

Febbraio – Aprile 2021

Progetto Myki

Compensazione disturbi
meccanici

Supervisors:

- Christian Cipriani
- Marta Gherardini
- Valerio Ianniciello

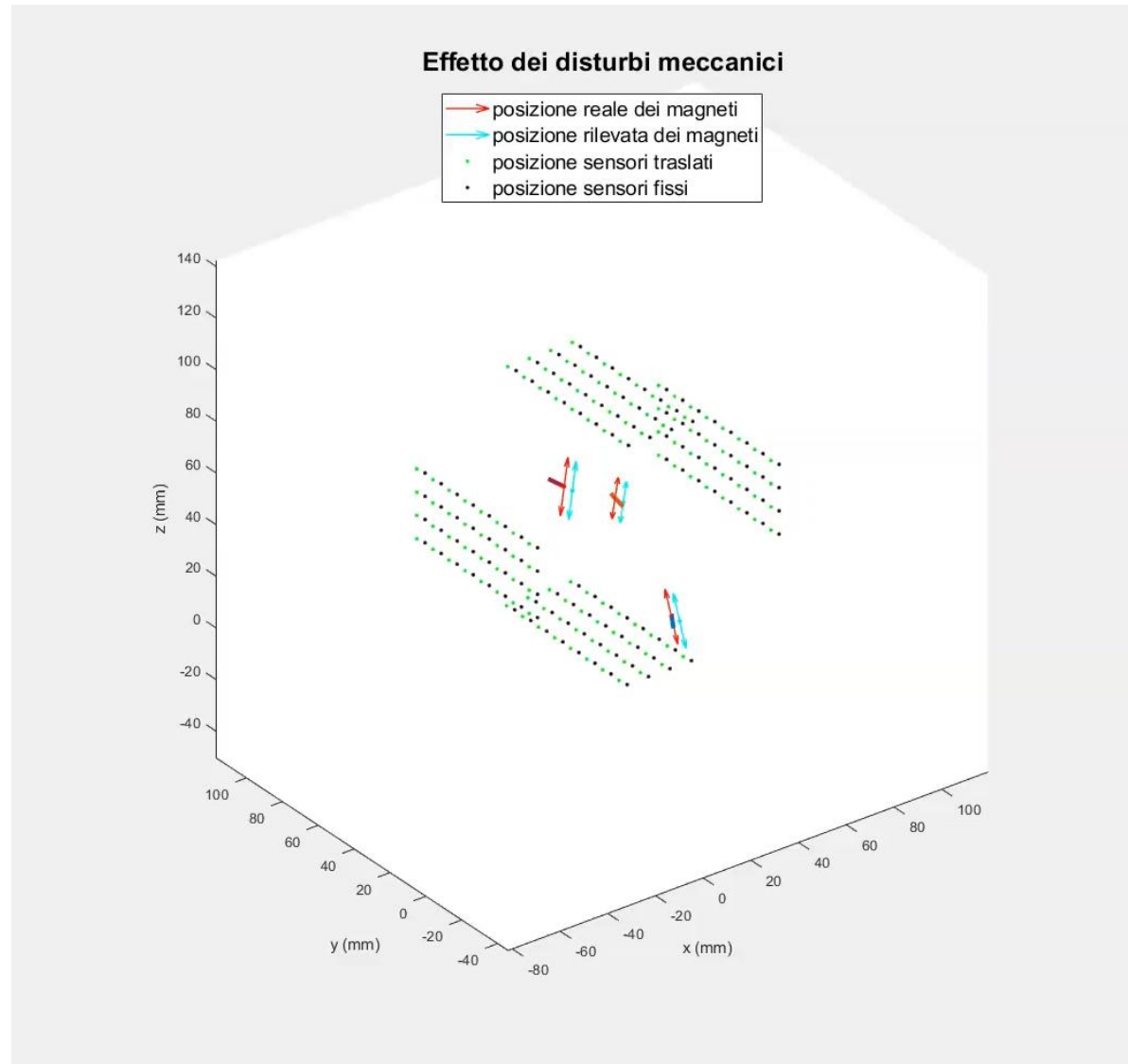
Obiettivo

Eliminare errori di rilevazione dovuti allo spostamento della protesi rispetto al braccio



Disturbi meccanici

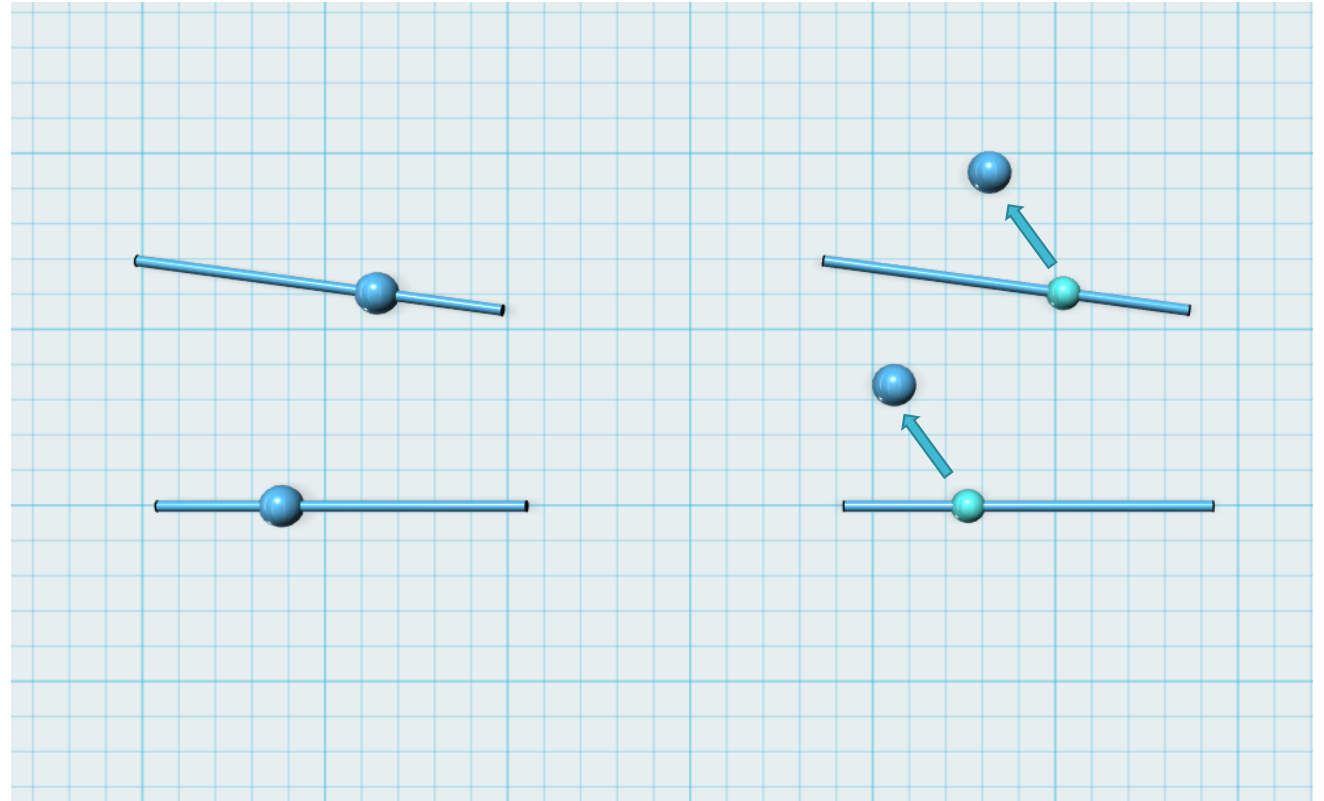
Se i sensori si muovono, alla
protesi sembra che i magneti
si stiano muovendo



Approccio alla soluzione

Ogni magnete si muove lungo una traiettoria

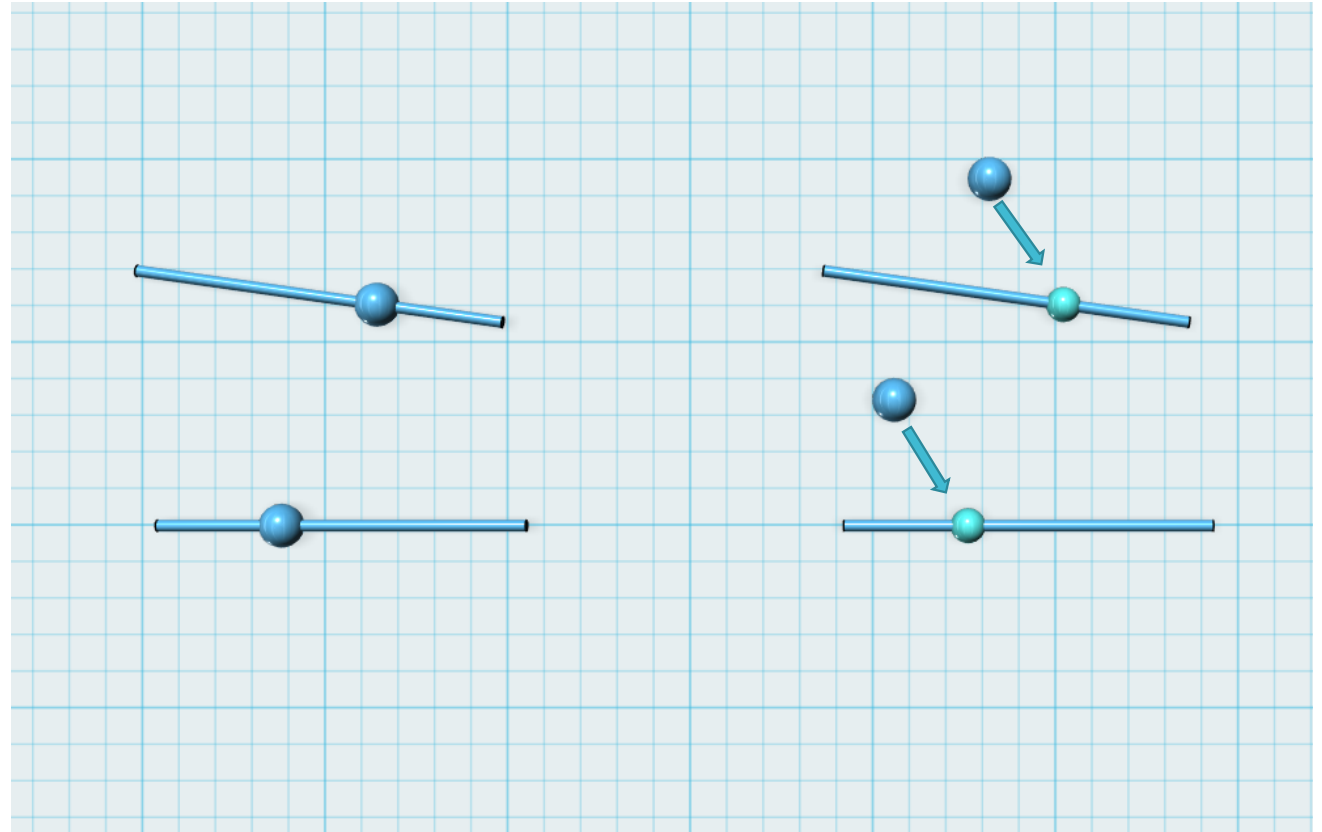
Se almeno un magnete si trova fuori dalla sua traiettoria, allora c'è stato un disturbo meccanico



