico dell'AP539.

Uno strumento del genere è quindi in grado di rilevare l'entità di un campo magnetico esterno lungo la direzione dei suoi tre assi (assi magnetometro), per cui fornirà in uscita i valori : Bx, By, Bz; allineando i suoi assi con quelli del SdR del corpo del quale si sta misurando il campo magnetico, i tre valori rappresenteranno dunque le sue componenti nel punto nel quale il magnetometro è posizionato.

La misura risulterà però normalmente affetta da errori caratteristici del magnetometro, quali il suo rumore intrinseco ed una sua calibrazione magari non perfetta, e da errori applicativi, quali errori di posizionamento ed orientazione del magnetometro stesso.

Il rumore intrinseco è variabile in modo random nel tempo attorno al valore nullo, con una deviazione standard rappresentata dal valore riportato nei datasheet, che nel caso del magnetometro AP539 è, come già detto, di 0.3nT.

I possibili errori collegati ad una non corretta calibrazione del magnetometro sono :

- Bias sulle letture dei rispettivi assi del magnetometro
- Fattori di scala dei singoli assi del magnetometro
- Non ortogonalità degli assi
- Rotazione rigida degli assi rispetto alla struttura del magnetometro

Essi agiscono complessivamente discostando la misura del campo magnetico dal valore reale, nel seguente modo :

$$\bar{B} = \begin{bmatrix} \bar{B}_x \\ \bar{B}_y \\ \bar{B}_z \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + \begin{bmatrix} X_s & XY & XZ \\ YX & Y_s & YZ \\ ZX & ZY & Z_s \end{bmatrix} \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} = E_0 + E_{art} B$$

Dove B è il valore del campo magnetico che si sta rilevando, E<sub>0</sub> è il vettore dei bias ed E<sub>art</sub> è una matrice non simmetrica la quale, tramite la decomposizione polare, può essere espressa come :

$$E_{art} = Q S$$

Dove :

- S è una matrice simmetrica, i cui elementi sulla diagonale principale rappresentano i fattori di scala dei singoli assi, e gli elementi al di fuori di essa sono correlati alla non ortogonalità degli assi stessi
- Q è una matrice ortogonale che rappresenta il disallineamento “rigido” del magnetometro

[8]

### 3.3 Determinazione del dipolo residuo

La procedura per la determinazione del dipolo residuo del satellite è di seguito descritta; in questo lavoro è stata effettuata una simulazione del procedimento sperimentale reale : il dipolo è stato ipotizzato come già noto, in modo da poter generare le misure dei magnetometri tramite un modello matematico, che sono state poi utilizzate per il procedimento numerico di stima (che sarà la parte da utilizzare nel lavoro sperimentale) dei parametri caratteristici del dipolo.

#### 3.3.1 Modello matematico

Di seguito è presentato il modello matematico utilizzato per simulare il dipolo magnetico residuo e poi nella procedura di ottimizzazione.

Il campo magnetico generato da un dipolo nello spazio libero, e rilevato nel punto  $(x,y,z)$ , ha la seguente espressione :

$$\vec{B}(x, y, z) = \frac{\mu_0}{4\pi} \left[ \frac{(3\hat{r}(\hat{r}\cdot\vec{m}) - \vec{m})}{|\vec{r}|^3} \right]$$

Dove  $\mathbf{B}$  è il vettore campo magnetico misurato in una certa posizione  $(x,y,z)$ , che sarebbe quella del magnetometro,  $\mu_0$  è la costante di permeabilità magnetica del vuoto ( $\mu_0 = 4\pi \cdot 10^{-7}$  H/m),  $\mathbf{m}$  è il vettore momento di dipolo magnetico, ed  $\mathbf{r}$  è il vettore che va dalla posizione del dipolo a  $(x,y,z)$ .

Dalla manipolazione di questa formula si ottengono le espressioni delle componenti ( $B_x, B_y, B_z$ ) del campo magnetico  $\mathbf{B}$  generato da un dipolo situato nel punto  $(a,b,c)$  e misurato in  $(x,y,z)$  :

$$B_x = \frac{\mu_0}{4\pi} \left[ \frac{(3[M(x-a) + N(y-b) + P(z-c)](x-a))}{R^5} - \frac{M}{R^3} \right]$$

$$B_y = \frac{\mu_0}{4\pi} \left[ \frac{(3[M(x-a) + N(y-b) + P(z-c)](y-b))}{R^5} - \frac{N}{R^3} \right]$$

$$B_z = \frac{\mu_0}{4\pi} \left[ \frac{(3[M(x-a) + N(y-b) + P(z-c)](z-c))}{R^5} - \frac{P}{R^3} \right]$$

$$R = \sqrt{(x-a)^2 + (y-b)^2 + (z-c)^2}$$

dove  $R$  rappresenta quindi la distanza dipolo-magnetometro, ed  $M, N, P$  sono le componenti del vettore dipolo [9].

Queste equazioni ( $B_x, B_y, B_z$ ) sono caratterizzate dall'aver un alto grado di non linearità, ossia le loro radici non potrebbero in genere essere espresse esplicitamente e, anche se ciò fosse possibile, l'espressione si presenterebbe in forma talmente complicata da risultare praticamente inutilizzabile. Per risolverle ci si ritrova quindi obbligati ad utilizzare metodi numeri iterativi, che cioè partono da approssimazioni della radice e producono una successione ad essa convergente.

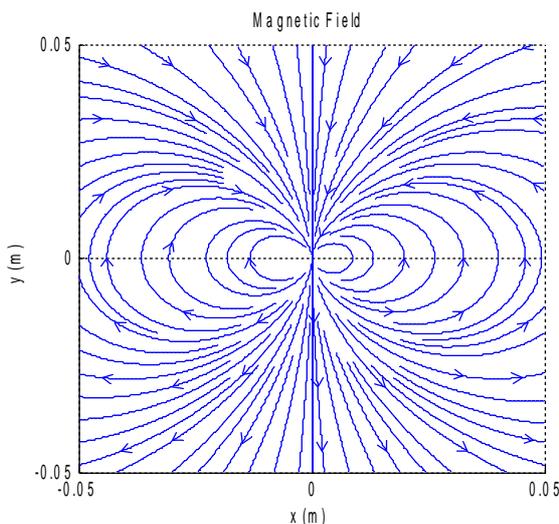
### 3.3.2 Script implementato

Dalle equazioni appena mostrate, una volta effettuata la misura del campo magnetico generato dal dipolo tramite i magnetometri, si potrebbero ricavare direttamente le componenti del dipolo magnetico, come detto con un procedimento numerico, data la non linearità; a causa però degli errori prima menzionati dai quali risulta affetta la misura, risulta necessario effettuarne più di una, in varie posizioni attorno al satellite, ed utilizzare una procedura di ottimizzazione, la quale tenti di ridurre l'errore e fornire così valori delle componenti del dipolo il più possibile vicini a quelli reali.

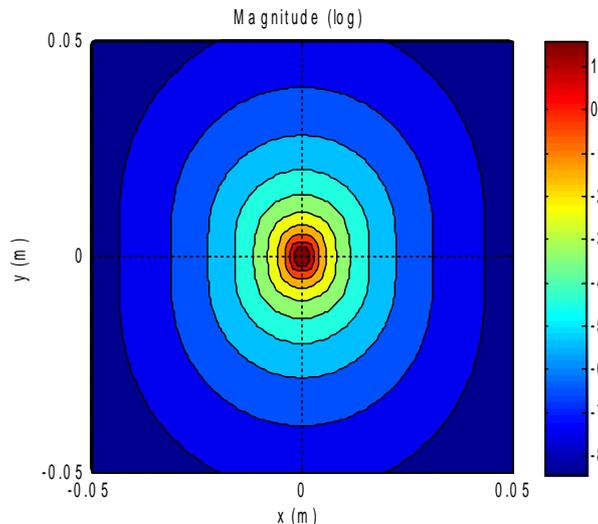
Durante il lavoro di tirocinio è stato a questo scopo realizzato uno script in ambiente MatLab® (considerando però un nano-satellite, ma ciò non cambia la validità della procedura), nel quale si sono innanzitutto simulate le misure dei magnetometri, utilizzando :

- Le formule viste nel paragrafo precedente.
- Un dipolo magnetico residuo simulato noto, con posizione  $(a,b,c) = (0,0,0)m$  e componenti del vettore intensità  $(M,N,P) = (0,-0.05,0)A \cdot m^2$ .
- Le posizioni dei magnetometri rispetto al centro del satellite (considerato come centro del SdR assi corpo dello stesso), disposti in 5 schemi differenti attorno allo stesso.

Le linee del campo magnetico generato dal dipolo sul piano x-y e la sua intensità sono riportate rispettivamente in Fig.3.2 e Fig.3.3 :



**Fig. 3.2** Linee di campo magnetico



**Fig. 3.3** Intensità del dipolo residuo

Si sono così ottenuti i valori ideali delle componenti  $B_x\_meas$ ,  $B_y\_meas$  e  $B_z\_meas$  del campo magnetico  $B$  generato dal dipolo residuo del satellite.

Queste misure sono quindi state 'sporcate' con il rumore dei magnetometri, modellizzato come un processo stocastico gaussiano bianco, con media nulla e deviazioni standard 'sigma\_noise' andanti da 0 a  $0.5\mu T$ , con incrementi di  $0.1\mu T$  :

```
noise_mag = sigmas_mag(qm)*randn(length(Bx_meas),3)
```

questo valore va sommato direttamente ai Bx\_meas, By\_meas, Bz\_meas, ottenendo i Bx\_meas\_noise, By\_meas\_noise e Bz\_meas\_noise.

A questo punto è stata sviluppata la parte di ottimizzazione : è stata utilizzata la funzione 'lsqnonlin' [10] (least squares non linear), facente parte dell'optimization toolbox; essa si basa sul metodo non lineare di approssimazione ai minimi quadrati, che consiste nel minimizzare la funzione costituita dalla somma quadratica delle differenze tra i valori di campo magnetico simulati ed i valori misurati, affetti da errore :

$$\min_x \|f(x)\|_2^2 = \min_x (f_1(x)^2 + f_2(x)^2 + \dots + f_n(x)^2)$$

Per valori simulati si intendono in questo caso i Bx\_meas, By\_meas, Bz\_meas, calcolati con i valori iniziali di stima dei parametri caratteristici del dipolo, dai quali partirà la minimizzazione, cercando di avvicinarli ai valori reali, mediante appunto il confronto con le misure errate; le f<sub>i</sub>(x) della formula precedente rappresenteranno perciò dalle differenze Bi\_guess – Bi\_meas (i = x,y,z), che vanno sotto il nome di 'residui'; introdotte in un ciclo for sulla lunghezza del vettore 'Pos', contenente le coordinate di ciascuna posizione dei magnetometri; si è costruita così la funzione vettoriale F, avente tre componenti (x,y,z), calcolati per ogni posizione :

```
for w = 1:length(Bx_meas_noise)

R1 = sqrt((Pos(w,1)-g(1))^2+(Pos(w,2)-g(2))^2+(Pos(w,3)-g(3))^2);

F = [F; ((mu0/(4*pi))*3*(g(4)*(Pos(w,1)-g(1))+g(5)*(Pos(w,2)-g(2))
+g(6)*(Pos(w,3)-g(3)))*(Pos(w,1)-g(1))/R1^5-g(4)/R1^3)-
Bx_meas_noise(w))/B0;
((mu0/(4*pi))*3*(g(4)*(Pos(w,1)-g(1))+g(5)*(Pos(w,2)-g(2))
+g(6)*(Pos(w,3)-g(3)))*(Pos(w,2)-g(2))/R1^5-g(5)/R1^3)-
By_meas_noise(w))/B0;
((mu0/(4*pi))*3*(g(4)*(Pos(w,1)-g(1))+g(5)*(Pos(w,2)-g(2))
+g(6)*(Pos(w,3)-g(3)))*(Pos(w,3)-g(3))/R1^5-g(6)/R1^3)-
Bz_meas_noise(w))/B0];

end
```

Il vettore g = [g1, g2, g3, g4, g5, g6] contiene i valori iniziali di stima [a, b, c, M, N, P] dai quali parte la minimizzazione. Il valore B0 è il valore massimo tra i B\_meas\_noise, utilizzato per 'normalizzare' i residui della F, cioè farli diventare di un ordine di grandezza maggiore, dato che sono dell'ordine dei micro Tesla (µT), per essere confrontabili con la tolleranza di default di lsqnonlin (1e-6).

In questo modo, dato che le misure dei magnetometri sono 'quasi esatte', a meno dell'errore considerato, l'algoritmo cerca di avvicinarle ai valori reali, e da esse ricava le variabili delle componenti della F, che in questo caso sono appunto le componenti del dipolo che si vogliono conoscere.