Approccio alla soluzione

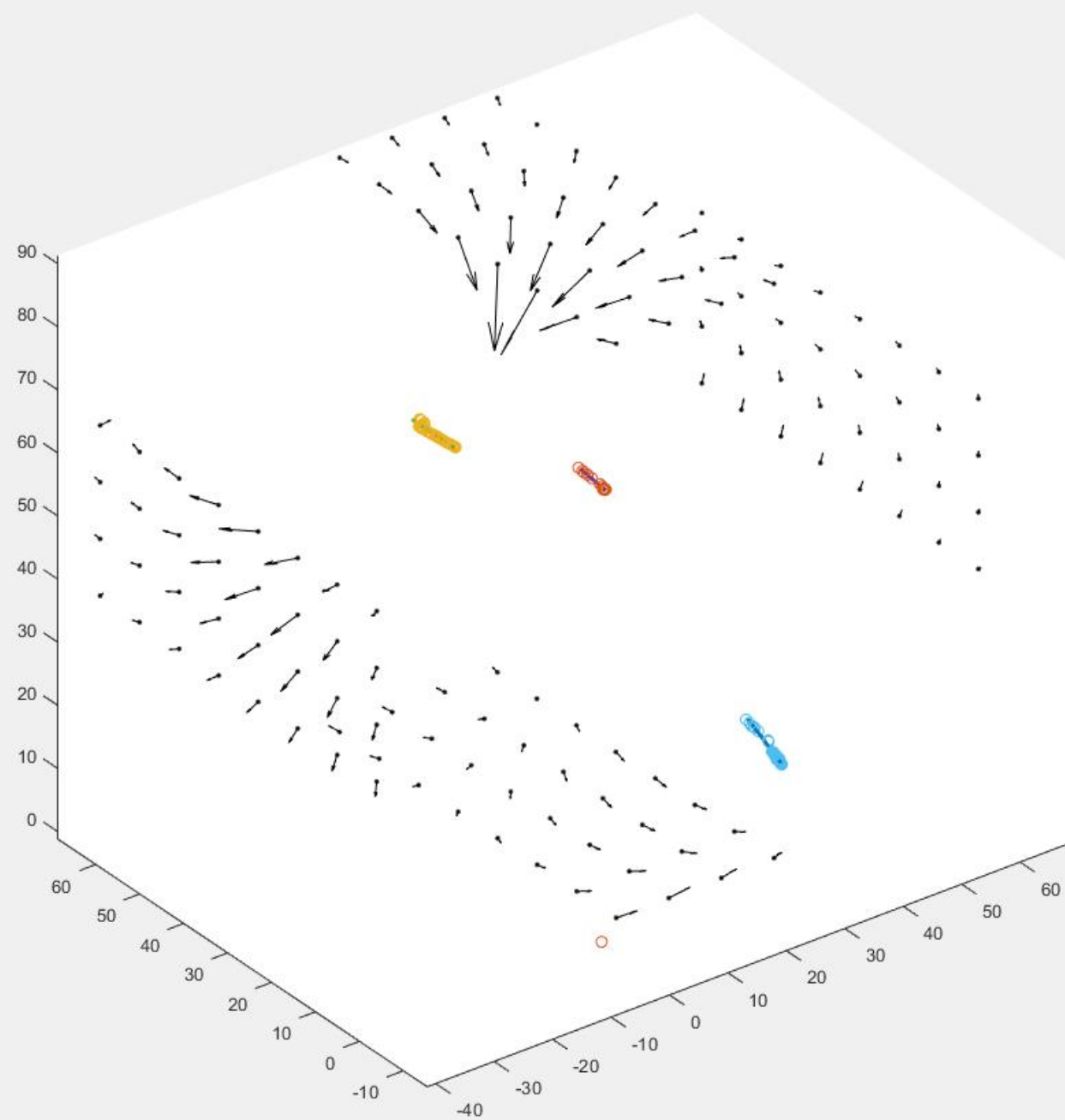
L'algoritmo cerca una rototraslazione dei magneti che li riporti tutti all'interno della loro traiettoria

Essendo le traiettorie non parallele tra loro, la soluzione è (salvo rari casi...) unica



Fase di setup

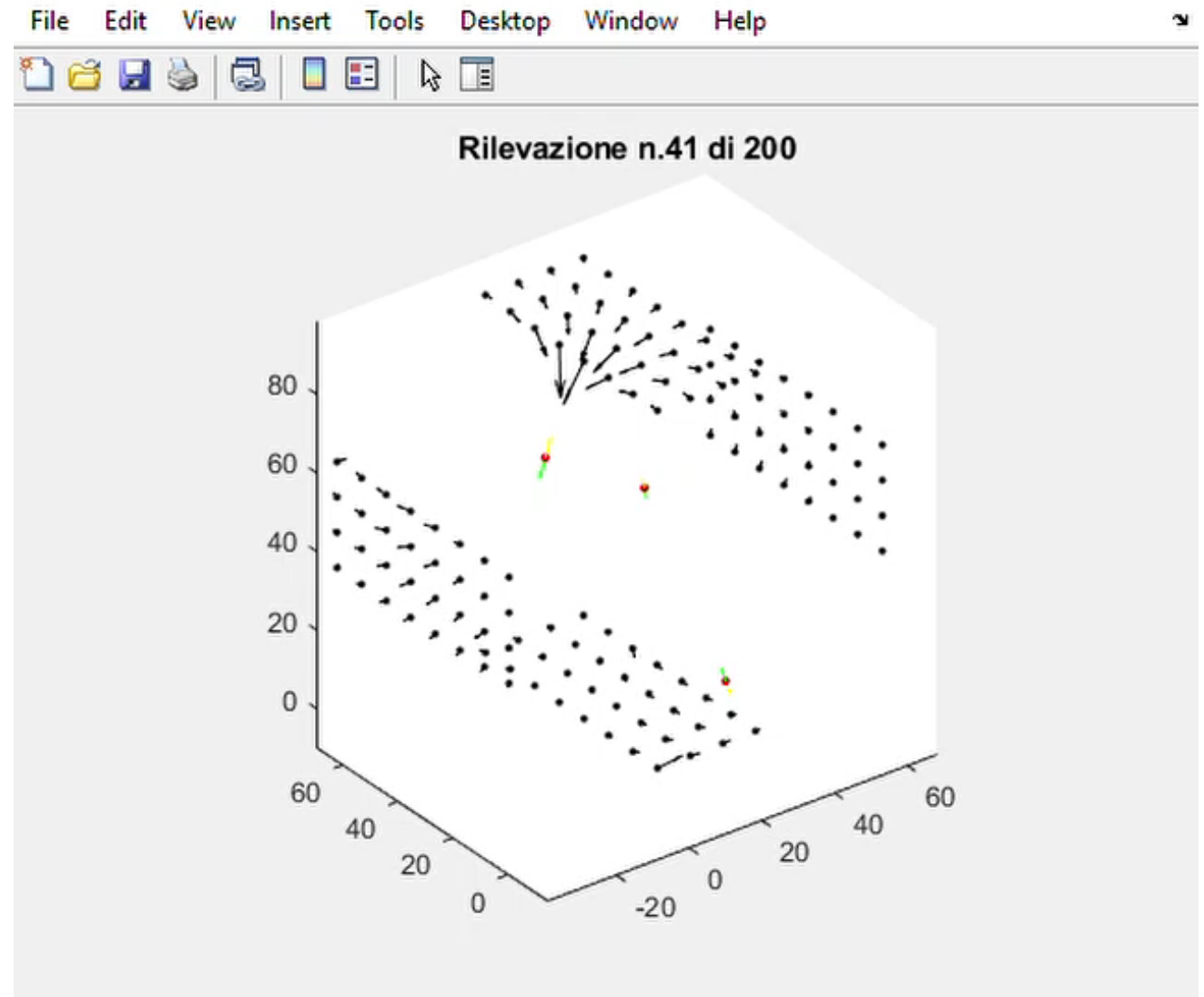
Ottenimento traiettorie



Ottenimento traiettorie

Il programma deve sapere dove si trovano i magneti in situazioni normali

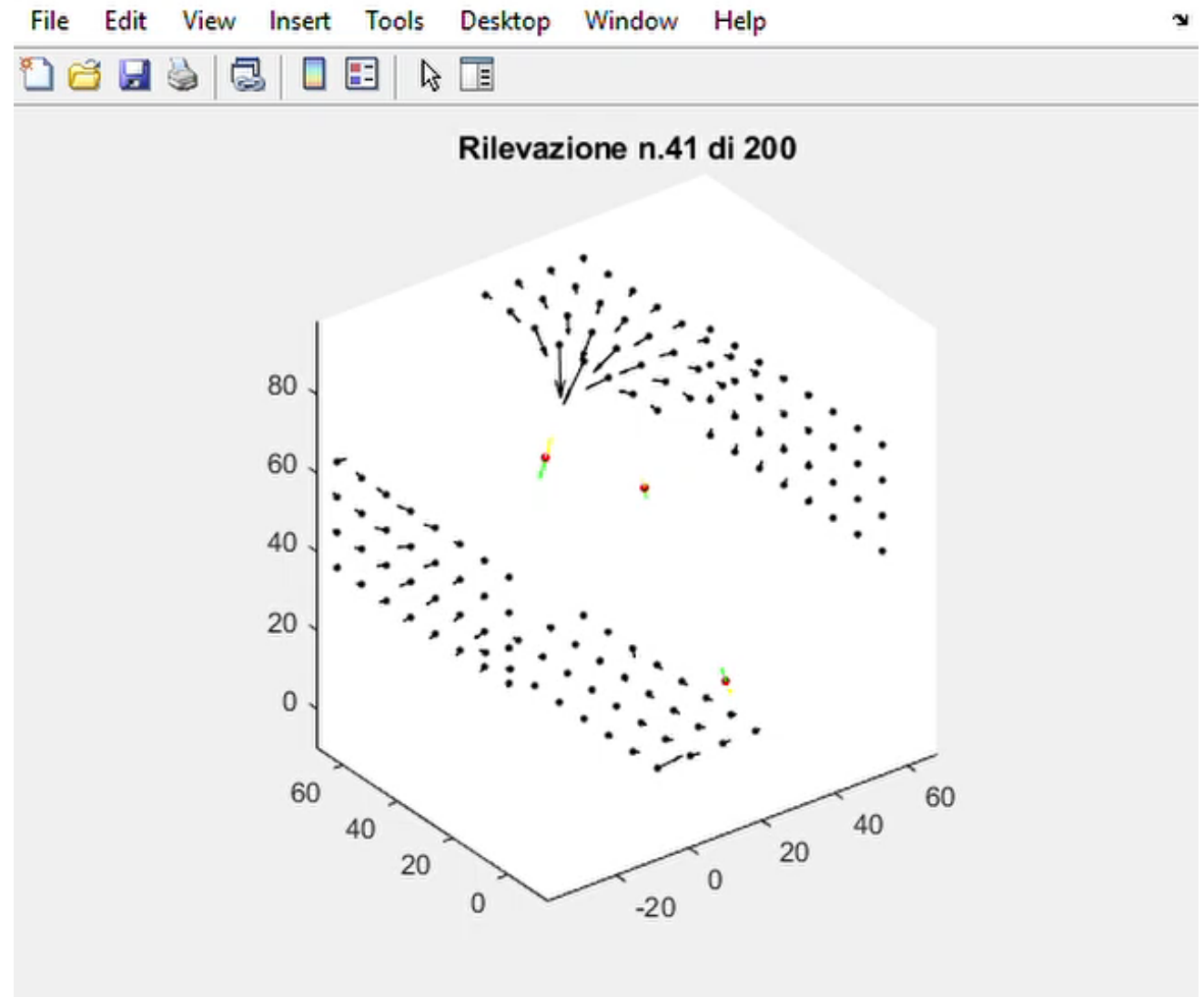
Vengono fatti muovere i magneti senza muovere il mockup e si registrano le posizioni



A livello pratico

Si chiede al paziente di contrarre i muscoli e si registrano le posizioni dei magneti

Si sta attenti che la protesi, nel frattempo, non si muova

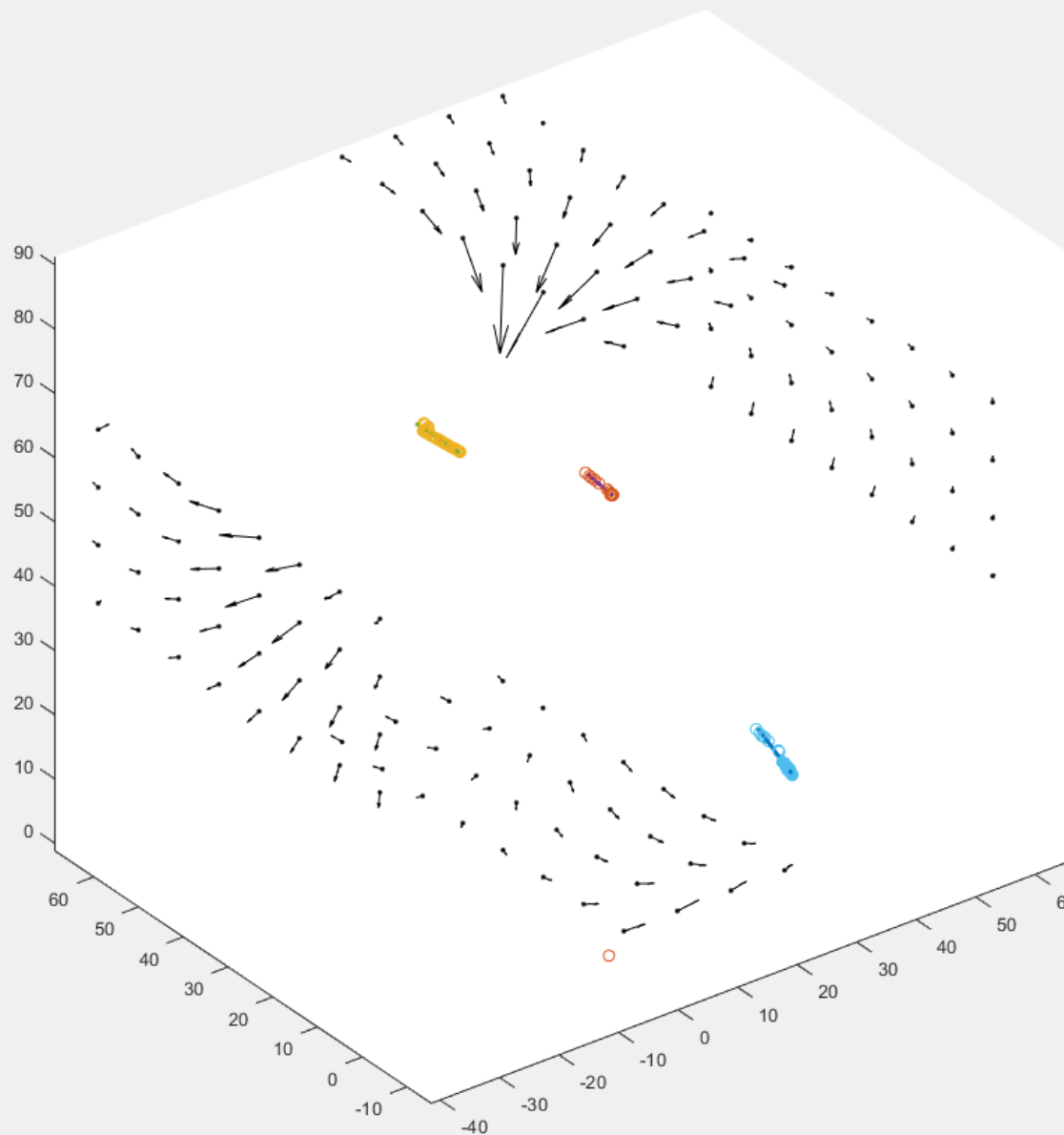


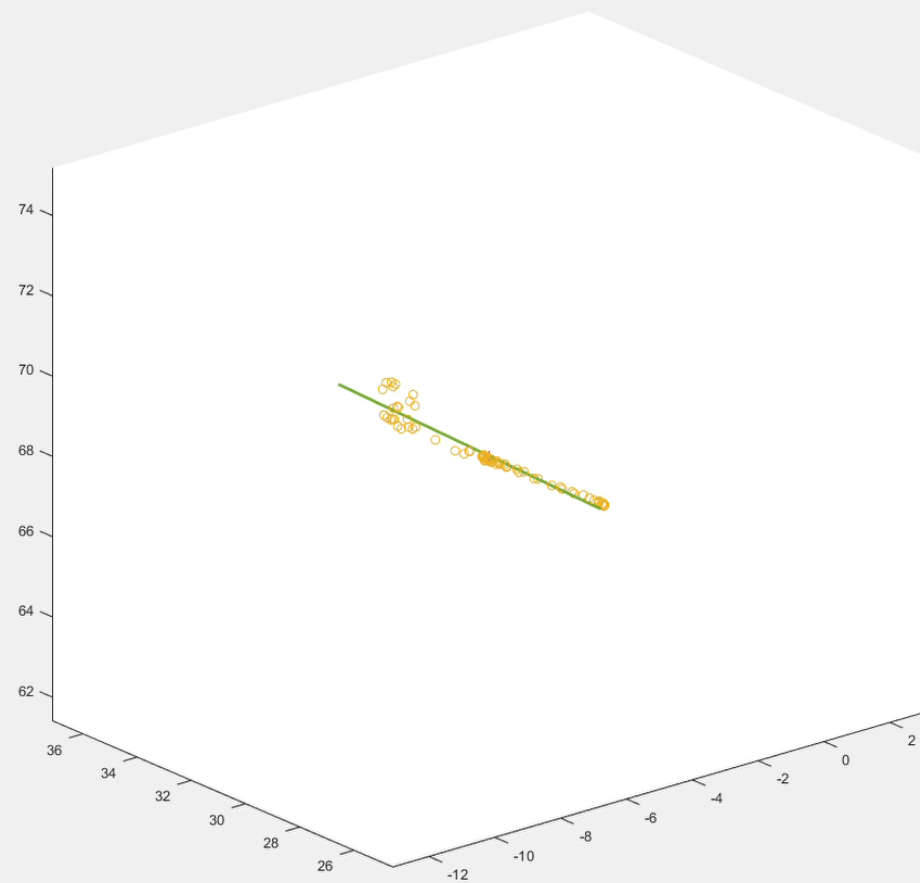
LINEAR FIT

Dopodiché, il programma attua un linear fit per ogni magnete

(trova il segmento che meglio approssima le posizioni dei magneti)