La parte di script nella quale viene richiamata `lsqnonlin` è :

```
[g, resnorm, residual, exitflag] =  
lsqnonlin('funzione_minimizzazione', g0, lb, ub);
```

Qui si notano quali sono gli input ed output di `lsqnonlin` :

INPUT :

- `g0` è il vettore contenente le componenti del dipolo da cui parte la stima; esse sono state prese in modo random in ciascuna prova, in particolare la posizione del dipolo entro il volume del satellite e le componenti del vettore dipolo tra  $-0.1A \cdot m^2$  e  $0.1A \cdot m^2$  (valori ritenuti già abbastanza elevati per un nano-satellite)
- `'funzione_minimizzazione'` è la function contenente la  $F$  esposta prima, i cui elementi vengono minimizzati da `lsqnonlin`; essa è definita come segue :  
$$F = \text{funzione\_minimizzazione}(g)$$
dove  $g$  è il vettore input della funzione, che contiene i parametri che si vogliono determinare;
- `lb` ed `ub` sono i limiti di ricerca dell'algorithmo di minimizzazione, il quale deve fare in modo che sia  $lb \leq g \leq ub$ ; in questo caso i vincoli sono le dimensioni del satellite per quanto riguarda  $a, b, c$  ed un valore minimo di  $-0.1A \cdot m^2$  e massimo di  $+0.1A \cdot m^2$  per quanto riguarda le componenti del vettore di dipolo.

OUTPUT :

- `g` è il vettore che permette di minimizzare la funzione  $F$ , e quindi la soluzione del problema;
- `resnorm` è la norma-2 dei residui finali;
- `residual` fornisce i valori dei residui finali della  $F$
- `exitflag` è un parametro che indica la condizione di uscita dell'algorithmo.

`lsqnonlin` può utilizzare diversi algoritmi per l'ottimizzazione; quello usato di default è detto 'trust-region-reflective'.

Per ciascuna configurazione e fonte di errore, la procedura è stata ripetuta 100 volte per caratterizzare meglio i risultati della procedura, dato che gli errori sono stati considerati randomici, con distribuzione più o meno uniforme.

Anche imponendo dei vincoli a `lsqnonlin`, se il volume del satellite è abbastanza elevato, si è osservato che l'algorithmo, in fase di minimizzazione, può convergere verso dei minimi locali anziché globali per quanto riguarda la posizione del dipolo, fornendo così delle soluzioni errate; si è quindi provveduto ad eliminare (durante il lavoro di tesi) le soluzioni troppo differenti dalla media verso cui convergevano la maggior parte delle altre, tramite la funzione costruita 'remove outliers' : essa controlla singolarmente gli elementi della  $g$  in output a `lsqnonlin` (comprese le componenti del vettore dipolo) e, se la distanza dalla media è troppo elevata dalla deviazione standard, ad esempio  $4\sigma$ , li rimuove dal calcolo degli errori medi.

Nel lavoro di tesi è stato ripreso lo stesso script, adattando però le dimensioni del satellite a quelle di ESEO e considerando deviazioni standard del rumore e dei coefficienti di calibrazione di magnetometri simili al tipo AP539; il dipolo residuo simulato è stato lo stesso.

Il lavoro effettuato è stato quello di considerare le altre fonti di errore che possono influenzare le misure dei magnetometri, quali un loro posizionamento ed una loro orientazione non del tutto corretti, ed una non ottima calibrazione degli stessi; di quest'ultima si sono studiati gli effetti di un bias di calibrazione e dei fattori di scala, trascurando quindi gli effetti di non ortogonalità e di rotazione rigida dello stesso sistema di assi, in quanto viene già considerata una rotazione dell'intero magnetometro. Si è cercato quindi di individuare una configurazione di magnetometri ottimale, dal punto di vista del numero degli stessi e degli errori generati, la quale potrebbe essere presa in considerazione per un lavoro sperimentale di questo genere.

È stato osservato che la funzione 'fmincon' [11] (find minimum constrained) dell'optimization toolbox risulta meno affetta da problemi di convergenza verso minimi locali rispetto a lsqnonlin, fornendo quindi una quantità minore di outliers, per cui è stata utilizzata essa per migliorare la procedura di ottimizzazione. fmincon ha l'obiettivo di cercare il minimo vincolato di una funzione scalare di più variabili, partendo da una stima iniziale; con essa è stato utilizzato l'algoritmo 'interior point'.

Lsqnonlin minimizza la somma quadratica degli elementi della funzione vettoriale ad essa passati, quindi per ogni misura dei magnetometri minimizza i corrispondenti residui su x,y e z, mentre fmincon minimizza una funzione scalare ad essa passata, entro una regione specificata da certi vincoli. Questa funzione è stata costruita come somma quadratica dei residui, in modo da replicare il funzionamento di lsqnonlin.

fmincon, similmente a lsqnonlin, è stata richiamata con la seguente sintassi :

```
opt=optimset('Display','off','Algorithm','interior-point',  
            'Jacobian','on');  
gnorm=fmincon('fmincon_norm',g0,[],[],[],[],lb,ub,[],opt);
```

INPUT :

- $g_0$  è, come prima, il vettore delle componenti del dipolo da cui parte la stima, le quali sono state fatte variare anche ora in modo random negli stessi limiti descritti precedentemente
- 'fmincon\_norm' è la function contenente la F che fmincon andrà a minimizzare; essa è costruita sostanzialmente nello stesso modo della F utilizzata da lsqnonlin, ma infine ad fmincon viene passata la somma dei quadrati dei residui, quindi una funzione scalare (per maggiori dettagli si veda l'appendice A in fondo a questo documento) . Essa è definita nel seguente modo :

```
[F,J] = fmincon_norm(gnorm)
```

dove  $J$  è la matrice Jacobiana della  $F$ , ossia la matrice contenente le derivate parziali della  $F$  rispetto a ciascuna delle variabili, introdotta manualmente per aumentare la velocità ed in parte la precisione dell'algoritmo di ottimizzazione; la sua espressione è stata calcolata sempre tramite MatLab<sup>®</sup>, utilizzando espressioni simboliche delle variabili e delle costanti della  $F$ .

- `Lb` ed `ub` sono i vincoli imposti ad `fmincon`, come fatto per `lsqnonlin` in precedenza
- `opt` definisce delle opzioni aggiuntive che vengono utilizzate da `fmincon`; `'Display','off'` consente di non mostrare sulla command window i risultati di ciascuna ottimizzazione; `'Algorithm','interior-point'` stabilisce l'algoritmo che `fmincon` utilizzerà, appunto 'interior-point'; `'Jacobian','on'` permette di utilizzare la matrice jacobiana esatta fornita all'interno di `fmincon_norm`, altrimenti verrebbe utilizzata una matrice jacobiana calcolata numericamente a partire dalla  $F$  stessa, con conseguente introduzione di errori numerici.

OUTPUT :

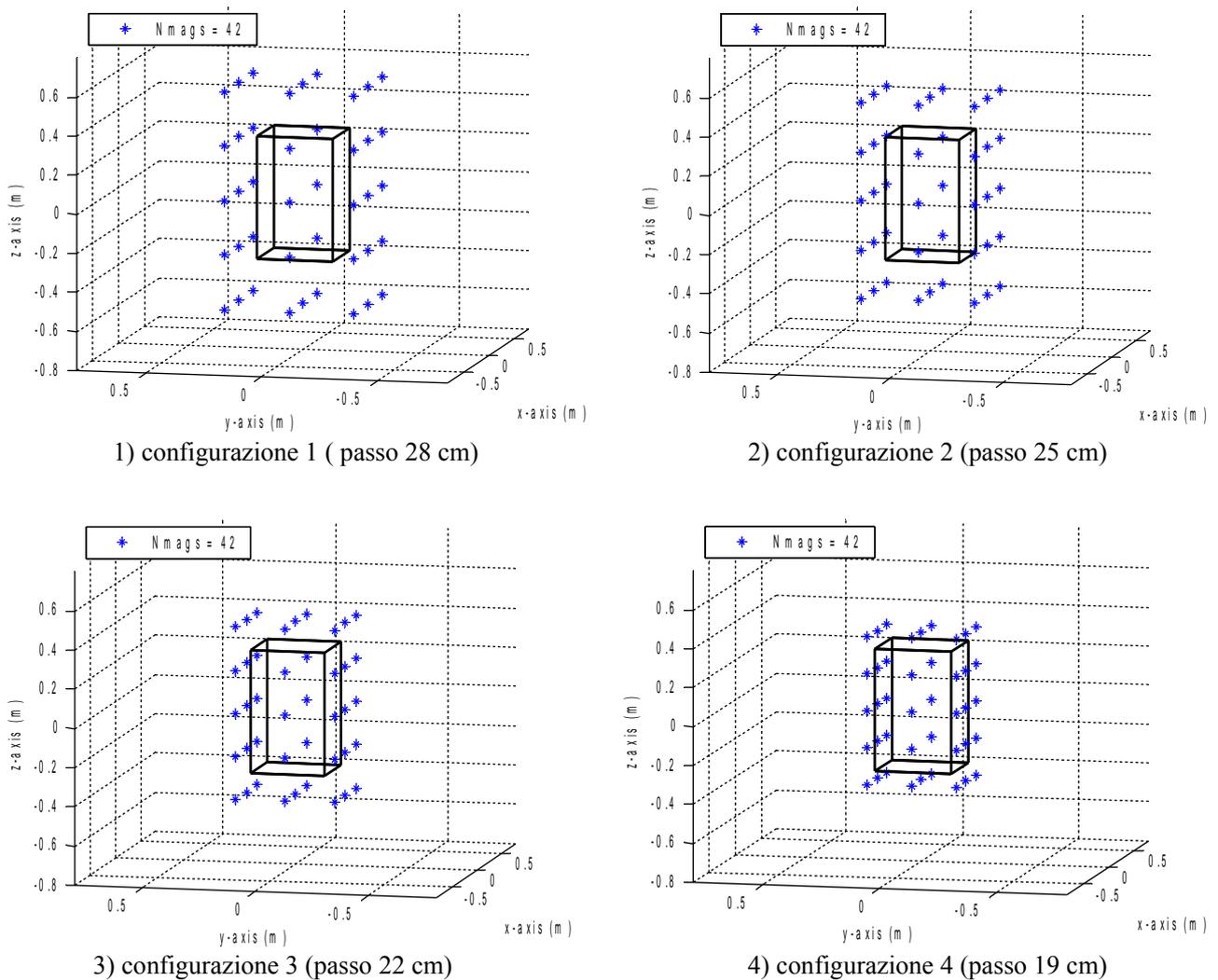
- `gnorm` è il vettore che permette di minimizzare la  $F$ , e quindi soluzione del problema.

### 3.3.2 Posizioni dei magnetometri

Considerate le dimensioni di ESEO (altezza : 0,63m; base : quadrata con lato 0,33m), si è cercato di trovare una distribuzione dei magnetometri attorno al satellite che fosse sì uniforme (dato che non è noto come il dipolo sia orientato, e che intensità abbia), ma anche caratterizzata da un numero non troppo elevato degli stessi, per non richiedere procedure di misura molto lunghe e complesse, e che fosse in grado di fornire una stima con livelli di errore accettabili.

Sono stati studiati 4 schemi caratterizzati dallo stesso numero di magnetometri (42), ma posizionati con passo differente tra di loro, ed in particolare decrescente, in modo da avere i magnetometri mano a mano più vicini al satellite.

Le configurazioni utilizzate sono riportate nei seguenti plot; i magnetometri sono indicati dagli asterischi \* :



**Fig. 3.4** Configurazioni di magnetometri con rispettive distanze (passi) tra di essi

L'obiettivo, con configurazioni del genere, è stato quindi quello di capire quale fosse l'influenza della vicinanza/lontananza dei magnetometri dal satellite.

### 3.3.4 Rumori considerati ed errori valutati

Le imprecisioni relative ai magnetometri che sono state prese singolarmente in considerazione nel lavoro di tesi, e gli errori da esse derivanti, sono di seguito esposti.

Dal momento che i rumori sono stati considerati come valori random, con distribuzione più o meno uniforme, sono state effettuate più prove, nominalmente 100, mediando in fine gli errori risultanti da ciascuna per lo stesso schema di magnetometri.

Gli errori studiati sono quelli relativi a posizione del dipolo, orientazione del vettore di campo magnetico e sua intensità. Per ogni schema di magnetometri e per ogni tipo di rumore sono stati riportati in grafici gli andamenti dei predetti errori, mediati come detto prima, all'aumentare della deviazione standard del rumore considerato, validando in questa maniera la funzionalità dell'algoritmo di ottimizzazione e caratterizzandone l'accuratezza. Per ogni livello di rumore è stata plottata anche la deviazione standard tra gli errori delle varie prove, utilizzando la funzione di stampa 'errorbar'; la linea della deviazione standard indica che con certi valori del rumore in considerazione, compresi comunque nel particolare valore della deviazione standard dello stesso, si può ottenere un livello dell'errore in considerazione maggiore o minore di quello medio, indicato dalla spezzata d'errore.

L'errore sulla **posizione del dipolo** è stato calcolato come norma del vettore congiungente la posizione del dipolo noto (simulato) e quella risultante dalla procedura di stima :

```
err_posizione(n, qm, qp, qr, qc, qs, cc) = sqrt((g(1)-a)^2+(g(2)-b)^2+(g(3)-c)^2);
```

Dove n indica il numero della prova, qm il livello di deviazione standard sul rumore dei magnetometri, qp il livello di deviazione standard sulla posizione dei magnetometri, qr sulla loro orientazione, qc sul bias di calibrazione, qs sui fattori di scala, e cc il numero della configurazione in analisi.