Ora, il programma conosce le traiettorie dei magneti



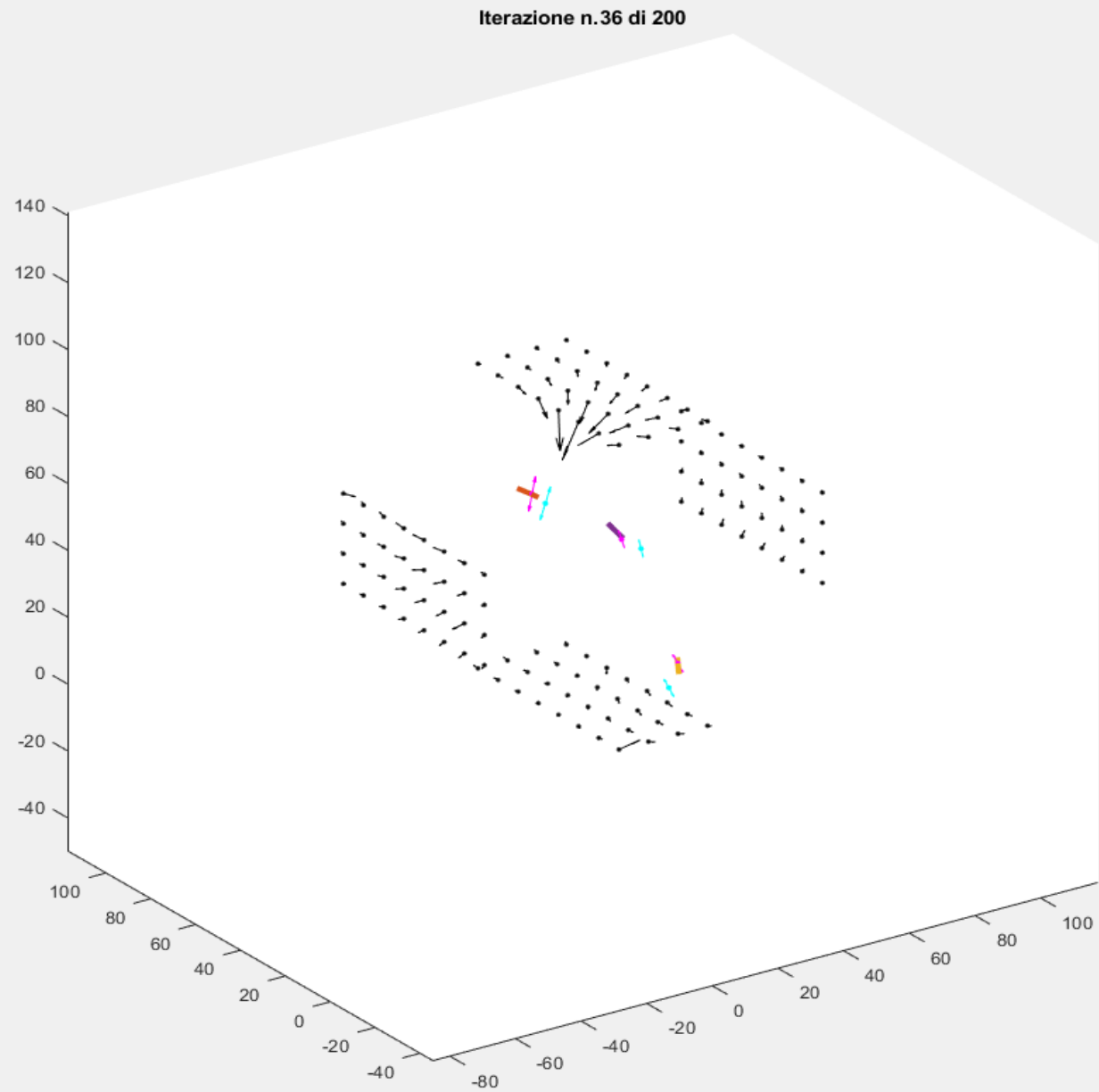


ASSUNZIONE

- I magneti percorrono traiettorie approssimabilmente rettilinee
- La validità di questa assunzione va verificata sperimentalmente su pazienti

Fase di Run

Compensazione Disturbi
Meccanici



Fase di Run

Questa parte del programma simula l'effettivo funzionamento della protesi:

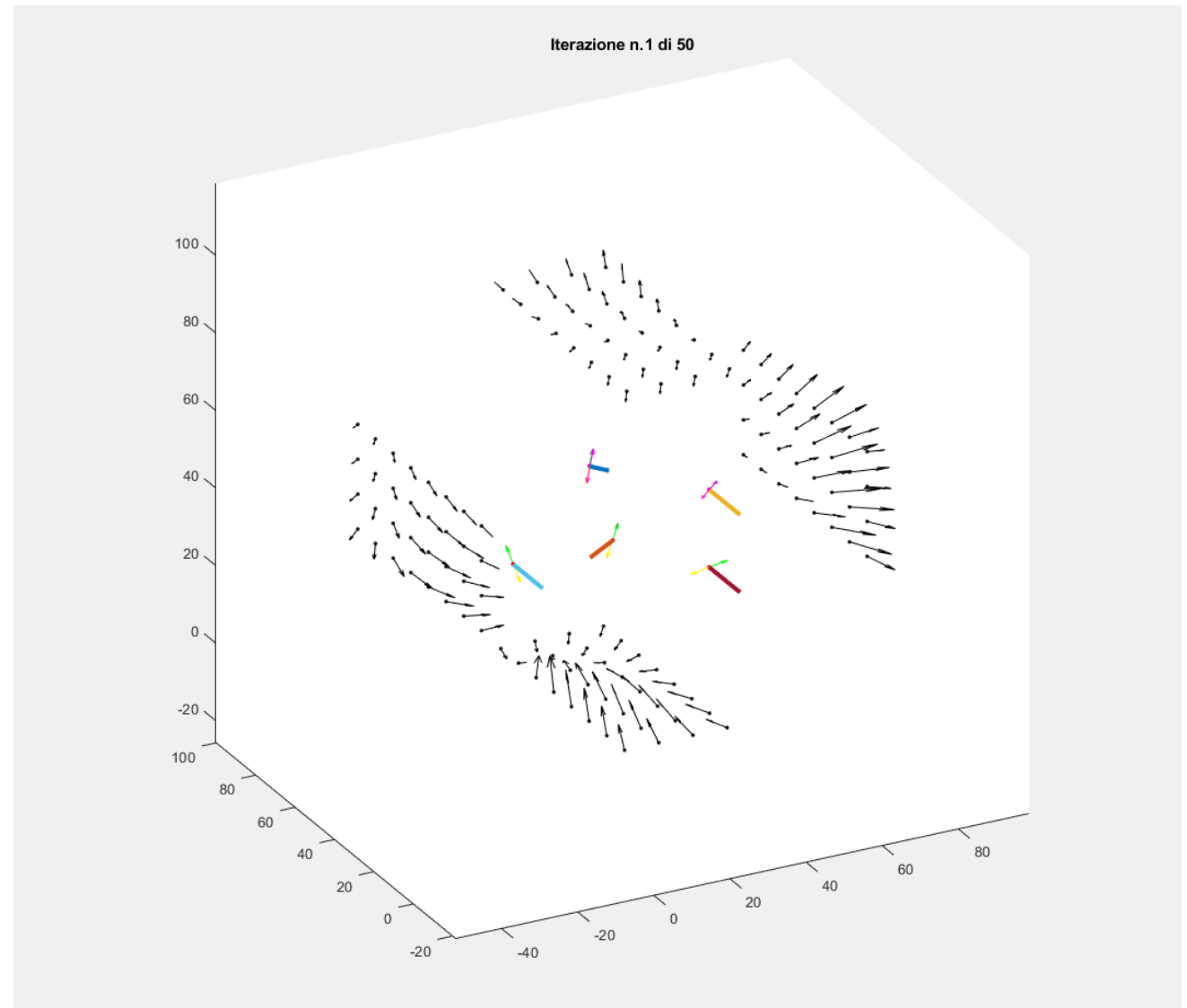
1. Riceve in input i valori del campo magnetico rilevato dai sensori
2. Ottiene la posizione dei magneti
3. Controlla se tutti si trovano all'interno della traiettoria
4. Se la risposta è NO, cerca la rototraslazione che riporta ogni magnete nella sua traiettoria
5. Eliminazione rumore delle misurazioni

1. Input dati

Inizialmente, posizioni dei magneti e disturbi esterni simulati da software

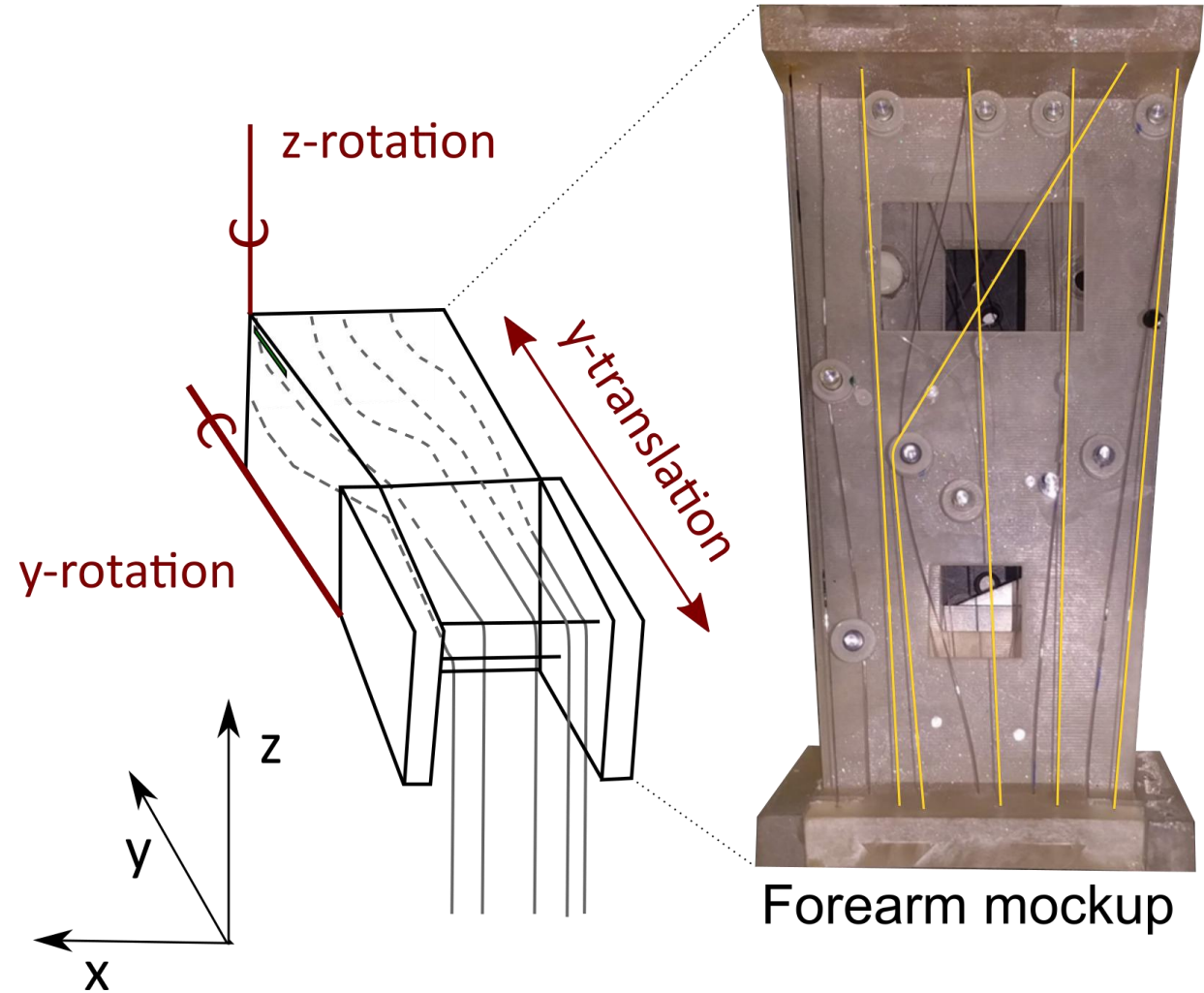
Necessario per:

1. capire come e quanto bene il programma stia funzionando
2. Capire in che situazioni il programma si comporta meglio



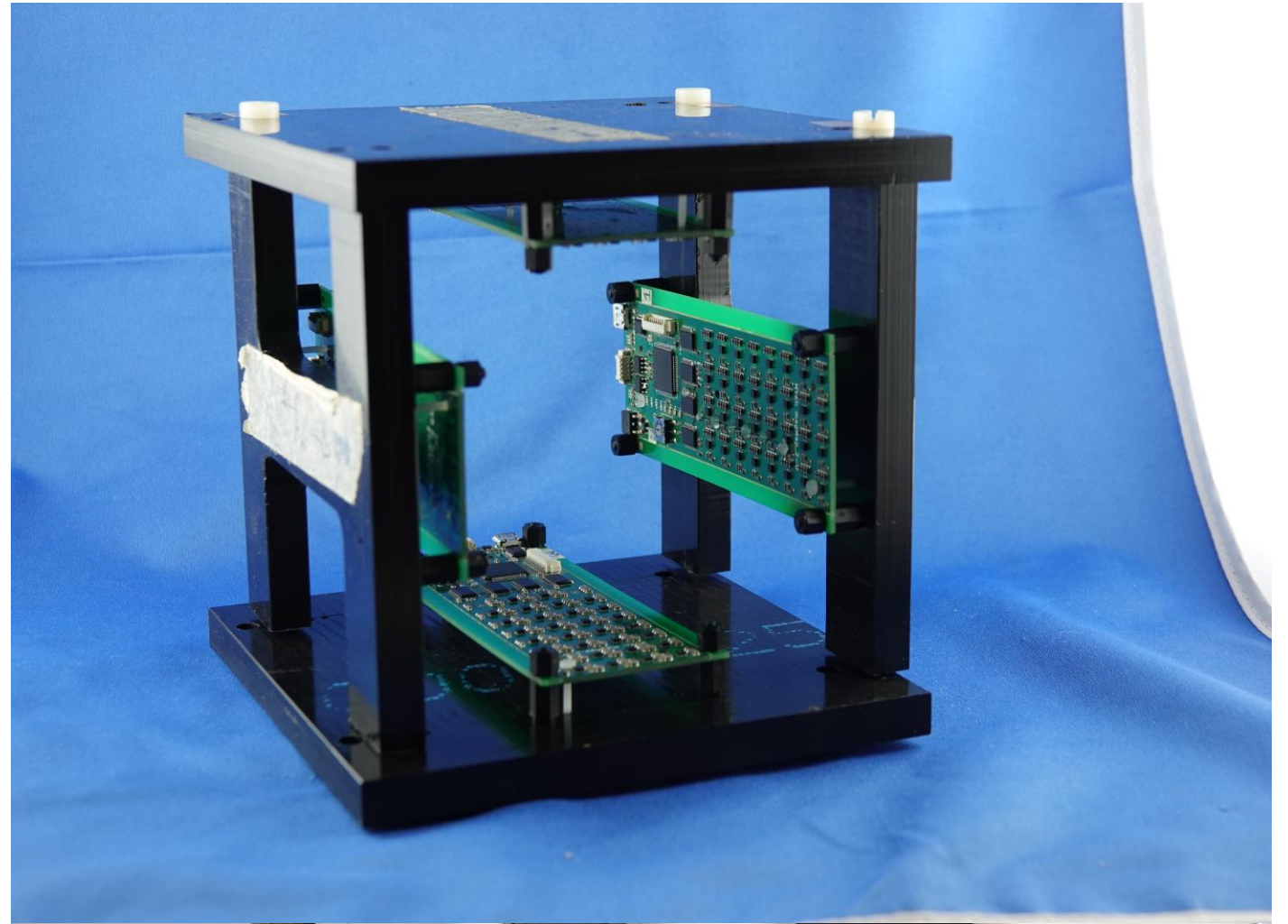
1. Input dati

Poi, una volta pronto l'algoritmo,
abbiamo raccolto dati
SPERIMENTALMENTE



Apparato sperimentale

1. un mockup di un avambraccio: dei servomotori fanno muovere dei fili su cui sono attaccati dei magneti
2. un mockup della protesi: contenente 4 schede con i sensori di campo magnetico (in totale: 128 sensori)



Fase di Run

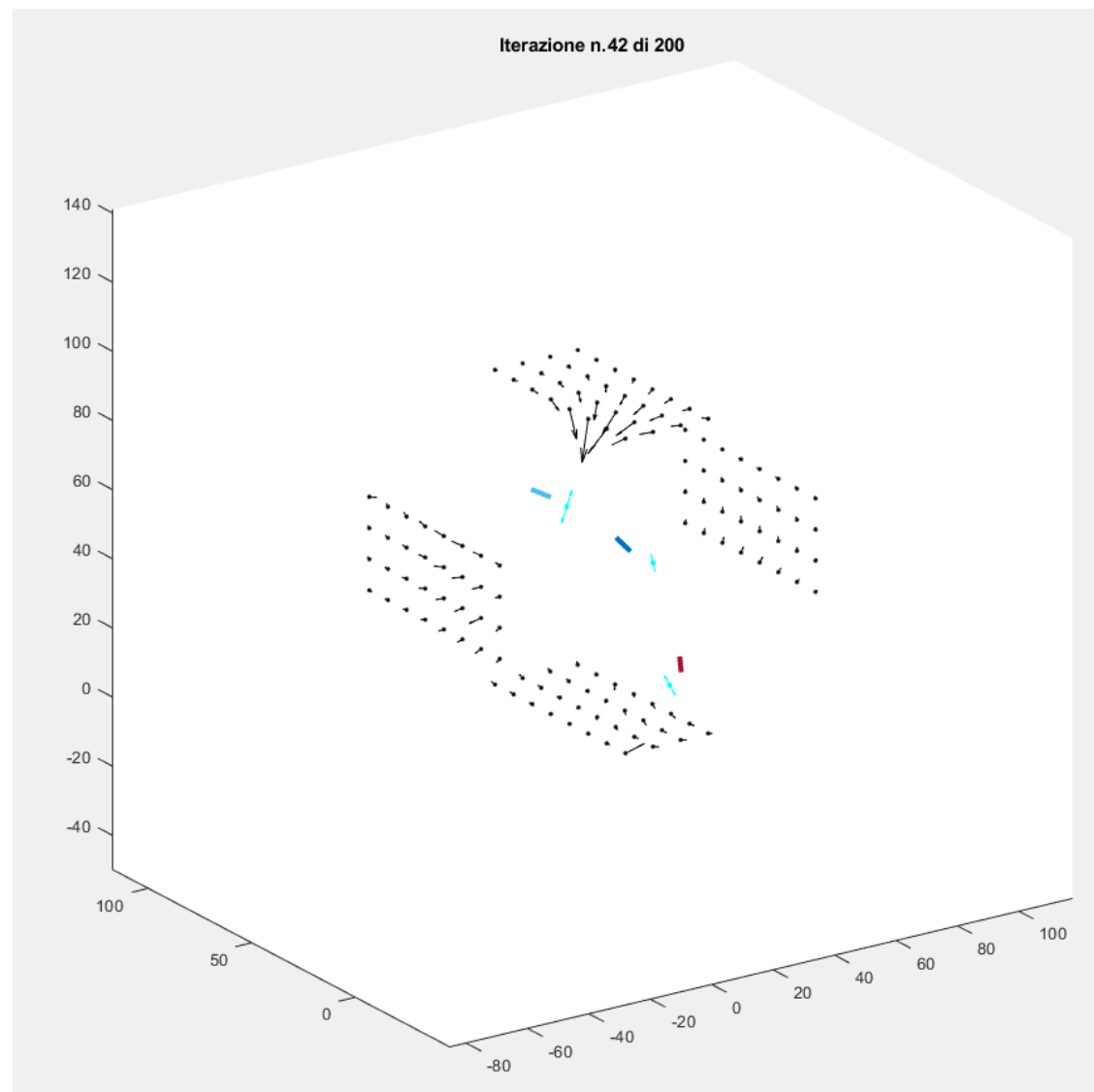
Questa parte del programma simula l'effettivo funzionamento della protesi:

1. Riceve in input i valori del campo magnetico rilevato dai sensori
2. Ottiene la posizione dei magneti
3. Controlla se tutti si trovano all'interno della traiettoria
4. Se la risposta è NO, cerca la rototraslazione che riporta ogni magnete nella sua traiettoria
5. Eliminazione rumore delle misurazioni

2. Ottenimento posizione dei magneti

(Lavoro svolto da Federico)

Dai valori del campo
magnetico, un algoritmo di
ottimizzazione trova la
posizione dei magneti