L'errore sull'**orientazione del dipolo** è stato calcolato come l'arcocoseno del rapporto tra il prodotto scalare di **x** (vettore dipolo simulato) e **y** (vettore dipolo stimato) ed il prodotto delle loro norme :

$$\theta = \arccos \left[ \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|} \right]$$

```
err_orientamento(n, qm, qp, qr, qc, qs, cc) =  
acosd( (M*g(4)+N*g(5)+P*g(6)) /  
(sqrt(M^2+N^2+P^2)*sqrt(g(4)^2+g(5)^2+g(6)^2)) );
```

L'errore sull'**intensità del dipolo** è stato calcolato come errore assoluto percentuale tra le intensità dei due vettori di dipolo, indicati prima :

$$\varepsilon = \left[ \frac{\|\vec{y}\| \|\vec{x}\|}{\|\vec{x}\|} \right] \cdot 100$$

```
err_intensita(n, qm, qp, qr, qc, qs, cc) = (abs(sqrt(M^2+N^2+P^2) -  
sqrt(g(4)^2+g(5)^2+g(6)^2)) / sqrt(g(4)^2+g(5)^2+g(6)^2)) * 100;
```

Sono quindi di seguito riportate le sorgenti di errori di misura considerati, ed i conseguenti grafici rappresentanti gli errori effettuati dall' algoritmo di stima e prima elencati, analizzati per ciascuno dei rumori separatamente.

**Rumore intrinseco :**

per studiare la dipendenza della precisione delle misure in funzione del rumore dei magnetometri, sono stati considerati più livelli dello stesso (6, compreso quello nullo); il rumore intrinseco è stato considerato come un rumore bianco con distribuzione gaussiana della densità di probabilità, con media nulla e deviazione standard aumentante da 0 ad 1nT ad incrementi di 0.2nT :

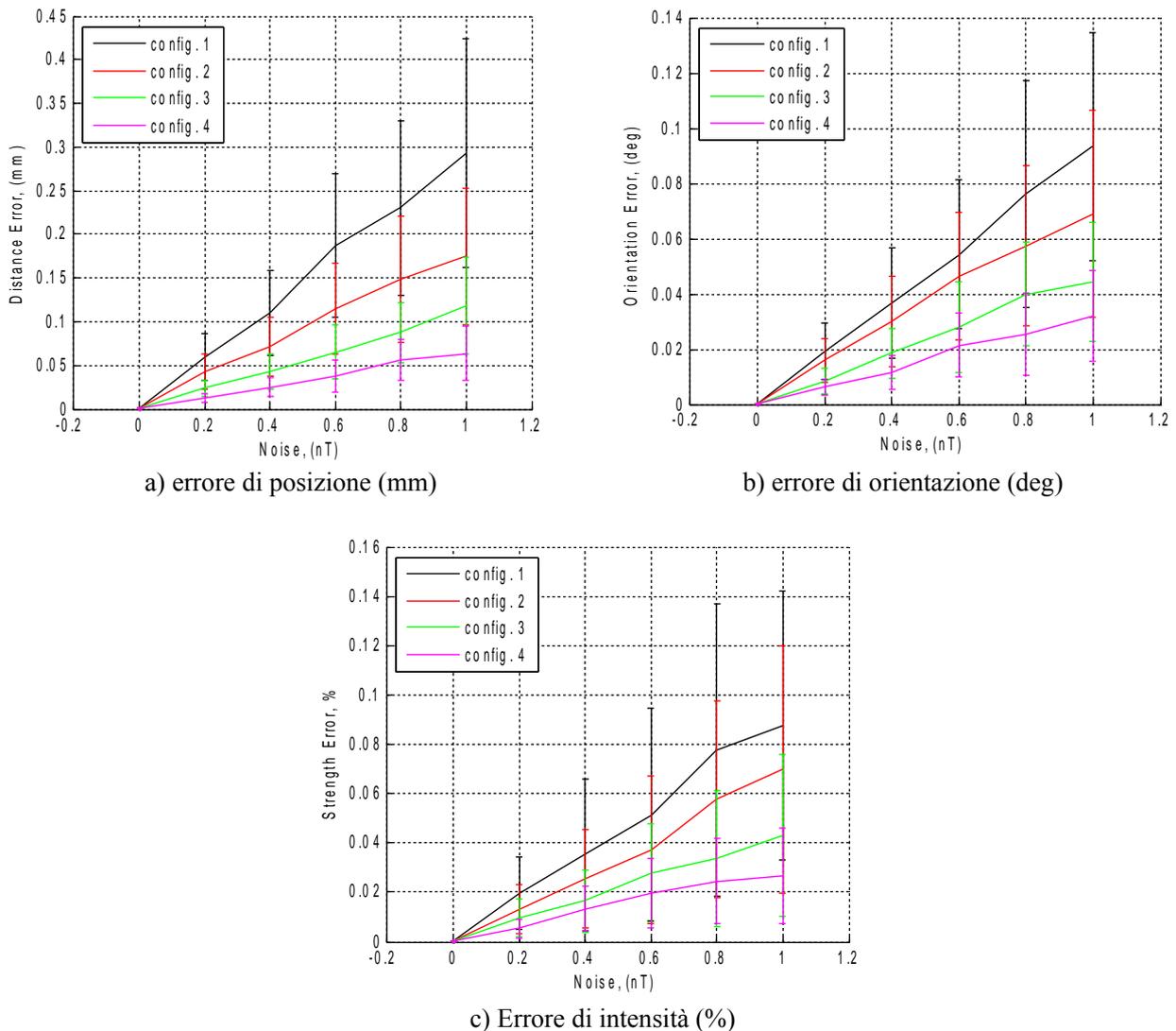
```
sigmas_mag = [0.0, 2e-10, 4e-10, 6e-10, 8e-10, 1e-9]
```

Il rumore intrinseco è stato quindi calcolato come :

```
noise_mag = sigmas_mag(qm) * randn(length(Bx_meas), 3)
```

ed è stato direttamente sommato ai valori Bx\_meas, By\_meas e Bz\_meas rilevati da ciascun magnetometro (ognuno avrà ad ogni prova il suo livello di rumore) e non affetti da alcun rumore, quindi ideali.

I risultati sono mostrati in Fig. 3.5 :



**Fig. 3.5 Errori di stima in funzione del rumore dei magnetometri**

Per rumore nullo l'errore di stima dell'algoritmo è molto basso, quindi trascurabile, e ciò verifica la sua funzionalità.

Sulla posizione del dipolo si è ottenuto un errore medio massimo pari a circa 0.3mm per lo schema 1 dei magnetometri, ma la barra della deviazione standard mostra che si potrebbe avere un picco pari a  $\sim 0.45\text{mm}$ , ma con una bassa probabilità. La configurazione 4 è invece caratterizzata da un errore medio massimo di  $\sim 0.07\text{mm}$ , quindi molto inferiore.

Sull'orientazione l'errore medio massimo è pari a  $\sim 0.1^\circ$ , abbastanza basso.

Sull'intensità l'errore medio massimo è di circa lo 0.1%, quindi non particolarmente rilevante.

Si può quindi concludere che con magnetometri caratterizzati da un basso valore del rumore intrinseco, dell'ordine dei nT, gli errori commessi nella misurazione del dipolo magnetico residuo che si possono commettere sono relativamente bassi, ed ancora minori utilizzando una configurazione di magnetometri più vicina al satellite.

### Errore di posizionamento :

è stata quindi presa in considerazione la possibilità di una collocazione non corretta dei magnetometri nella loro posizione.

Anche quest'errore è stato considerato come un rumore bianco gaussiano, con media nulla e deviazione standard andante da 0 ad 1cm, ad intervalli di 2mm :

```
sigmas_pos = [0.0, 2e-3, 4e-3, 6e-3, 8e-3, 1e-2]
```

L'errore di posizionamento è stato quindi calcolato come :

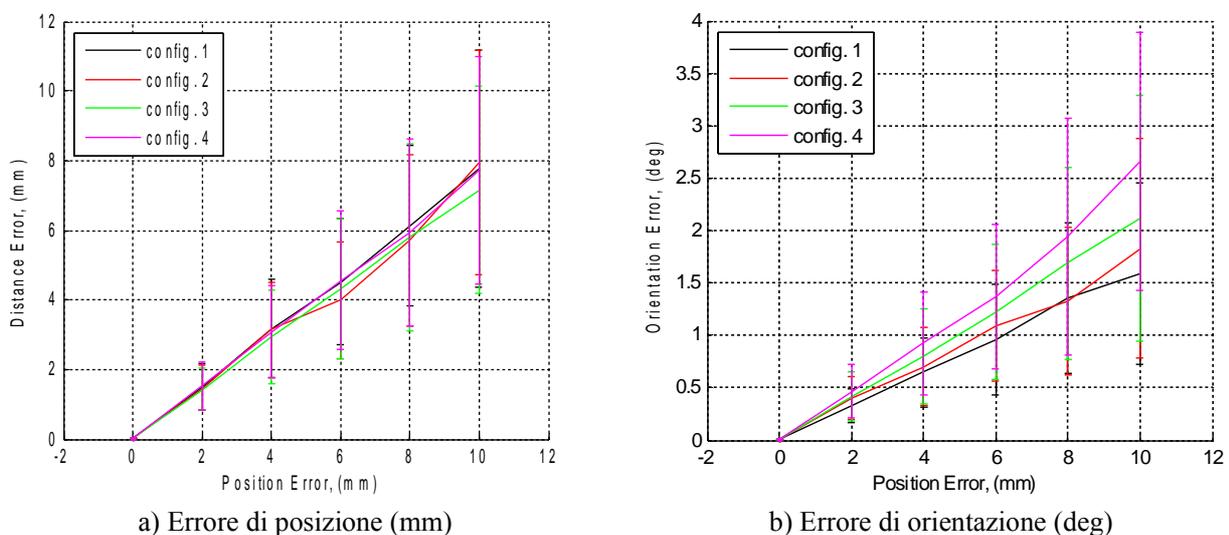
```
noise_pos = sigmas_pos(qp) * randn(length(Bx_meas), 3)
```

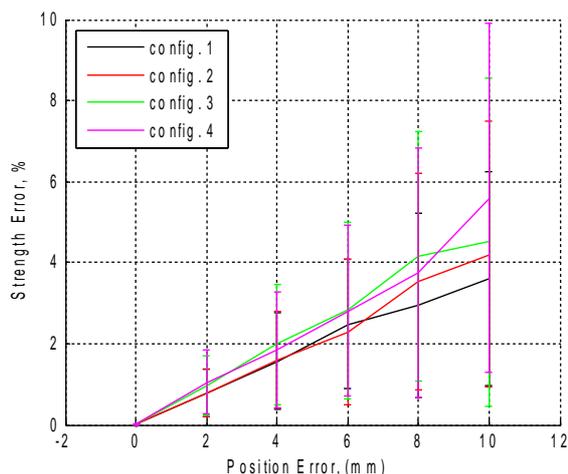
ed è stato sommato alle componenti x,y,z delle posizioni corrette dei magnetometri :

```
Pos_meas = Pos + noise_pos;
```

Sono quindi stati calcolati i valori  $B_x\_meas$ ,  $B_y\_meas$ ,  $B_z\_meas$  con queste nuove posizioni errate.

Gli errori derivanti da questi errori di posizionamento sono riportati in Fig. 3.6 :





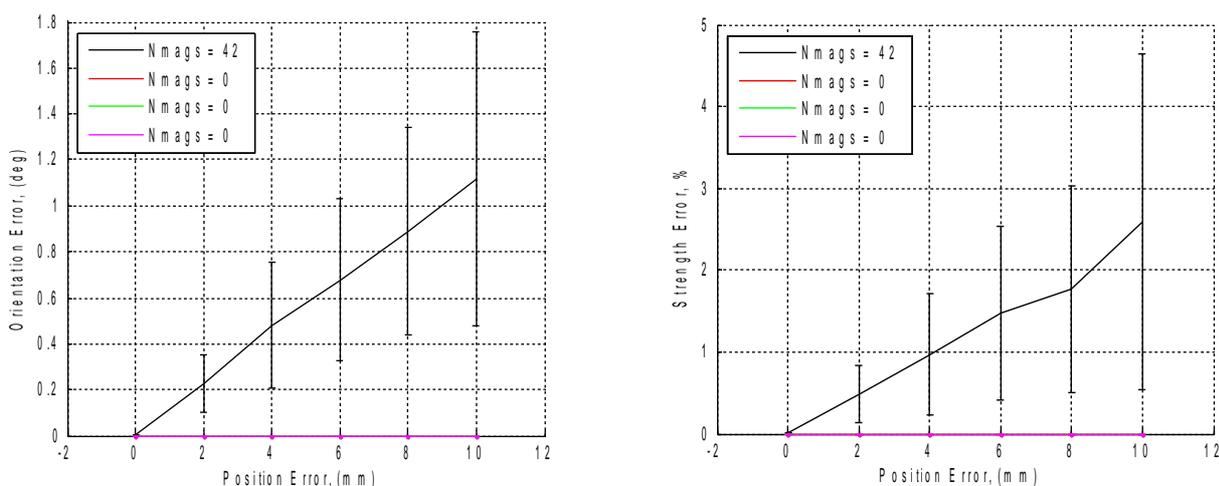
c) Errore di intensità (%)

**Fig. 3.6 Errori di stima in funzione delle posizioni dei magnetometri**

Sulla posizione del dipolo si è notato un andamento dell'errore di stima molto simile per ogni schema, con un massimo di circa 8mm.

Il comportamento degli errori di orientazione ed intensità è invece a prima vista ambiguo, in quanto lo schema di magnetometri che precedentemente si è rivelato migliore (il quarto), adesso sembra fornire gli errori maggiori. Si può cercare di fornire una spiegazione a questo comportamento evidenziando il fatto che più si è vicini al satellite, più elevata è l'intensità del dipolo residuo, per cui evidentemente un piccolo errore di posizionamento dei magnetometri dello schema 1, che sono quelli posizionati più in lontananza, disturba poco la misura, mentre lo stesso errore, inserito sui magnetometri posizionati più vicino al satellite, la fa variare di una quantità maggiore.

Questo comportamento è stato accertato posizionando i magnetometri della configurazione 1 ancora più distanti dal satellite : il risultato è stato un insieme di errori su orientazione ed intensità del dipolo ancora più bassi, come mostrato in Fig. 3.7 :



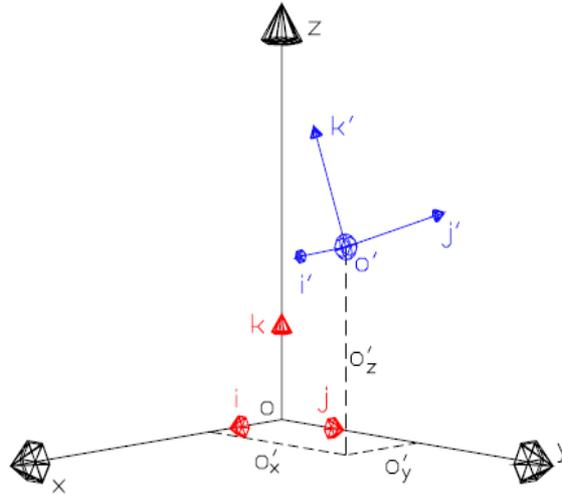
**Fig. 3.7 Errori di orientazione ed intensità per una configurazione 1 più distante dal satellite**

Si può quindi concludere che, volendo utilizzare una disposizione di magnetometri più vicina al satellite, bisogna porre una particolare attenzione al posizionamento degli stessi.

### Errore di orientazione :

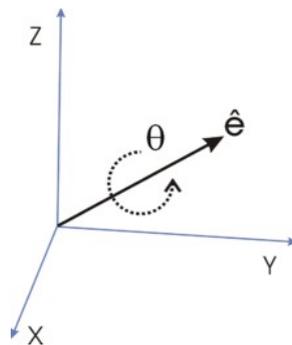
un possibile errore di allineamento degli assi dei magnetometri rispetto a quelli del satellite è stato modellizzato con una “piccola” rotazione casuale del sistema assi magnetometro.

Una rotazione del SdR assi magnetometro (blu) rispetto al SdR del satellite (rosso) può essere rappresentata come nella Fig. 3.8 :



**Fig. 3.8 Rotazione del sistema assi magnetometro rispetto al SdR del satellite**

La rotazione è stata generata tramite la rappresentazione asse-angolo di Eulero. Tale rappresentazione è basata sul teorema di Eulero, il quale stabilisce che ogni composizione di rotazioni può sempre essere ricondotta ad una rotazione di un angolo ('angolo di Eulero')  $\theta$ , intorno ad un asse ('versore di Eulero') di rotazione  $u$  [12].



**Fig. 3.9 Rotazione asse/angolo di Eulero**

La matrice di rotazione corrispondente sarà pertanto :

$$Rot(\hat{e}, \theta) = \begin{bmatrix} e_x^2(1-\cos\theta) + \cos\theta & e_x e_y(1-\cos\theta) - e_z \sin\theta & e_x e_z(1-\cos\theta) + e_y \sin\theta \\ e_x e_y(1-\cos\theta) + e_z \sin\theta & e_y^2(1-\cos\theta) + \cos\theta & e_y e_z(1-\cos\theta) - e_x \sin\theta \\ e_x e_z(1-\cos\theta) - e_y \sin\theta & e_y e_z(1-\cos\theta) + e_x \sin\theta & e_z^2(1-\cos\theta) + \cos\theta \end{bmatrix}$$

È stata quindi considerata una rotazione random di questo tipo per ogni magnetometro.

Per simulare un errore di rotazione casuale la rotazione è stata generata nel seguente modo:

- Asse di Eulero: versore unitario random con distribuzione uniforme su una semisfera dello spazio caratterizzata da  $z > 0$ . Questo perché, dato un asse di Eulero di questo tipo, le rotazioni opposte sono rappresentate da un angolo di Eulero negativo.
- Angolo di Eulero: processo stocastico bianco con distribuzione gaussiana, con deviazioni standard da  $0^\circ$  a  $5^\circ$ .

La rotazione è stata introdotta nel seguente modo :

```
ez = rand(length(Bx_meas), 1);
phi = 2*pi*rand(length(Bx_meas), 1);
ex = sin(acos(ez)).*cos(phi);
ey = sin(acos(ez)).*sin(phi);

sigmas_rot = [0, 1, 2, 3, 4, 5];
eul_angle = sigmas_rot(qr)*(pi/180)*randn(length(Bx_meas), 1);

for p = 1:length(Pos)
    Rot_mat = vrrotvec2mat([ex(p); ey(p); ez(p); eul_angle(p)]);

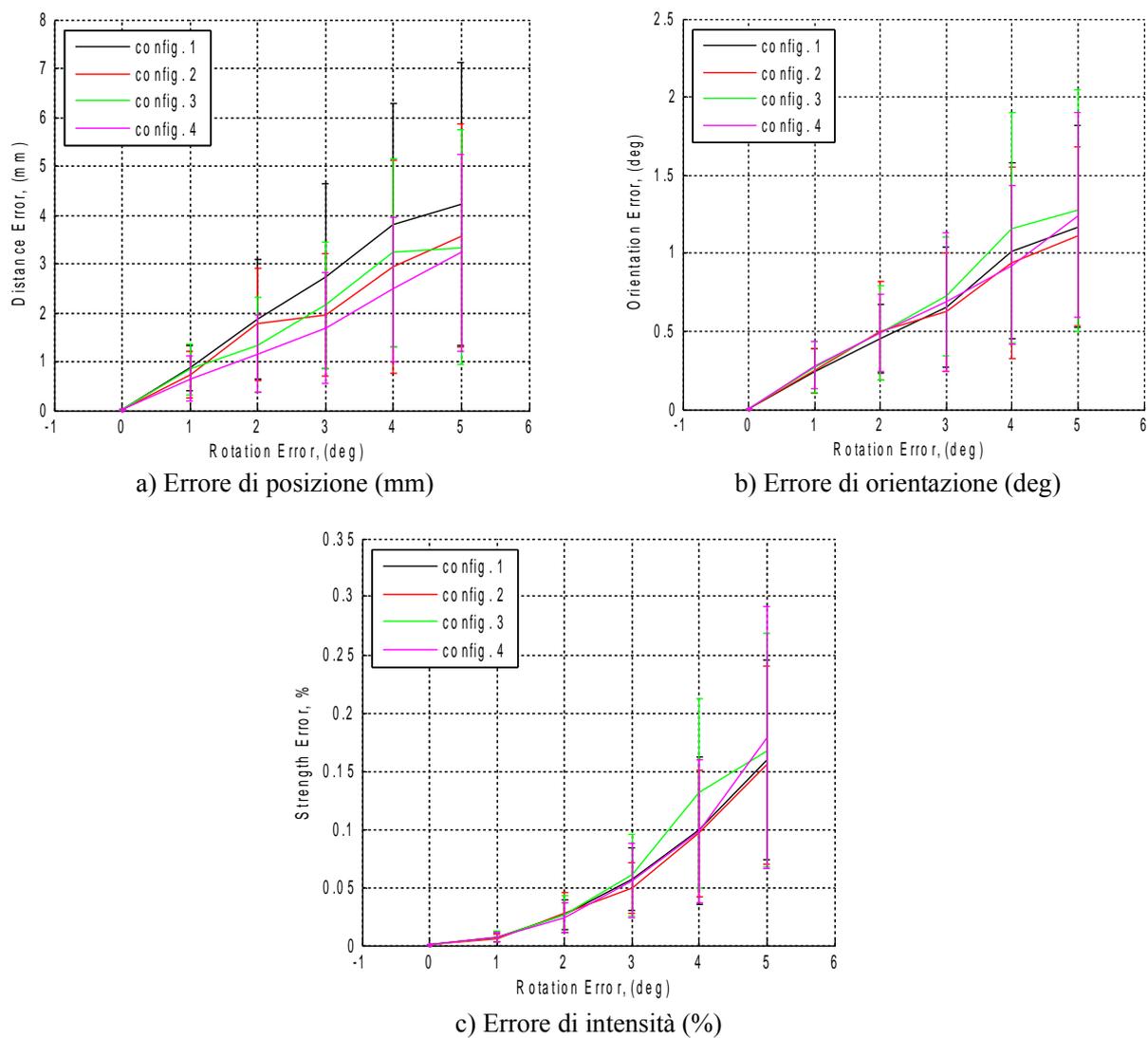
    B_meas_rot = Rot_mat*[Bx_meas(p,1); By_meas(p,1); Bz_meas(p,1)];

    Bx_meas_noise(p,1) = B_meas_rot(1);
    By_meas_noise(p,1) = B_meas_rot(2);
    Bz_meas_noise(p,1) = B_meas_rot(3);
end
```

La matrice di rotazione è stata calcolata utilizzando la funzione 'vrrotvec2mat', che dalle componenti del versore di Eulero ( $e_x, e_y, e_z$ ) e dall'angolo di Eulero  $\theta$  costruisce automaticamente la matrice.

È stata quindi moltiplicata, per ogni posizione dei magnetometri, la matrice di rotazione per la matrice contenente le componenti del campo magnetico rilevato, ottenendo i  $B_{i\_meas\_noise}$  da utilizzare nell'ottimizzazione.

Gli errori risultanti da queste rotazioni sono mostrati in Fig. 3.10 :



**Fig. 3.10 Errori di stima in funzione dell'orientazione dei magnetometri**

Si vede quindi che un'orientazione errata dei magnetometri influisce maggiormente sulla stima della posizione del dipolo se i magnetometri sono a distanza maggiore dal satellite, avendo una stima errata (media massima) della posizione di circa 4mm. La distanza a cui sono posti i magnetometri influisce molto meno sulla stima di orientazione ed intensità del dipolo, in quanto dai grafici si vede che le curve hanno un andamento molto simile e ravvicinato, per cui un'orientazione errata dei magnetometri avrebbe ripercussioni sugli errori di stima di cui prima in modo molto simile per tutte le configurazioni considerate; sull'orientazione del dipolo si può commettere ugualmente un'errore massimo tra 1° ed 1,5°, mentre sull'intensità un errore medio di circa lo 0,2% .

### Bias di calibrazione :

Riguardo a possibili errori di calibrazione dei magnetometri sono stati considerati, come accennato in precedenza, solamente il bias ed i fattori di scala, non considerando disallineamento e rotazione degli assi in quanto si è già presa in esame la rotazione dei magnetometri.

I due coefficienti di calibrazione considerati sono stati analizzati separatamente.