Fase di Run

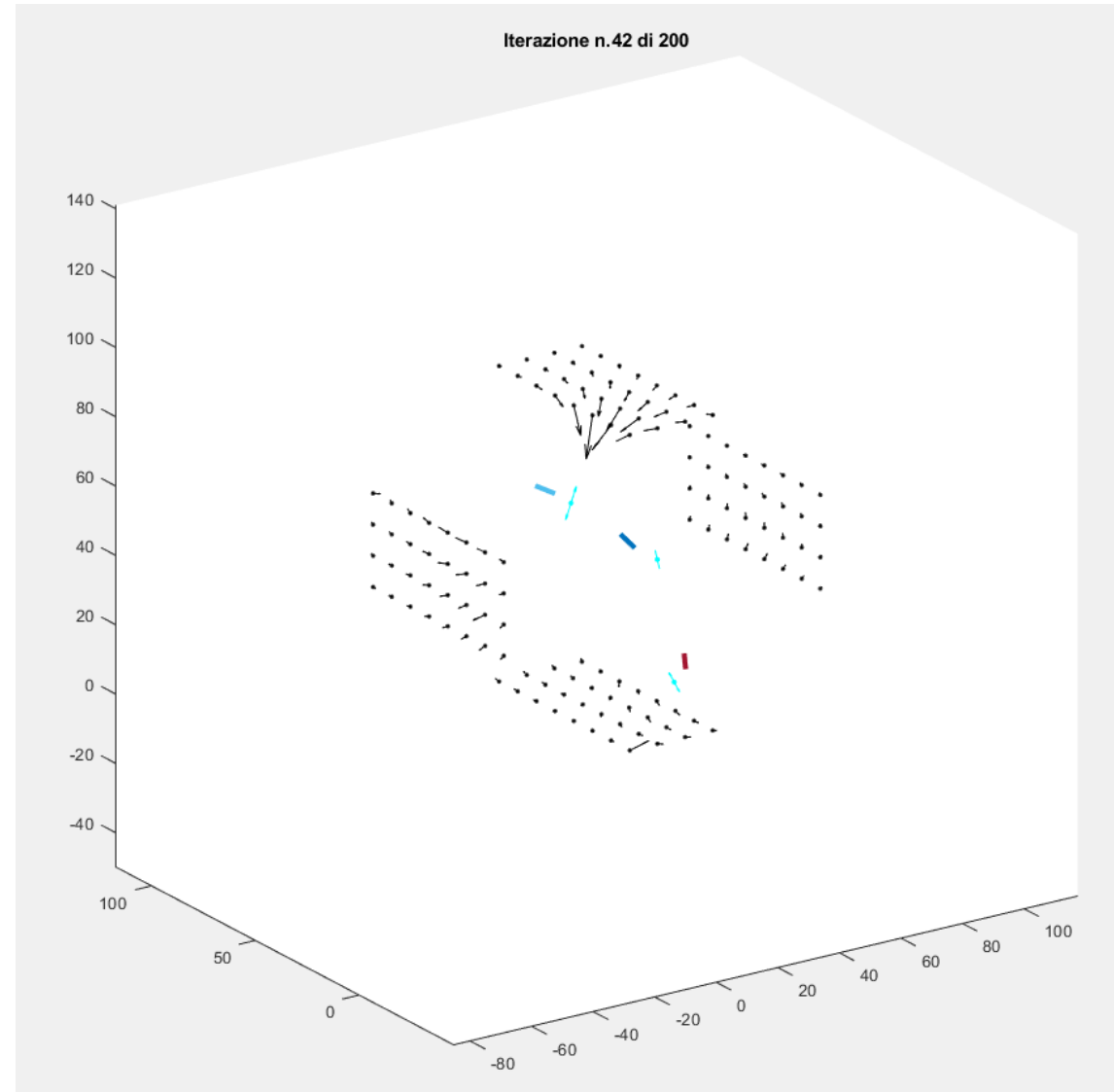
Questa parte del programma simula l'effettivo funzionamento della protesi:

1. Riceve in input i valori del campo magnetico rilevato dai sensori
2. Ottiene la posizione dei magneti
3. Controlla se tutti si trovano all'interno della traiettoria
4. Se la risposta è NO, cerca la rototraslazione che riporta ogni magnete nella sua traiettoria
5. Eliminazione rumore delle misurazioni

3. Controllo magneti nelle traiettorie

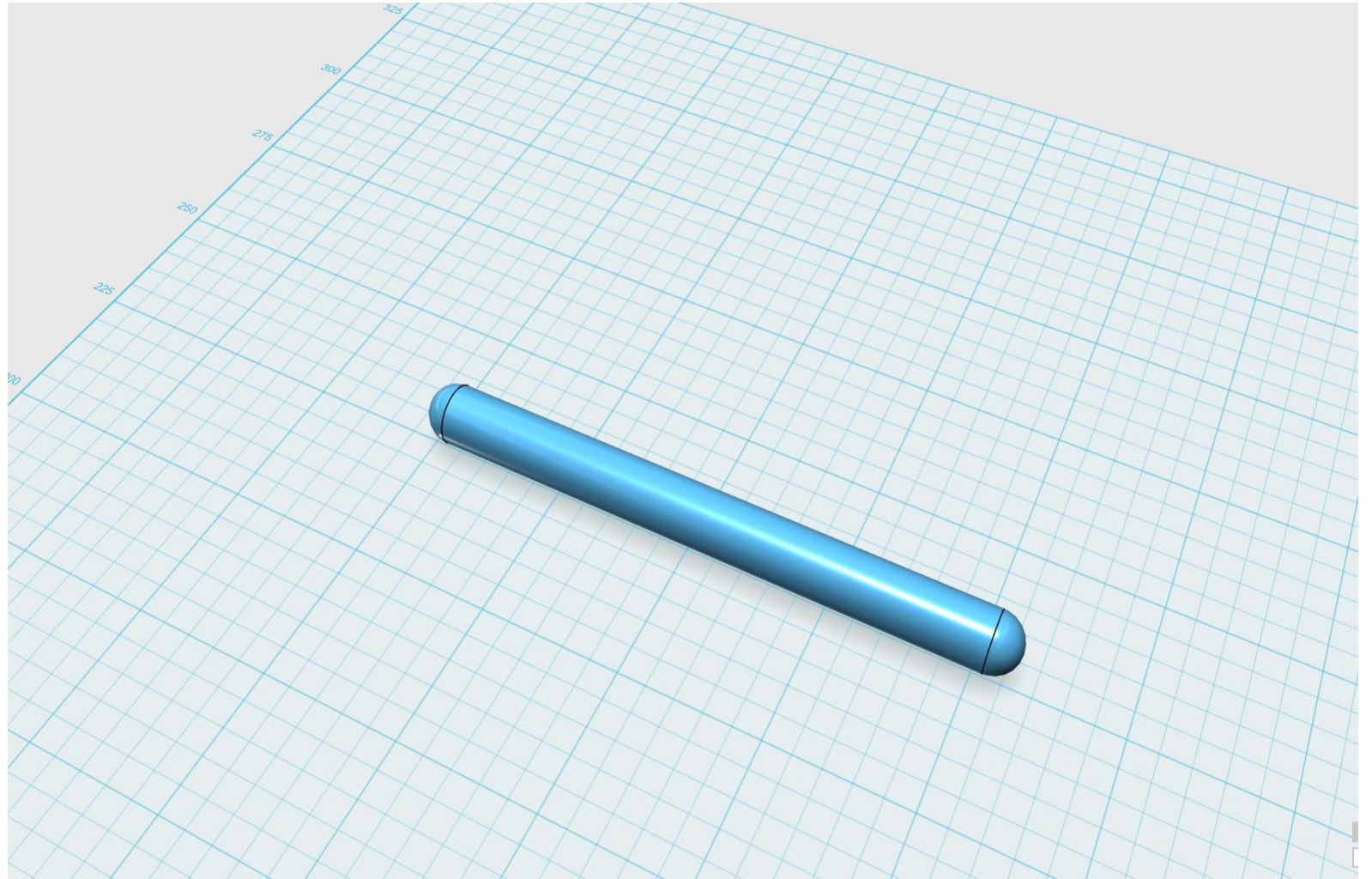
- Se tutti i magneti sono nelle traiettorie, nessun problema
- Se almeno un magnete si trova al di fuori, allora c'è stato un disturbo meccanico esterno.

Viene quindi attivato l'algoritmo di compensazione



Cos'è una traiettoria?

- errori di misurazione -> non può essere un semplice segmento
- Traiettoria modellizzata come un cilindro con ai capi due emisfere
- Raggio "errMax" = 1 mm (variabile)
- Un magnete si trova nella sua traiettoria se il suo centro si trova all'interno di questa sezione di spazio



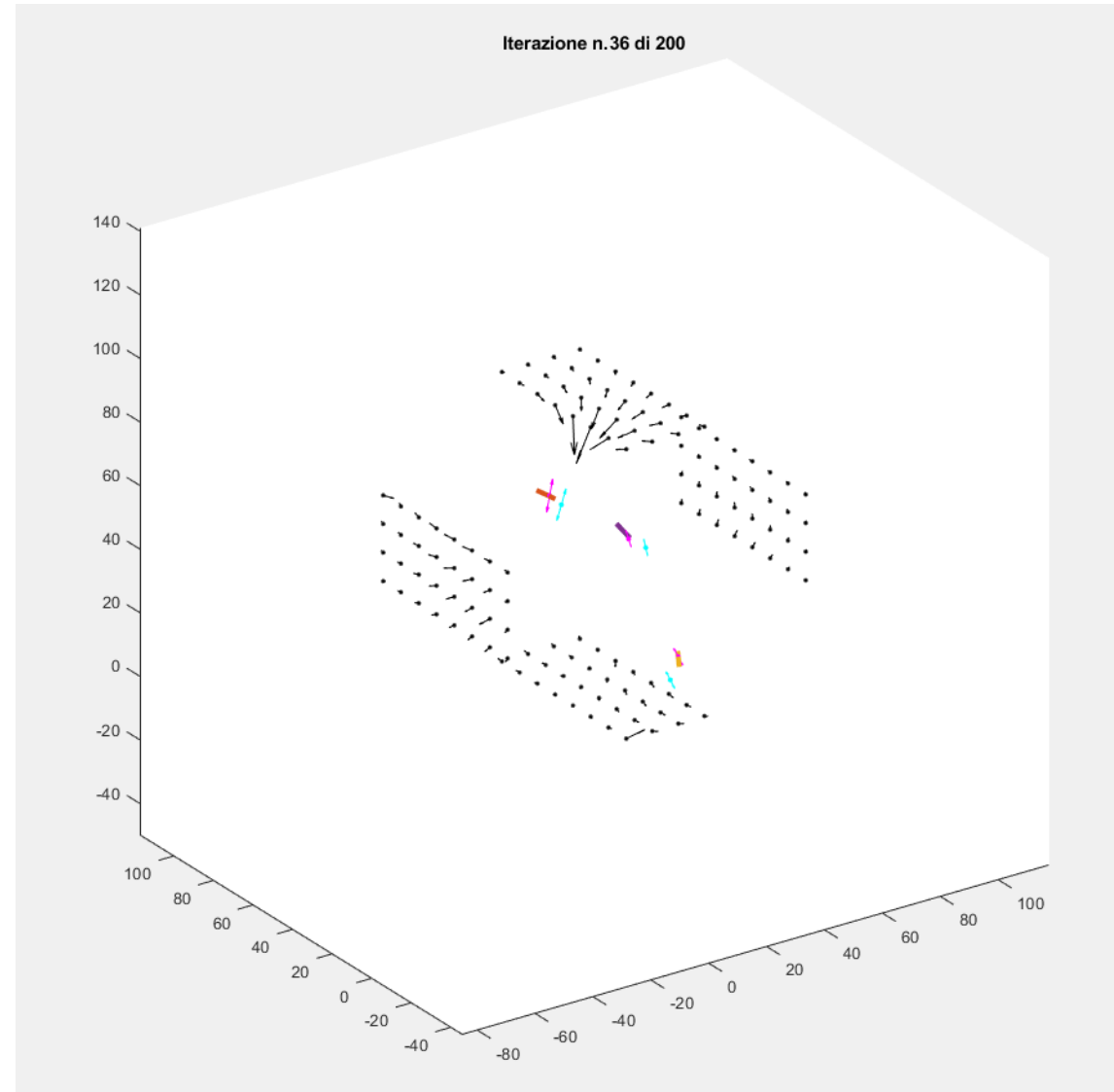
Fase di Run

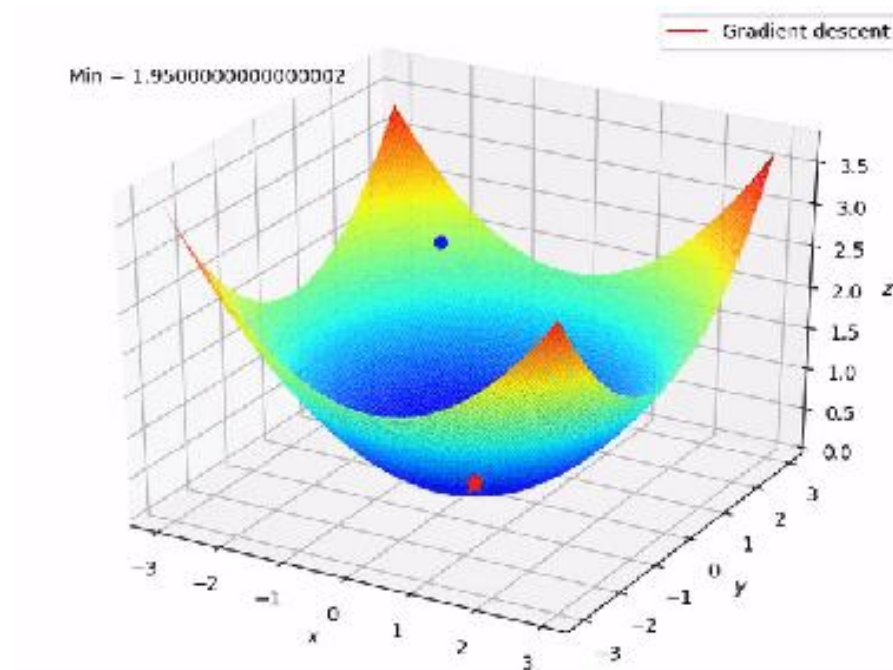
Questa parte del programma simula l'effettivo funzionamento della protesi:

1. Riceve in input i valori del campo magnetico rilevato dai sensori
2. Ottiene la posizione dei magneti
3. Controlla se tutti si trovano all'interno della traiettoria
4. Se la risposta è NO, cerca la rototraslazione che riporta ogni magnete nella sua traiettoria
5. Eliminazione rumore delle misurazioni

4. Algoritmo di compensazione

- Cerca la rototraslazione che riporta ogni magnete all'interno della propria traiettoria
- Per farlo, utilizza un algoritmo di minimizzazione constrained





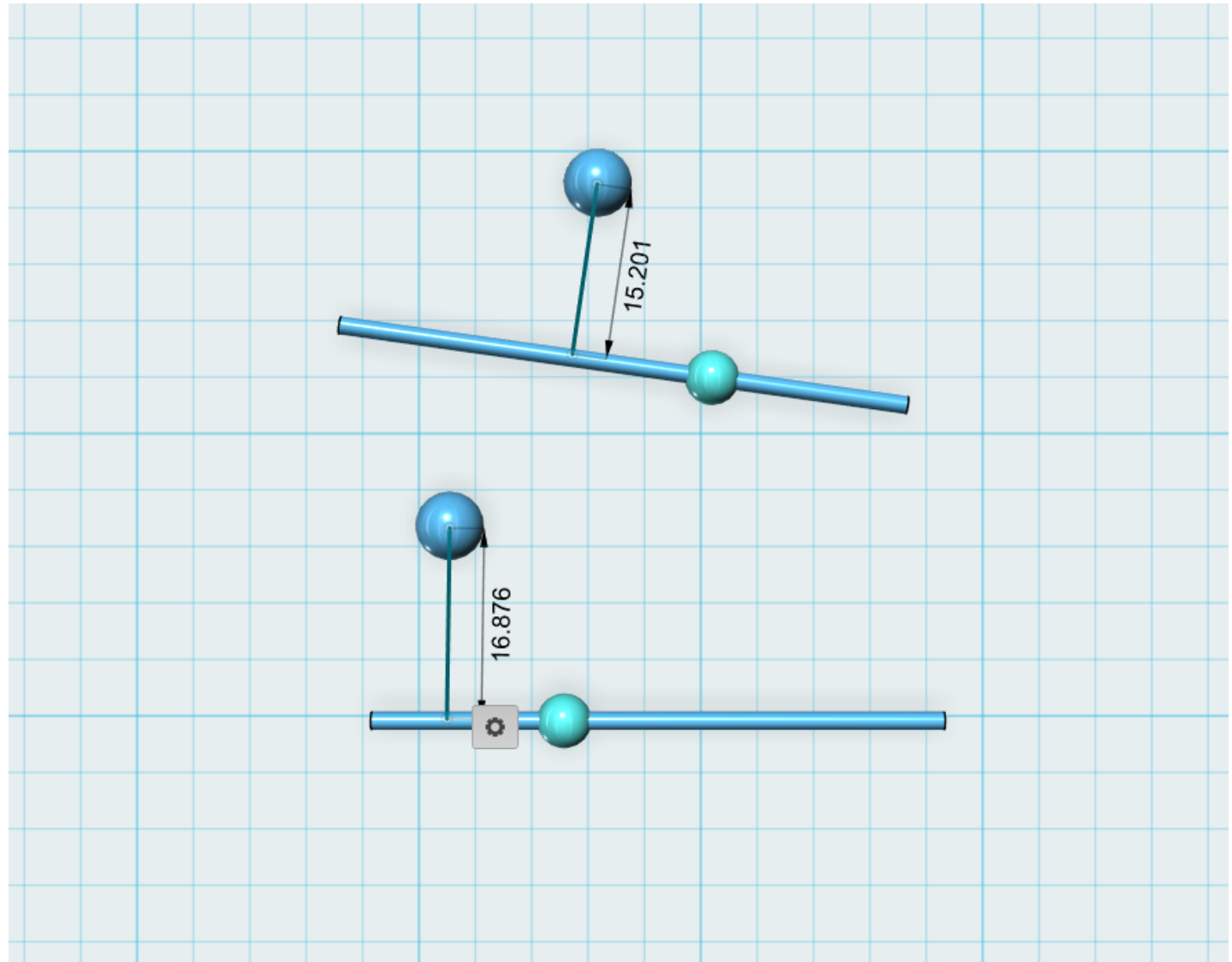
Algoritmo di
minimizzazione

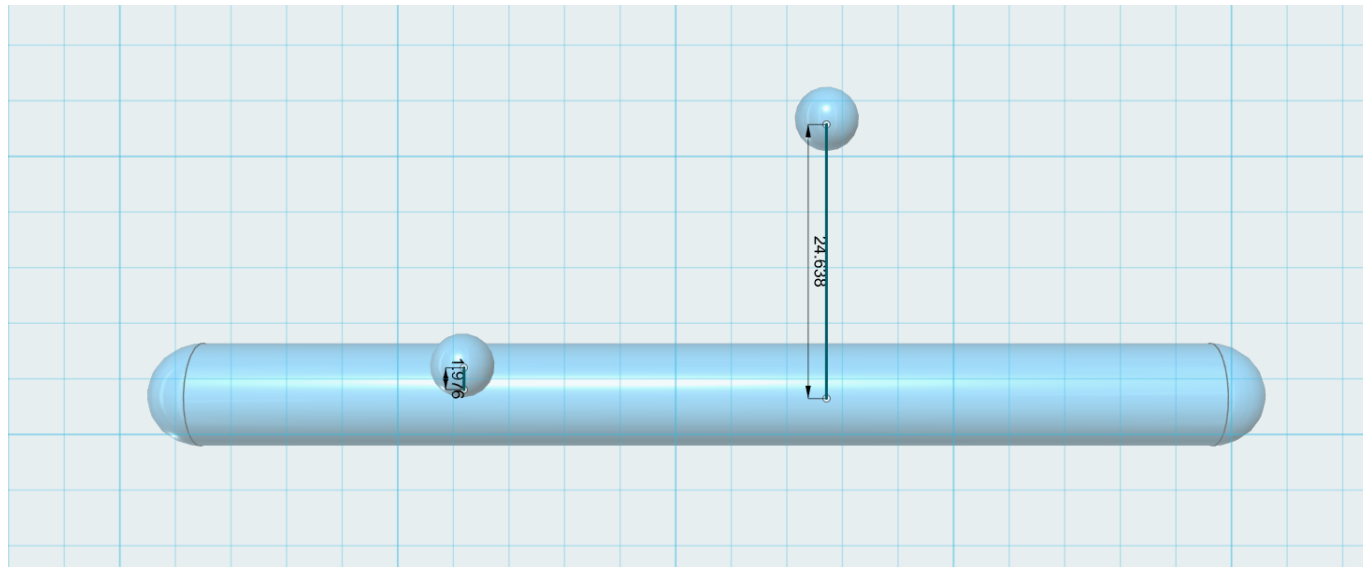
Algoritmo che, data una funzione, cerca di minimizzarla

Algoritmo di minimizzazione

- Algoritmo che, data una funzione, cerca di minimizzarla
- La funzione da minimizzare si basa sull'unico dato che abbiamo a disposizione: la DISTANZA DALLA TRAIETTORIA