Il bias è stato considerato come rumore costante, uguale su ciascun magnetometro, ma è stato cambiato in ogni prova in modo random con distribuzione uniforme della probabilità tra  $-\sigma$  e  $+\sigma$  :

```
sigmas_bias_calibration = [0.0, 0.05e-6, 0.1e-6, 0.15e-6, 0.2e-6,  
                           0.25e-6]
```

```
bias_calibration = sigmas_bias_calibration(qc) * (-1+2.*rand)
```

Gli errori risultanti sono i seguenti, mostrati in Fig. 3.11 :

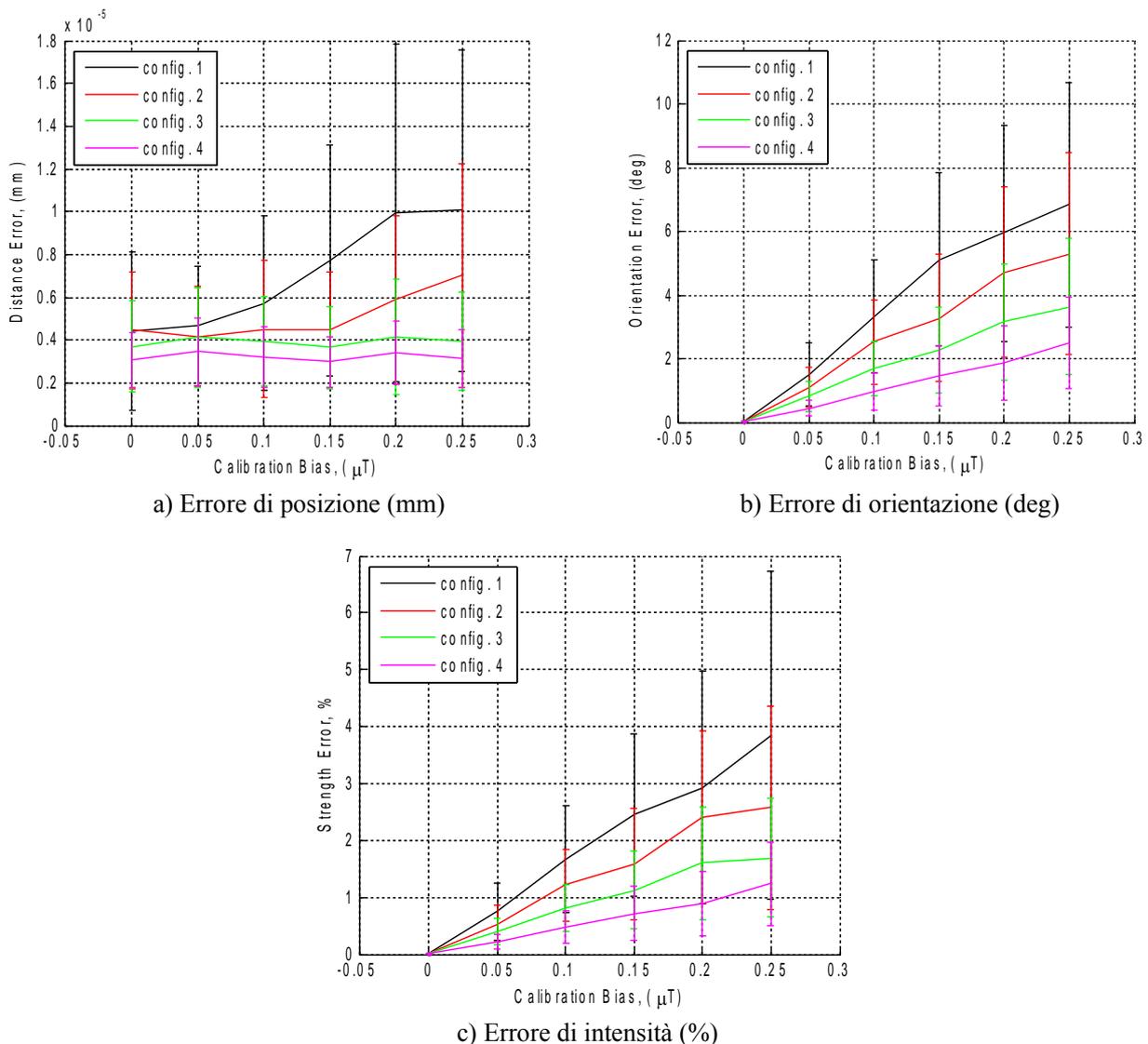


Fig. 3.11 Errori di stima in funzione dei bias di calibrazione

L'errore di posizione si è rivelato essere molto basso, dell'ordine dei  $\mu\text{m}$ , pertanto trascurabile; l'errore medio massimo sull'orientazione è dell'ordine dei  $7^\circ$  ed è dato dalla configurazione di magnetometri più lontana dal satellite; stesso risultato per l'errore di intensità del dipolo, con un massimo di circa il 4%.

### Fattori di scala di calibrazione k :

I fattori di scala sono stati modellizzati come un rumore costante, uguale su ciascun asse del magnetometro, ma è stato cambiato in ogni prova in modo random con densità di probabilità gaussiana, con media nulla e deviazioni standard andanti da 0 a 0.05:

```
sigmas_scaling_factors = [0.0, 0.01, 0.02, 0.03, 0.04, 0.05];
k = sigmas_scaling_factors(qs)*randn(3,1);
```

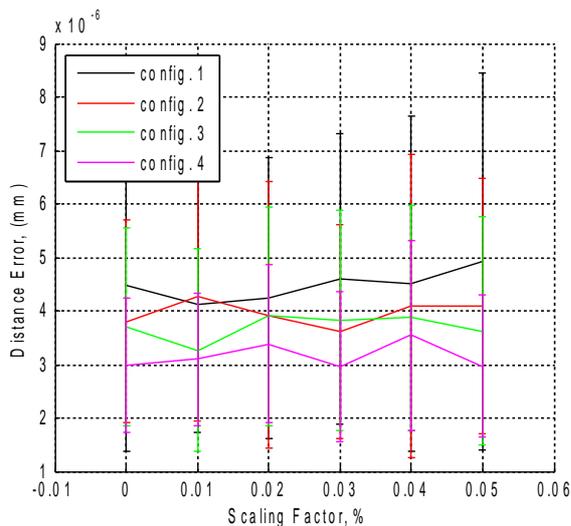
E la matrice dei fattori di scala è stata quindi generata come :

```
scaling_mat = [1+k(1,1), 0, 0; 0, 1+k(2,1), 0; 0, 0, 1+k(3,1)];
```

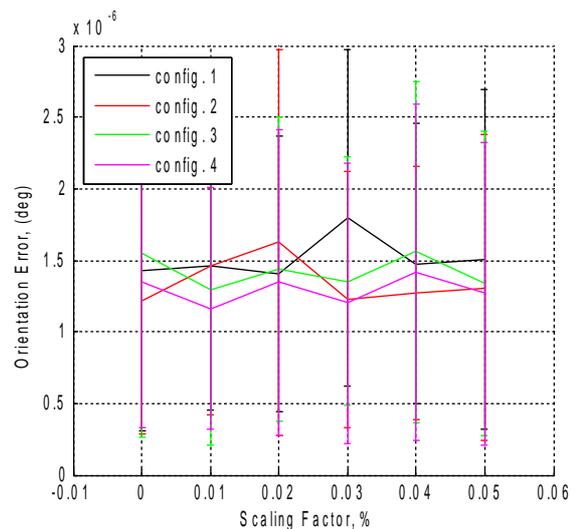
Avendo così i fattori di scala corrispondenti al massimo ad un errore del 5%; il rumore è stato quindi introdotto moltiplicando la matrice dei fattori di scala per la matrice contenente i  $B_i$  meas :

```
for t = 1:length(Pos)
    B_meas_noise = scaling_mat*[Bx_meas(t,1);
                               By_meas(t,1); Bz_meas(t,1)];
    Bx_meas_noise(t,1) = B_meas_noise(1,1);
    By_meas_noise(t,1) = B_meas_noise(2,1);
    Bz_meas_noise(t,1) = B_meas_noise(3,1);
end
```

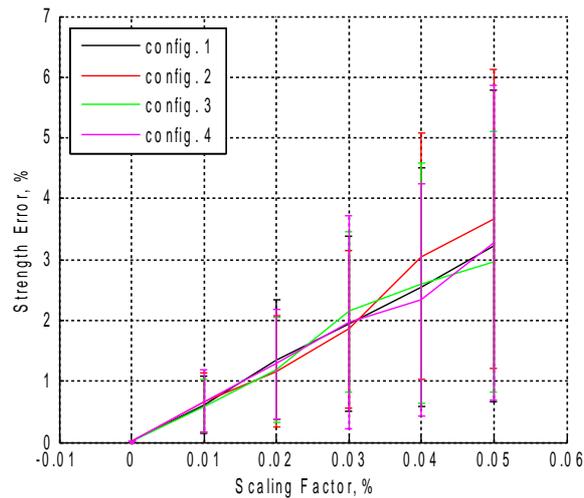
Gli errori risultanti sono stati quelli mostrati in Fig. 3.12 :



a) Errore di posizione (mm)



b) Errore di orientazione (deg)



c) Errore di intensità (%)

**Fig. 3.12 Errori di stima in funzione dei fattori di scala**

Su posizione ed orientamento del dipolo, per ogni configurazione di magnetometri, gli errori sono risultati molto bassi, dell'ordine dei  $\mu\text{m}$  e dei  $\mu^\circ$  rispettivamente, quindi in generale possono essere ritenuti trascurabili; sull'intensità gli errori si sono rivelati simili a quelli derivanti dai rumori considerati in precedenza, ma molto ravvicinati per tutte le configurazioni di satelliti, ed al massimo di circa il 4%.

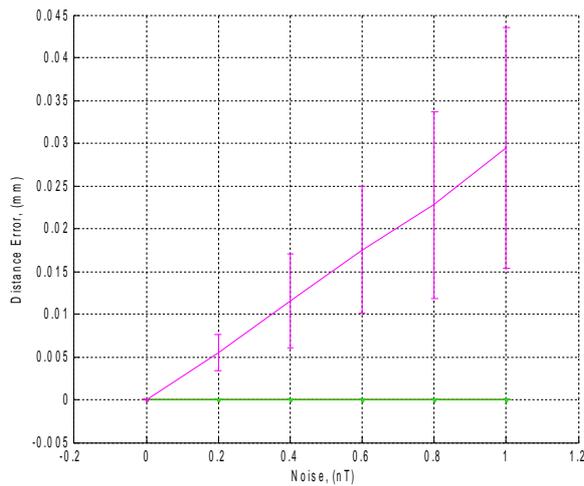
### 3.3.5 Dipendenza dal numero di misure

È stata infine analizzata la dipendenza dei valori degli errori fin'ora considerati dal numero di misure effettuate da ogni magnetometro. Nel precedente lavoro era stata considerata un'unica misura per ciascun magnetometro; ad un suo aumento si è riscontrata una maggiore accuratezza della stima effettuata dalla procedura di ottimizzazione, ma esclusivamente per quanto riguarda il rumore intrinseco dei magnetometri e gli errori di posizione ed orientazione legati ai fattori di scala dei magnetometri; l'aumento del numero di misure è infatti risultato non influire (a meno che la variazione non sia così piccola da poter essere considerata trascurabile) sugli errori dovuti all'errore di posizionamento e all'errore di orientazione, molto probabilmente perché, considerando i rumori intrinseco e di calibrazione nulli, il magnetometro, lasciato in quella stessa posizione e con quella stessa orientazione per effettuare più misure, è sempre affetto dagli stessi errori sulla sua posizione e sulla sua orientazione, rilevando così sempre lo stesso valore del campo magnetico generato dal dipolo; il numero di misure non è risultato influire neanche sugli errori dovuti ai bias di calibrazione, essendo essi costanti nel tempo.

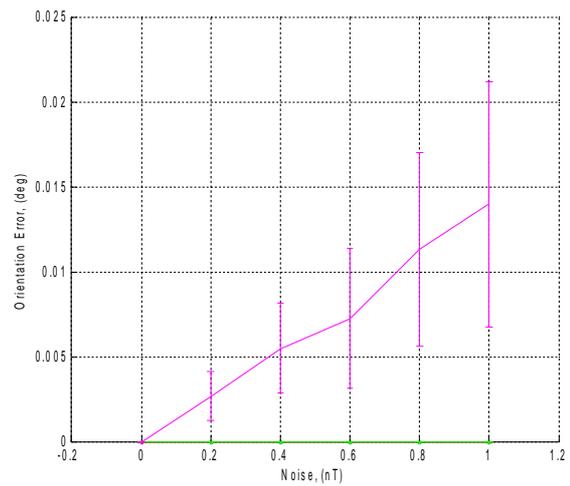
Considerando però che nella procedura sperimentale per la rilevazione del dipolo residuo tutti gli errori analizzati incorreranno con una certa entità, si vede come un numero maggiore di misure puoi aiutare ad ottenere risultati migliori.

L'aumento dell'accuratezza è dovuto al fatto che l'algoritmo di ottimizzazione ha in questo caso a sua disposizione una quantità maggiore di dati da analizzare e quindi elaborare; questo riduce gli effetti degli errori bianchi di un fattore pari alla radice quadrata del numero di misure, in quanto è come se essi venissero mediati.

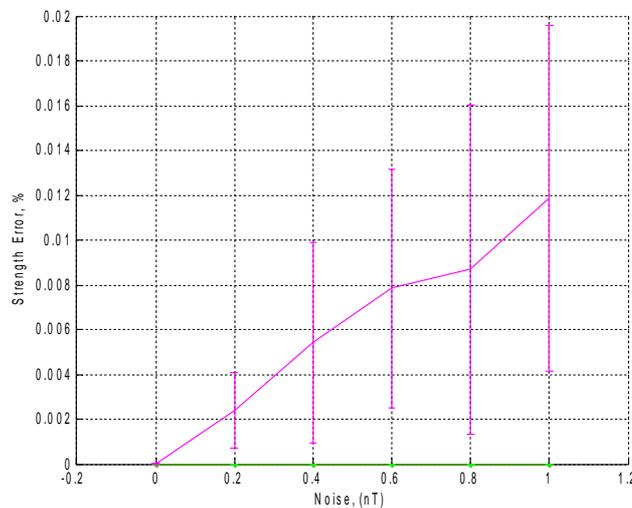
Si riportano quindi in Fig. 3.13 i grafici degli errori dovuti al rumore intrinseco dei magnetometri per la configurazione 4, confrontabili con i grafici ottenuti in precedenza, che mostrano una diminuzione degli errori ottenuti effettuando 5 misure con ciascun magnetometro anziché una sola :



a) Errore di posizione (mm)



b) Errore di orientazione (°)



c) Errore di intensità (%)

**Fig. 3.13 Errori di stima in funzione del rumore intrinseco per la configurazione 4, effettuando 5 misure**

Si nota una diminuzione dell'errore medio massimo sulla posizione del dipolo da  $\sim 0.06\text{mm}$  a  $0.03\text{mm}$ , una diminuzione dell'errore medio massimo di orientazione da  $\sim 0.03^\circ$  a  $\sim 0.015^\circ$ , ed una diminuzione dell'errore medio massimo sull'intensità dallo  $0,03\%$  a circa lo  $0,012\%$ , quindi appunto di un fattore di circa la radice quadrata di 5 .

### 3.4 Valutazione dei risultati ottenuti

Analizzando i grafici sugli errori di stima delle componenti del presunto dipolo residuo del satellite ESEO fin'ora ottenuti, considerando l'utilizzo di magnetometri caratterizzati da bassi valori di rumore intrinseco e coefficienti di calibrazione, si può concludere che :

- L'errore commesso dalla procedura di stima sulla posizione del dipolo risulta maggiore utilizzando una configurazione di magnetometri più distante dal satellite; quest'errore risulta poi influenzato maggiormente dal posizionamento dei magnetometri e dalla loro orientazione, e soprattutto dalla prima delle due.
- L'errore sull'orientazione del vettore di dipolo risulta maggiormente influenzato da posizione, orientazione e bias di calibrazione dei magnetometri; la configurazione migliore è risultata anche in questo caso quella più vicina al satellite, tranne che per l'errore sul posizionamento dei magnetometri, caso in cui questa configurazione fornisce gli errori più elevati sull'orientazione, per cui dovrebbe essere posta particolare attenzione a questo aspetto durante le prove sperimentali.
- L'errore sull'intensità del dipolo è risultato influenzato maggiormente dagli errori di posizionamento e di calibrazione (sia bias che fattori di scala), con errori minori ottenuti generalmente con la configurazione più vicina al satellite tranne che, come prima, per l'errore di posizionamento dei magnetometri.

In generale quindi, più i magnetometri sono in prossimità del satellite, più precisa risulterà la stima del dipolo residuo, ed è consigliabile effettuare una calibrazione dei magnetometri prima del lavoro sperimentale di rilevazione del campo magnetico generato dal dipolo.

L'errore generale di stima nel caso di utilizzo sperimentale del metodo può comunque essere ridotto effettuando ad ogni posizione dei magnetometri più misurazioni e fornendole tutte al procedimento di ottimizzazione.

## CAPITOLO 4

### Conclusioni

Lo studio effettuato in questo lavoro di tesi è consistito nell'implementazione simulata e caratterizzazione di un metodo sperimentale per la misura del dipolo magnetico residuo di micro-satelliti. Esso ha preso come satellite di riferimento la sonda European Student Earth Orbiter (ESEO) dell'agenzia spaziale europea (ESA). La conoscenza del dipolo residuo di un satellite è importante sia per la quantificazione della coppia di disturbo che agirà su di esso una volta in orbita, che per la calibrazione dei suoi magnetometri, quindi per il dimensionamento del sistema di controllo d'assetto, in quanto la sua presenza influenzerà le misure magnetiche a bordo.

La procedura di misura del dipolo consiste, una volta posizionato il satellite all'interno di una gabbia di Helmholtz per far sì che si trovi in una zona magneticamente pulita, nell'acquisizione di misure del campo magnetico generato dal dipolo stesso per mezzo di magnetometri a tre assi. Per risalire alle grandezze caratteristiche del dipolo (posizione, orientazione ed intensità) sono necessarie più misurazioni attorno al satellite a causa delle diverse fonti di errori che caratterizzano normalmente le misure stesse; queste misure vanno poi inserite in una procedura numerica di ottimizzazione, la quale cerca di minimizzare gli effetti di questi errori sulle misure di campo magnetico; le sorgenti d'errore considerate in questo lavoro sono il rumore intrinseco dei magnetometri, gli errori di calibrazione dei magnetometri, ed un posizionamento ed un'orientazione degli stessi non corretti.

Tale procedura è stata simulata numericamente in ambiente MatLab®; è stato cioè considerato noto il dipolo residuo del micro-satellite, in modo da poter generare, tramite formule tratte da bibliografia, le misure dei magnetometri; la procedura di ottimizzazione è stata implementata con l'utilizzo delle funzioni di minimizzazione `fmincon` e `lsqnonlin` del software stesso. Ciascuna delle fonti d'errore considerate è stata modellizzata con distribuzione di probabilità gaussiana attorno al valore nullo, ad eccezione dei bias di calibrazione, considerati con distribuzione di probabilità uniforme.

Stabilite 4 configurazioni di magnetometri attorno al satellite, simili per disposizione ma con distanza tra i magnetometri differente, l'obiettivo è stato quello di studiare per ciascuna di queste l'andamento degli errori di stima commessi dall'algoritmo di ottimizzazione su posizione, orientazione ed intensità del vettore di dipolo residuo, all'aumentare della deviazione standard di ciascuno degli errori prima considerati; questi errori sono stati calcolati come differenze tra i valori simulati esatti ed i risultati della procedura di stima. Per ogni sorgente di errore sono stati quindi studiati separatamente i grafici rappresentanti questi andamenti, notando un maggiore aumento degli errori per la configurazione di magnetometri più distante dal satellite, ad eccezione del caso in cui è solamente presente l'errore di posizionamento, il che dice che quest'aspetto ha una maggiore importanza nelle configurazioni in cui i magnetometri sono più prossimi al satellite. Si è inoltre valutata l'influenza del numero di misure prese da ciascun magnetometro, notando un incremento di precisione dell'algoritmo di ottimizzazione nel caso di errori causati dal rumore intrinseco dei magnetometri.

Si può quindi infine suggerire, per il lavoro sperimentale di determinazione del dipolo magnetico residuo, una disposizione uniforme e vicina al satellite dei magnetometri, posizionati con la precisione più alta possibile, tutto ciò in modo da limitare gli errori caratteristici della procedura di stima.

## RINGRAZIAMENTI

Vorrei ringraziare innanzitutto la mia famiglia, che mi ha dato la possibilità di intraprendere questo percorso di studi ormai giunto al termine, il quale si spera rappresenti il preludio di una carriera carica di soddisfazioni e buoni propositi.

Un ringraziamento particolare va al professor Paolo Tortora, per avermi permesso di effettuare un lavoro nel quale testare le mie capacità acquisite nel corso della laurea triennale ed aver creduto in esse, al mio correlatore Marco Zannoni, per il continuo aiuto e la costante pazienza nel rispondere alle mie raffiche di e-mail, ed a tutti i ragazzi del laboratorio di radio scienza ed esplorazione planetaria per la loro accoglienza ed il loro sostegno morale, intellettuale... ed anche tecnico!

Un ultimo ringraziamento, speciale come gli altri, va a mio fratello Andrea per avermi aiutato particolarmente in questo periodo di lavoro di tesi, e senz'altro a tutti i miei compagni di vita e di studi che mi sono stati vicini in questi anni di impegno (e tenacia) costante(/i) e con i quali ho condiviso alcuni dei migliori momenti della mia vita, da studente fuori sede!

## BIBLIOGRAFIA

- [1] S.Pirrotta, P.Tortora; *Dispense del corso di avionica e strumentazione spaziale* A.A. 2013/2014
- [2] D.Modenini, P.Tortora; *Lecture notes in Spacecraft Attitude Dynamics and Control* A.Y. 2013/2014
- [3] Takaya Inamori, Nobutada Sako, Shinichi Nakasuka; *Magnetic dipole moment estimation and compensation for an accurate attitude control in nano-satellite missions*, 24 October 2010
- [4] ESEO mission; [http://www.esa.int/Education/ESEO\\_mission](http://www.esa.int/Education/ESEO_mission)
- [5] James Edgar Parks; *Helmholtz Coils – Uniform Magnetic fields*, september 2013
- [6] Francesco Cargioli; *Dimensionamento strutturale di un simulatore di campo magnetico ad elevata uniformità spaziale per test a terra di sistemi spaziali; elaborato finale di laurea in impianti aerospaziali*, II facoltà di ingegneria, Università di Bologna
- [7] Enciclopedia Treccani; *Magnetometro*
- [8] Marco Zannoni; *Sviluppo di algoritmi per la calibrazione e la fault detection dei magnetometri di bordo del microsatellite ALMASat-1; relazione di tirocinio*, Laboratorio di microsatelliti e microsistemi spaziali, II facoltà di ingegneria, Università di Bologna
- [9] John C. Sprigmann, James W. Cutler, Hasan Bahcivan; *Magnetic sensor calibration and residual dipole characterization for application to nanosatellites*, AIAA/AAS Astrodynamics specialist conference, 2-5 August 2010, Toronto, Ontario, Canada
- [10] Mathworks, “lsqnonlin” <http://it.mathworks.com/help/optim/ug/lsqnonlin.html>
- [11] Mathworks, “fmincon” <http://it.mathworks.com/help/optim/ug/fmincon.html>
- [12] Basilio Bona; *Rotazioni*, DAUIN-Politecnico di Torino, 2008

# APPENDICE A

## A.1 Algoritmo\_ESEO.m :

```
clear all
%close all
clc

tic;

global Bx_meas_noise By_meas_noise Bz_meas_noise Pos_meas x0 y0 z0 Sx0 Sy0 Sz0

%-----
% INPUTS
%-----

Nprove = 100;    %n° di prove con la stessa configurazione, per calcolare gli errori medi
                 (realtà - lsqnonlin) e plottarli
Nmisure = 5;

a = 0;          %
b = 0;          %
c = 0;          %dipolo magnetico 'noto' (per generare le misure simulate dei
                 magnetometri)
M = 0;          %
N = -0.05;      %
P = 0;          %

l = 0.33;       %dimensioni (l=lato, h=altezza) del micro-satellite
h = 0.63;       %
Smax = 0.1;

%Scelta del tipo di rumore e dello schema di posizionamento dei magnetometri :

dm1 = 0.02;     %distanza magnetometro-faccia del satellite caratteristica

configs = 1:4;
Nmags = zeros(length(configs),1);

sigmas_mag = [0.0, 2e-10, 4e-10, 6e-10, 8e-10, 1e-9];
sigmas_pos = [0.0]; % 2e-3, 4e-3, 6e-3, 8e-3, 1e-2];
sigmas_rot = [0]; % 1, 2, 3, 4, 5];
sigmas_bias_calibration = [0.0]; % 0.05e-6, 0.1e-6, 0.15e-6, 0.2e-6, 0.25e-6];
sigmas_scaling_factors = [0.0]; % 0.01, 0.02, 0.03, 0.04, 0.05];

%-----
% GENERAZIONE MISURE DEI MAGNETOMETRI SIMULATE
%-----

g01 = zeros(length(configs), length(sigmas_mag), length(sigmas_pos), length(sigmas_rot),
length(sigmas_bias_calibration), length(sigmas_scaling_factors), Nprove);
g02 = zeros(length(configs), length(sigmas_mag), length(sigmas_pos), length(sigmas_rot),
length(sigmas_bias_calibration), length(sigmas_scaling_factors), Nprove);
g03 = zeros(length(configs), length(sigmas_mag), length(sigmas_pos), length(sigmas_rot),
length(sigmas_bias_calibration), length(sigmas_scaling_factors), Nprove);
g04 = zeros(length(configs), length(sigmas_mag), length(sigmas_pos), length(sigmas_rot),
length(sigmas_bias_calibration), length(sigmas_scaling_factors), Nprove);
g05 = zeros(length(configs), length(sigmas_mag), length(sigmas_pos), length(sigmas_rot),
length(sigmas_bias_calibration), length(sigmas_scaling_factors), Nprove);
g06 = zeros(length(configs), length(sigmas_mag), length(sigmas_pos), length(sigmas_rot),
length(sigmas_bias_calibration), length(sigmas_scaling_factors), Nprove);
```

```

g1 = zeros(length(configs), length(sigmas_mag), length(sigmas_pos), length(sigmas_rot),
length(sigmas_bias_calibration), length(sigmas_scaling_factors), Nprove);
g2 = zeros(length(configs), length(sigmas_mag), length(sigmas_pos), length(sigmas_rot),
length(sigmas_bias_calibration), length(sigmas_scaling_factors), Nprove);
g3 = zeros(length(configs), length(sigmas_mag), length(sigmas_pos), length(sigmas_rot),
length(sigmas_bias_calibration), length(sigmas_scaling_factors), Nprove);
g4 = zeros(length(configs), length(sigmas_mag), length(sigmas_pos), length(sigmas_rot),
length(sigmas_bias_calibration), length(sigmas_scaling_factors), Nprove);
g5 = zeros(length(configs), length(sigmas_mag), length(sigmas_pos), length(sigmas_rot),
length(sigmas_bias_calibration), length(sigmas_scaling_factors), Nprove);
g6 = zeros(length(configs), length(sigmas_mag), length(sigmas_pos), length(sigmas_rot),
length(sigmas_bias_calibration), length(sigmas_scaling_factors), Nprove);

err_posizione = zeros(length(configs), length(sigmas_mag), length(sigmas_pos),
length(sigmas_rot), length(sigmas_bias_calibration), length(sigmas_scaling_factors),
Nprove);
err_orientamento = zeros(length(configs), length(sigmas_mag), length(sigmas_pos),
length(sigmas_rot), length(sigmas_bias_calibration), length(sigmas_scaling_factors),
Nprove);
err_intensita = zeros(length(configs), length(sigmas_mag), length(sigmas_pos),
length(sigmas_rot), length(sigmas_bias_calibration), length(sigmas_scaling_factors),
Nprove);

err_posizione_medio = zeros(length(configs), length(sigmas_mag), length(sigmas_pos),
length(sigmas_rot), length(sigmas_bias_calibration), length(sigmas_scaling_factors));
err_orientamento_medio = zeros(length(configs), length(sigmas_mag), length(sigmas_pos),
length(sigmas_rot), length(sigmas_bias_calibration), length(sigmas_scaling_factors));
err_intensita_medio = zeros(length(configs), length(sigmas_mag), length(sigmas_pos),
length(sigmas_rot), length(sigmas_bias_calibration), length(sigmas_scaling_factors));
err_posizione_std = zeros(length(configs), length(sigmas_mag), length(sigmas_pos),
length(sigmas_rot), length(sigmas_bias_calibration), length(sigmas_scaling_factors));
err_orientamento_std = zeros(length(configs), length(sigmas_mag), length(sigmas_pos),
length(sigmas_rot), length(sigmas_bias_calibration), length(sigmas_scaling_factors));
err_intensita_std = zeros(length(configs), length(sigmas_mag), length(sigmas_pos),
length(sigmas_rot), length(sigmas_bias_calibration), length(sigmas_scaling_factors));

err_posizione_medio0 = zeros(length(configs), length(sigmas_mag), length(sigmas_pos),
length(sigmas_rot), length(sigmas_bias_calibration), length(sigmas_scaling_factors));

xmin = -0.4;
xmax = +0.4;
zmin = -0.6;
zmax = +0.6;

% Progress bar
cd=0;
Ntot=length(configs)*length(sigmas_mag)*length(sigmas_pos)*length(sigmas_rot)*length(sigm
as_bias_calibration)*length(sigmas_scaling_factors)*Nprove;

for cc = 1:length(configs);

    if configs(cc) == 1

        dx = 0.28;
        x = [0:-dx:xmin, 0:dx:xmax];
        y = [0:-dx:xmin, 0:dx:xmax];
        z = [0:-dx:zmin, 0:dx:zmax];

    elseif configs(cc) == 2

        dx = 0.25;
        x = [0:-dx:xmin, 0:dx:xmax];
        y = [0:-dx:xmin, 0:dx:xmax];
        z = [0:-dx:zmin, 0:dx:zmax];

```

```

elseif configs(cc) == 3;

    dx = 0.22;
    x = [0:-dx:xmin, 0:dx:xmax];
    y = [0:-dx:xmin, 0:dx:xmax];
    z = [0:-dx:zmin, 0:dx:zmax];

elseif configs(cc) == 4

    xmin = -0.3;
    xmax = +0.3;
    zmin = -0.5;
    zmax = +0.5;
    dx = 0.19;
    x = [0:-dx:xmin, 0:dx:xmax];
    y = [0:-dx:xmin, 0:dx:xmax];
    z = [0:-dx:zmin, 0:dx:zmax];

end

POS = [];
for i = 1:length(x)
    for j = 1:length(y)
        for k = 1:length(z)
            for o = 1:Nmeasure
                if (x(i)<(-l/2-dm1) || x(i)>(l/2+dm1))
                    POS = [POS; x(i) y(j) z(k)];
                elseif (y(j)<(-l/2-dm1) || y(j)>(l/2+dm1))
                    POS = [POS; x(i) y(j) z(k)];
                elseif (z(k)<(-h/2-dm1) || z(k)>(h/2+dm1))
                    POS = [POS; x(i) y(j) z(k)];
                end
            end
        end
    end
end

Pos = unique(POS, 'rows');

Pos = repmat(Pos, Nmeasure, 1);

% Number of magnetometers :
Nmags(cc) = size(Pos, 1);

% Misura Ideali :
Bx_meas = zeros(length(Pos), 1);
By_meas = zeros(length(Pos), 1);
Bz_meas = zeros(length(Pos), 1);
for v = 1:length(Pos)
    [Bx_meas(v, 1), By_meas(v, 1), Bz_meas(v, 1)] = bfield_dipole(Pos(v, :), [a, b, c],
                                                                    [M, N, P]);
end

%-----
% AGGIUNTA RUMORI, CALCOLO COMPONENTI TRAMITE ALGORITMO DI OTTIMIZZAZIONE QUADRATI E
% PLOTTAGGIO ERRORI :
%-----

for qm = 1:length(sigmas_mag)

    for qp = 1:length(sigmas_pos)

        for qr = 1:length(sigmas_rot)

            for qc = 1:length(sigmas_bias_calibration)

```

```

for qs = 1:length(sigmas_scaling_factors)

    for n = 1:Nprove

        % Update counter :
        cd = cd+1;

        % Progress bar :
        progress_bar(cd,Ntot,2);

        % Scaling factors calibrazione :
        k = sigmas_scaling_factors(qs)*randn(3,1);

        scaling_mat = [1+k(1,1), 0, 0; 0, 1+k(2,1), 0; 0, 0,
                        1+k(3,1)];

        for t = 1:length(Pos)
            B_meas_noise = scaling_mat*[Bx_meas(t,1); By_meas(t,1);
                                       Bz_meas(t,1)];
            Bx_meas_noise(t,1) = B_meas_noise(1,1);
            By_meas_noise(t,1) = B_meas_noise(2,1);
            Bz_meas_noise(t,1) = B_meas_noise(3,1);
        end

        % Bias calibrazione :
        bias_calibration = sigmas_bias_calibration(qc)*(-1+2.*rand);

        Bx_meas_noise = Bx_meas_noise+bias_calibration;
        By_meas_noise = By_meas_noise+bias_calibration;
        Bz_meas_noise = Bz_meas_noise+bias_calibration;

        % Rumore misure :
        noise_mag = sigmas_mag(qm)*randn(length(Bx_meas),3);

        Bx_meas_noise = Bx_meas_noise + noise_mag(:,1);
        By_meas_noise = By_meas_noise + noise_mag(:,2);
        Bz_meas_noise = Bz_meas_noise + noise_mag(:,3);

        % Rumore posizioni :
        noise_pos = sigmas_pos(qp)*randn(length(Bx_meas),3);

        Pos_meas = Pos + noise_pos;

        % Rumore rotazioni :
        eul_angle = sigmas_rot(qr)*(pi/180)*randn(length(Bx_meas),1);

        ez = rand(length(Bx_meas),1);
        phi = 2*pi*rand(length(Bx_meas),1);
        ex = sin(acos(ez)).*cos(phi);
        ey = sin(acos(ez)).*sin(phi);

        for p = 1:length(Pos)
            Rot_mat = vrrotvec2mat([ex(p); ey(p); ez(p);
                                   eul_angle(p)]);

            B_meas_rot = Rot_mat*[Bx_meas_noise(p,1);
                                  By_meas_noise(p,1); Bz_meas_noise(p,1)];

            Bx_meas_noise(p,1) = B_meas_rot(1);
            By_meas_noise(p,1) = B_meas_rot(2);
            Bz_meas_noise(p,1) = B_meas_rot(3);
        end
    end
end

```

```

end

% Normalizza inputs :
x0 = 1/2; y0 = 1/2; z0 = h/2;
Sx0 = Smax; Sy0 = Smax; Sz0 = Smax;

% Condizione iniziale (normalizzate) :
g01(cc, qm, qp, qr, qc, qs, n) = (-1+2*rand); %
g02(cc, qm, qp, qr, qc, qs, n) = (-1+2*rand); %
g03(cc, qm, qp, qr, qc, qs, n) = (-1+2*rand); % random starting
% estimates

g04(cc, qm, qp, qr, qc, qs, n) = (-1+2*rand); %
g05(cc, qm, qp, qr, qc, qs, n) = (-1+2*rand); %
g06(cc, qm, qp, qr, qc, qs, n) = (-1+2*rand); %

g0 =
[g01(cc, qm, qp, qr, qc, qs, n), g02(cc, qm, qp, qr, qc, qs, n), g03(cc, qm, qp, qr, qc, qs, n), g04(cc, qm, qp,
qr, qc, qs, n), g05(cc, qm, qp, qr, qc, qs, n), g06(cc, qm, qp, qr, qc, qs, n)]; %starting estimates :
g01=a0, g02=b0, g03=c0, g04=M0, g05=N0, g06=P0

% Upper and lower bounds (normalized) :
lb = [-1,-1,-1,-1,-1,-1];
ub = [+1,+1,+1,+1,+1,+1];

% LSQNONLIN :
%opt = optimset('Display','off', 'Jacobian','on');
%[gnorm, resnorm, residual, exitflag, output] =
lsqnonlin('funzione_minimizzazione_norm', g0, lb, ub, opt);

% FMINCON :
opt = optimset('Display','off', 'Algorithm','interior-point',
'Jacobian','on');
gnorm=fmincon('fmincon_norm', g0, [], [], [], [], lb, ub, [], opt);

% De-normalize solution :
g1(cc, qm, qp, qr, qc, qs, n) = gnorm(1)*x0;
g2(cc, qm, qp, qr, qc, qs, n) = gnorm(2)*y0;
g3(cc, qm, qp, qr, qc, qs, n) = gnorm(3)*z0;
g4(cc, qm, qp, qr, qc, qs, n) = gnorm(4)*Sx0;
g5(cc, qm, qp, qr, qc, qs, n) = gnorm(5)*Sy0;
g6(cc, qm, qp, qr, qc, qs, n) = gnorm(6)*Sz0;

err_posizione(cc, qm, qp, qr, qc, qs, n) = sqrt((a-
g1(cc, qm, qp, qr, qc, qs, n))^2+(b-g2(cc, qm, qp, qr, qc, qs, n))^2+(c-g3(cc, qm, qp, qr, qc, qs, n))^2);

err_orientamento(cc, qm, qp, qr, qc, qs, n) =
acosd((M*g4(cc, qm, qp, qr, qc, qs, n)+N*g5(cc, qm, qp, qr, qc, qs, n)+P*g6(cc, qm, qp, qr, qc, qs, n))/
(sqrt(M^2+N^2+P^2))*sqrt(g4(cc, qm, qp, qr, qc, qs, n)^2+g5(cc, qm, qp, qr, qc, qs, n)^2+g6(cc, qm, qp, q
r, qc, qs, n)^2));

err_intensita(cc, qm, qp, qr, qc, qs, n) =
(abs(sqrt(g4(cc, qm, qp, qr, qc, qs, n)^2+g5(cc, qm, qp, qr, qc, qs, n)^2+g6(cc, qm, qp, qr, qc, qs, n)^2)-
sqrt(M^2+N^2+P^2))/(sqrt(M^2+N^2+P^2)))*100;

end

I1 = remove_outliers(squeeze(g1(cc, qm, qp, qr, qc, qs, :)), 4);
I2 = remove_outliers(squeeze(g2(cc, qm, qp, qr, qc, qs, :)), 4);
I3 = remove_outliers(squeeze(g3(cc, qm, qp, qr, qc, qs, :)), 4);
I4 = remove_outliers(squeeze(g4(cc, qm, qp, qr, qc, qs, :)), 4);
I5 = remove_outliers(squeeze(g5(cc, qm, qp, qr, qc, qs, :)), 4);
I6 = remove_outliers(squeeze(g6(cc, qm, qp, qr, qc, qs, :)), 4);
I = (I1==1 & I2==1 & I3==1 & I4==1 & I5==1 & I6==1);

```

```

Nout = sum(I==0);
if Nout>0
    disp( ['Warnings: ' num2str(Nout) ' outliers found! (cc='
          num2str(cc) ', qm=' num2str(qm) ', qp=' num2str(qp)
          ', qr=' num2str(qr) ', qc=' num2str(qc)
          ', qs=' num2str(qs) ']' )
end
err_posizione_medio(cc,qm,qp,qr,qc,qs) =
    mean(err_posizione(cc,qm,qp,qr,qc,qs,I));
err_orientamento_medio(cc,qm,qp,qr,qc,qs) =
    mean(err_orientamento(cc,qm,qp,qr,qc,qs,I));
err_intensita_medio(cc,qm,qp,qr,qc,qs) =
    mean(err_intensita(cc,qm,qp,qr,qc,qs,I));
err_posizione_std(cc,qm,qp,qr,qc,qs) =
    std(err_posizione(cc,qm,qp,qr,qc,qs,I));
err_orientamento_std(cc,qm,qp,qr,qc,qs) =
    std(err_orientamento(cc,qm,qp,qr,qc,qs,I));
err_intensita_std(cc,qm,qp,qr,qc,qs) =
    std(err_intensita(cc,qm,qp,qr,qc,qs,I));
end
end
end
end
end
end
toc

colors = ['k','r','g','m'];

% Rumore magnetometro (no errore posizione) :
if length(sigmas_mag)>1
    figure
    hold on
    leg=[];
    for cc=1:length(configs)
        errorbar(sigmas_mag*1e9, err_posizione_medio(cc,:,1,1,1,1)*1e3,
                 err_posizione_std(cc,:,1,1,1,1)*1e3, [colors(cc) '-']);
        leg{cc} = ['Nmags = ', num2str(Nmags(cc))];
    end
    grid on
    legend(leg,'Location','northwest');
    xlabel('Noise, (nT)')
    ylabel('Distance Error, (mm)')

    figure
    hold on
    leg=[];
    for cc=1:length(configs)
        errorbar(sigmas_mag*1e9, err_orientamento_medio(cc,:,1,1,1,1),
                 err_orientamento_std(cc,:,1,1,1,1), [colors(cc) '-']);
        leg{cc} = ['Nmags = ', num2str(Nmags(cc))];
    end
    grid on
    legend(leg,'Location','northwest');
    xlabel('Noise, (nT)')
    ylabel('Orientation Error, (deg)')

    figure
    hold on
    leg=[];
    for cc=1:length(configs)
        errorbar(sigmas_mag*1e9, err_intensita_medio(cc,:,1,1,1,1),
                 err_intensita_std(cc,:,1,1,1,1), [colors(cc) '-']);
        leg{cc} = ['Nmags = ', num2str(Nmags(cc))];
    end
    grid on

```

```

    legend(leg, 'Location', 'northwest');
    xlabel('Noise, (nT)')
    ylabel('Strength Error, %')
end

% Errore posizione (no rumore magnetometro) :
if length(sigmas_pos)>1
    figure
    hold on
    leg=[];
    for cc=1:length(configs)
        errorbar(sigmas_pos*1e3, err_posizione_medio(cc,1, :,1,1,1)*1e3,
                err_posizione_std(cc,1, :,1,1,1)*1e3, [colors(cc) '-']);
        leg{cc} = ['Nmags = ', num2str(Nmags(cc))];
    end
    grid on
    legend(leg, 'Location', 'northwest');
    xlabel('Position Error, (mm)')
    ylabel('Distance Error, (mm)')

    figure
    hold on
    leg=[];
    for cc=1:length(configs)
        errorbar(sigmas_pos*1e3, err_orientamento_medio(cc,1, :,1,1,1),
                err_orientamento_std(cc,1, :,1,1,1), [colors(cc) '-']);
        leg{cc} = ['Nmags = ', num2str(Nmags(cc))];
    end
    grid on
    legend(leg, 'Location', 'northwest');
    xlabel('Position Error, (mm)')
    ylabel('Orientation Error, (deg)')

    figure
    hold on
    leg=[];
    for cc=1:length(configs)
        errorbar(sigmas_pos*1e3, err_intensita_medio(cc,1, :,1,1,1),
                err_intensita_std(cc,1, :,1,1,1), [colors(cc) '-']);
        leg{cc} = ['Nmags = ', num2str(Nmags(cc))];
    end
    grid on
    legend(leg, 'Location', 'northwest');
    xlabel('Position Error, (mm)')
    ylabel('Strength Error, %')
end

% Errore rotazione (no rumore magnetometro & errore posizione) :
if length(sigmas_rot)>1
    figure
    hold on
    leg=[];
    for cc=1:length(configs)
        errorbar(sigmas_rot, err_posizione_medio(cc,1,1, :,1,1)*1e3,
                err_posizione_std(cc,1,1, :,1,1)*1e3, [colors(cc) '-']);
        leg{cc} = ['Nmags = ', num2str(Nmags(cc))];
    end
    grid on
    legend(leg, 'Location', 'northwest');
    xlabel('Rotation Error, (deg)')
    ylabel('Distance Error, (mm)')

    figure
    hold on
    leg=[];
    for cc=1:length(configs)
        errorbar(sigmas_rot, err_orientamento_medio(cc,1,1, :,1,1),
                err_orientamento_std(cc,1,1, :,1,1), [colors(cc) '-']);

```

```

        leg{cc} = ['Nmags = ', num2str(Nmags(cc))];
    end
    grid on
    legend(leg, 'Location', 'northwest');
    xlabel('Rotation Error, (deg)')
    ylabel('Orientation Error, (deg)')

    figure
    hold on
    leg=[];
    for cc=1:length(configs)
        errorbar(sigmas_rot, err_intensita_medio(cc,1,1, :,1,1),
                err_intensita_std(cc,1,1, :,1,1), [colors(cc) '-']);
        leg{cc} = ['Nmags = ', num2str(Nmags(cc))];
    end
    grid on
    legend(leg, 'Location', 'northwest');
    xlabel('Rotation Error, (deg)')
    ylabel('Strength Error, %')
end

% Errore bias calibrazione (no rumore magnetometro && errore posizione && errore
rotazione) :
if length(sigmas_bias_calibration)>1
    figure
    hold on
    leg=[];
    for cc=1:length(configs)
        errorbar(sigmas_bias_calibration*1e6, err_posizione_medio(cc,1,1,1, :,1)*1e3,
                err_posizione_std(cc,1,1,1, :,1)*1e3, [colors(cc) '-']);
        leg{cc} = ['Nmags = ', num2str(Nmags(cc))];
    end
    grid on
    legend(leg, 'Location', 'northwest');
    xlabel('Calibration Bias, (\muT)')
    ylabel('Distance Error, (mm)')

    figure
    hold on
    leg=[];
    for cc=1:length(configs)
        errorbar(sigmas_bias_calibration*1e6, err_orientamento_medio(cc,1,1,1, :,1),
                err_orientamento_std(cc,1,1,1, :,1), [colors(cc) '-']);
        leg{cc} = ['Nmags = ', num2str(Nmags(cc))];
    end
    grid on
    legend(leg, 'Location', 'northwest');
    xlabel('Calibration Bias, (\muT)')
    ylabel('Orientation Error, (deg)')

    figure
    hold on
    leg=[];
    for cc=1:length(configs)
        errorbar(sigmas_bias_calibration*1e6, err_intensita_medio(cc,1,1,1, :,1),
                err_intensita_std(cc,1,1,1, :,1), [colors(cc) '-']);
        leg{cc} = ['Nmags = ', num2str(Nmags(cc))];
    end
    grid on
    legend(leg, 'Location', 'northwest');
    xlabel('Calibration Bias, (\muT)')
    ylabel('Strength Error, %')
end
end

```

```

% Errore scaling calibrazione (no rumore magnetometro && errore posizione && errore
rotazione) :
if length(sigmas_scaling_factors)>1
    figure
    hold on
    leg=[];
    for cc=1:length(configs)
        errorbar(sigmas_scaling_factors, err_posizione_medio(cc,1,1,1,1,:)*1e3,
                err_posizione_std(cc,1,1,1,1,:)*1e3, [colors(cc) '-']);
        leg{cc} = ['Nmags = ', num2str(Nmags(cc))];
    end
    grid on
    legend(leg,'Location','northwest');
    xlabel('Scaling Factor, %')
    ylabel('Distance Error, (mm)')

    figure
    hold on
    leg=[];
    for cc=1:length(configs)
        errorbar(sigmas_scaling_factors, err_orientamento_medio(cc,1,1,1,1,:),
                err_orientamento_std(cc,1,1,1,1,:), [colors(cc) '-']);
        leg{cc} = ['Nmags = ', num2str(Nmags(cc))];
    end
    grid on
    legend(leg,'Location','northwest');
    xlabel('Scaling Factor, %')
    ylabel('Orientation Error, (deg)')

    figure
    hold on
    leg=[];
    for cc=1:length(configs)
        errorbar(sigmas_scaling_factors, err_intensita_medio(cc,1,1,1,1,:),
                err_intensita_std(cc,1,1,1,1,:), [colors(cc) '-']);
        leg{cc} = ['Nmags = ', num2str(Nmags(cc))];
    end
    grid on
    legend(leg,'Location','northwest');
    xlabel('Scaling Factor, %')
    ylabel('Strength Error, %')
end
end

```

## A.2 fmincon\_norm :

```
function [F,J] = fmincon_norm_MZ(gnorm)

global Bx_meas_noise By_meas_noise Bz_meas_noise Pos_meas x0 y0 z0 Sx0 Sy0 Sz0

% De-normalize inputs
g = [gnorm(1)*x0, gnorm(2)*y0, gnorm(3)*z0, gnorm(4)*Sx0, gnorm(5)*Sy0, gnorm(6)*Sz0];

B0 = max( [max(abs(Bx_meas_noise)), max(abs(By_meas_noise)), max(abs(Bz_meas_noise))] );

F = zeros(3*length(Bx_meas_noise),1);

count = 0;
for w = 1:length(Bx_meas_noise)

    [Bx, By, Bz] = bfield_dipole_MZ(Pos_meas(w,:), [g(1), g(2), g(3)], [g(4), g(5),
g(6)]);

    count = count + 1;
    F(count) = Bx/B0-Bx_meas_noise(w)/B0;

    count = count +1;
    F(count) = By/B0-By_meas_noise(w)/B0;

    count = count +1;
    F(count) = Bz/B0-Bz_meas_noise(w)/B0;

end

if nargin>1
    mu0 = 4*pi*1e-7;
    J = zeros(1,length(g));

    for w = 1:length(Bx_meas_noise)

        matrice jacobiana (non mostrata data l'eccessiva dimensione della sua
        espressione simbolica)

    end
end

F = sum(F.^2);
```

## A.3 progress\_bar.m :

```
function progress_bar(cd,Ntot,step)
if cd==1
    disp('0%')
elseif cd==Ntot
    disp('100%')
else
    % Percentage
    pd = 100*cd/Ntot;
    % Next
    pdn = 100*(cd+1)/Ntot;
    % Nearest percentage to show
    ps = round(pd/step)*step;
    if pd<=ps && pdn>ps
        disp([num2str(ps) '%'])
    end
end
```

## A.4 bfield\_dipole.m :

```
function [Bx,By,Bz] = bfield_dipole(pos1, pos0, dip)

x=pos1(1);
y=pos1(2);
z=pos1(3);

a=pos0(1);
b=pos0(2);
c=pos0(3);

M=dip(1);
N=dip(2);
P=dip(3);

mu0 = 4*pi*1e-7;      % H/m
mu0_4pi = 1e-7;
R = sqrt( (x-a)^2 + (y-b)^2 + (z-c)^2 );

Bx = mu0_4pi*( 3*( (x-a)*M + (y-b)*N + (z-c)*P )*(x-a)/R^5 - M/R^3 );
By = mu0_4pi*( 3*( (x-a)*M + (y-b)*N + (z-c)*P )*(y-b)/R^5 - N/R^3 );
Bz = mu0_4pi*( 3*( (x-a)*M + (y-b)*N + (z-c)*P )*(z-c)/R^5 - P/R^3 );
```

## A.5 remove\_outliers.m :

```
function I=remove_outliers(x,k)

imax=100;

I = x==x;
for i=1:imax
    % Max and sigma without outliers
    x_mean = mean(x(I));
    x_sigma = std(x(I));

    % Values outside mean +/- k*sigma
    I0 = I;
    I = abs(x-x_mean)<=k*x_sigma;
    if all(I0==I)
        break
    end
end

% Second method
for i=1:imax
    % Max current value (wrt mean)
    dx_max = max( abs(x(I)-x_mean) );

    % Remove the max value
    I2 = abs(x-x_mean)<dx_max;
    if all(I2==0)
        break
    end
    % Mean and sigma without max value
    x_mean2 = mean(x(I & I2));
    x_sigma2 = std(x(I & I2));

    % Values outside mean2 +/- k*sigma2
    I0 = I;
    I = abs(x-x_mean2)<=k*x_sigma2;
```

```
    if all(I0==I)
        break
    end
end

if i==imax
    disp('Maximum iterations reached!')
end
```

## APPENDICE B

### Scheda tecnica magnetometro AP 539 :

PHYSICAL	
Width	1.6" (40.64 mm)
Height	1.125" (28.575 mm)
Length	4.08" (103.632 mm)
Weight	150 g
Input Connections (wire option)	9-pin nonmagnetic "D" (Female)

ELECTRICAL	
Power input	50 ma @ +4.95 to +9 VDC
Analog to Digital	16-bit Sigma Delta
Baud rate (user selectable)	300, 1200, 2400, 4800, 9600, 19200, 38400
Maximum data transfer speed (38.4k baud)	250 3-axis samples/sec

ENVIRONMENTAL	
Operating Temperature	-25°C to +70°C
Accuracy	±1 % Full Scale
Noise Level	0.3 nT RMS/Hz <sup>1/2</sup> 3 μG RMS/Hz <sup>1/2</sup>
Range	±65 μT (±0.65 G), ±100 μT optional
Scale Stability	0.05% Full Scale/°C
Initial Offset	<± 200 nT (±2 mG)
Offset vs. Temperature	<5 nT/°C (<0.05 mG)
Orthogonality of Axes	Better than ± 0.5°
Alignment of Axes with Package	Better than ± 0.5°
Linearity	± 0.1% full scale

Tratta dal sito internet <http://www.appliedphysics.com/> .