** Non si può utilizzare la distanza dalla posizione reale, essendo questa l'incognita del problema! **

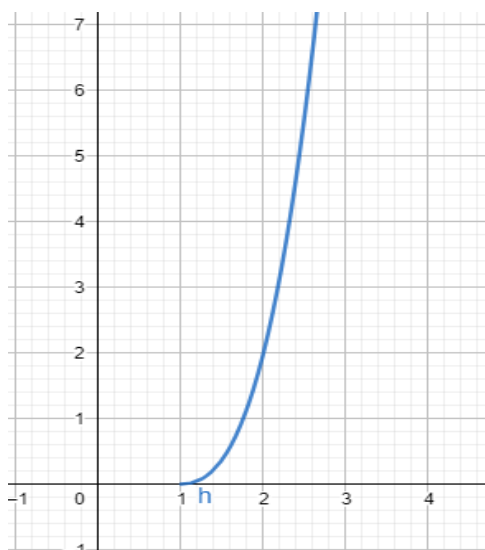
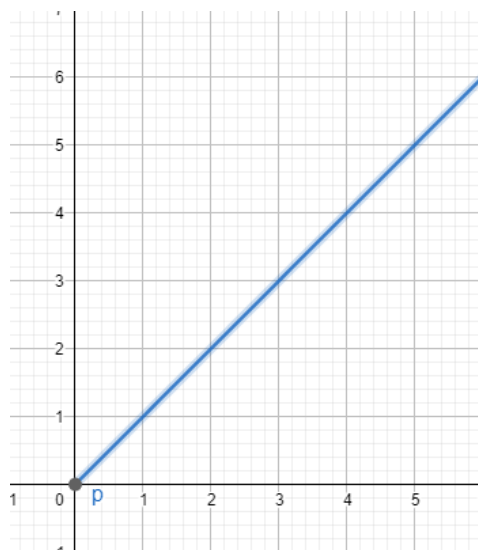
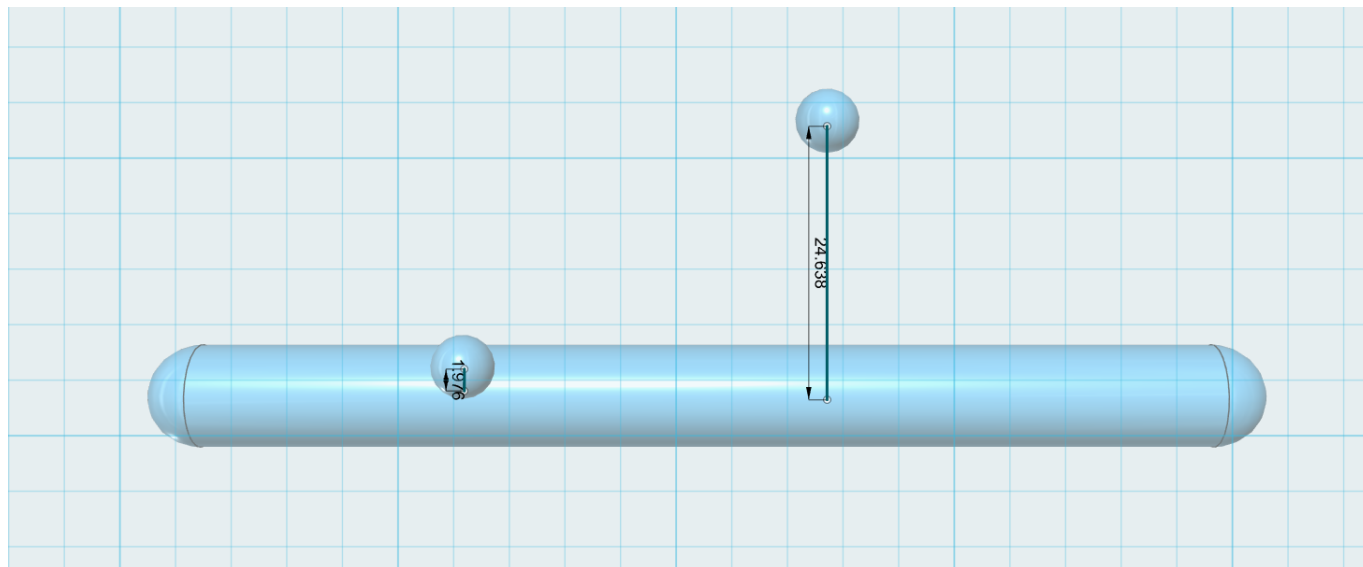




Funzione costo

- La traiettoria è un cilindro
- Si vuole imporre innanzitutto che il magnete rientri nella traiettoria
- Dopodiché, se il magnete è già nella traiettoria, cerchiamo di farlo avvicinare all'asse. Questo è però già meno necessario

(non dobbiamo penalizzare errori derivanti dalla misurazione)



Funzione costo

Per questo distinguiamo due casi:

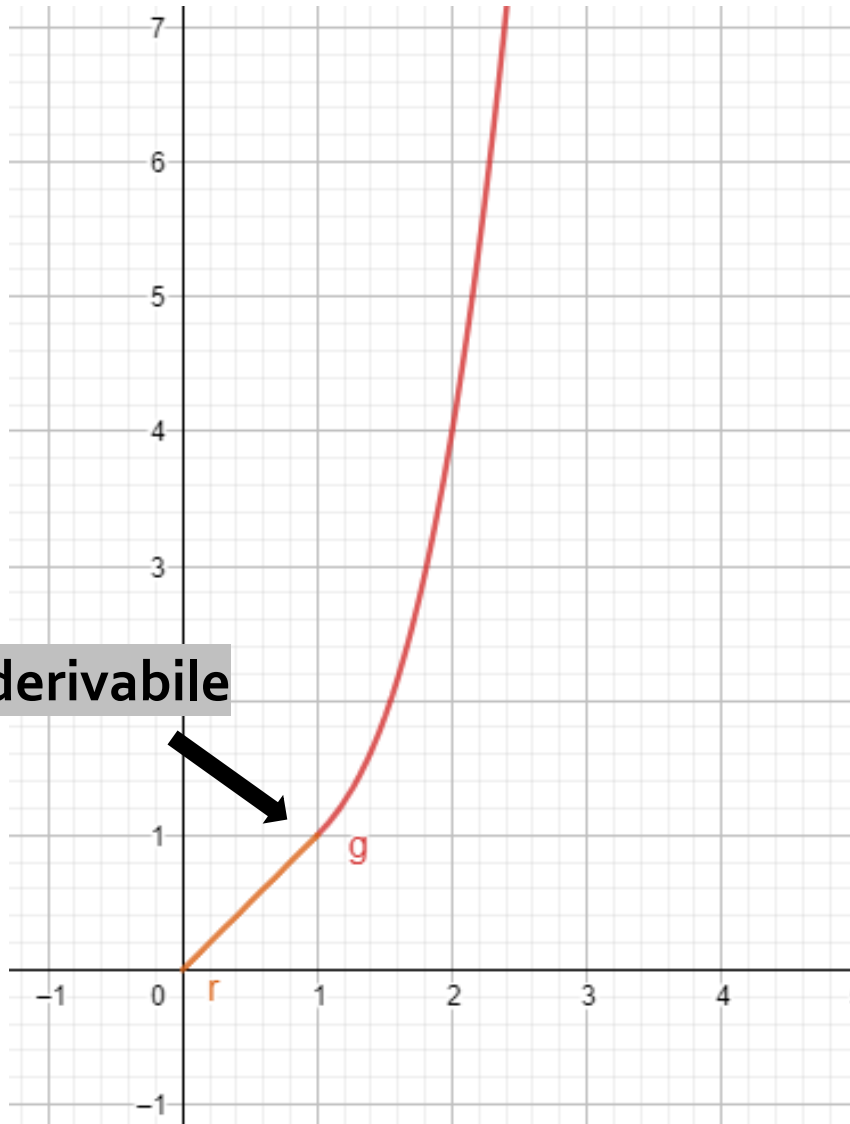
1. Distanza \leq errMax:

La distanza contribuisce linearmente alla funzione costo

2. Distanza $>$ errMax:

La distanza contribuisce quadraticamente alla funzione costo

Nota: è derivabile



Funzione costo

Per questo distinguiamo due casi:

1. Distanza $\leq \text{errMax}$:

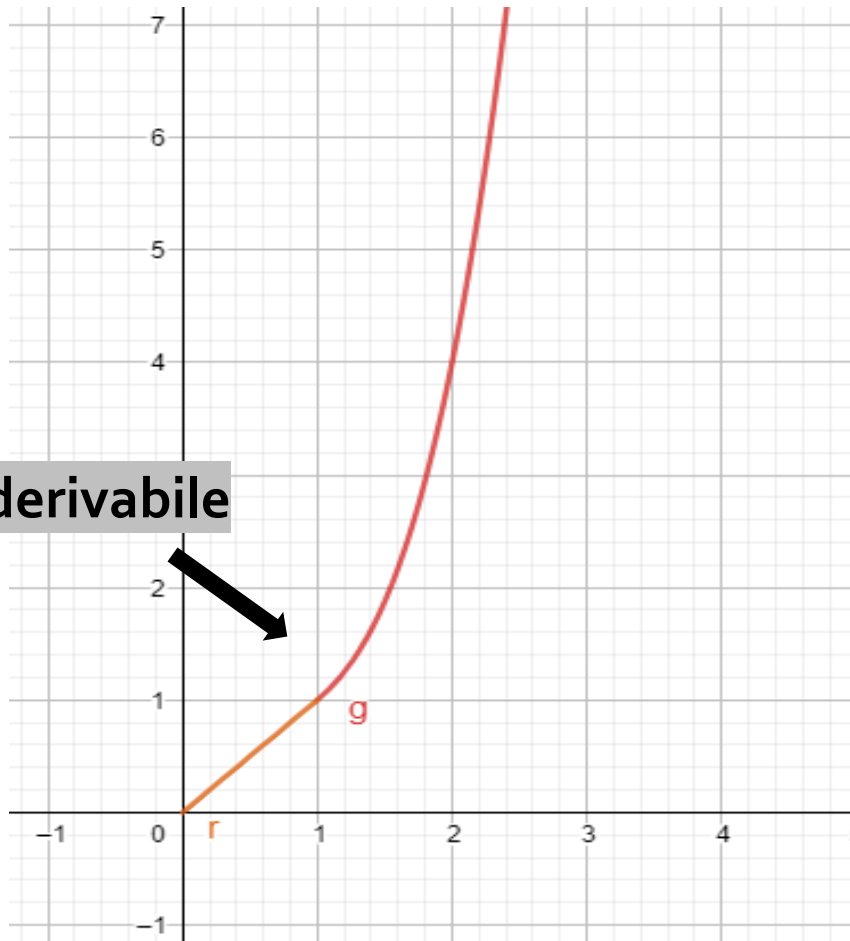
La distanza contribuisce linearmente alla funzione costo

2. Distanza $> \text{errMax}$:

La distanza contribuisce quadraticamente alla funzione costo

$$Cost = \sum_1^{nMag} \begin{cases} di & \text{if } (di \leq errMax) \\ di + (di - errMax)^2 & \text{if } (di \geq errMax) \end{cases}$$

Nota: è derivabile



Funzione costo

Per questo distinguiamo due casi:

1. Distanza $\leq errMax$:

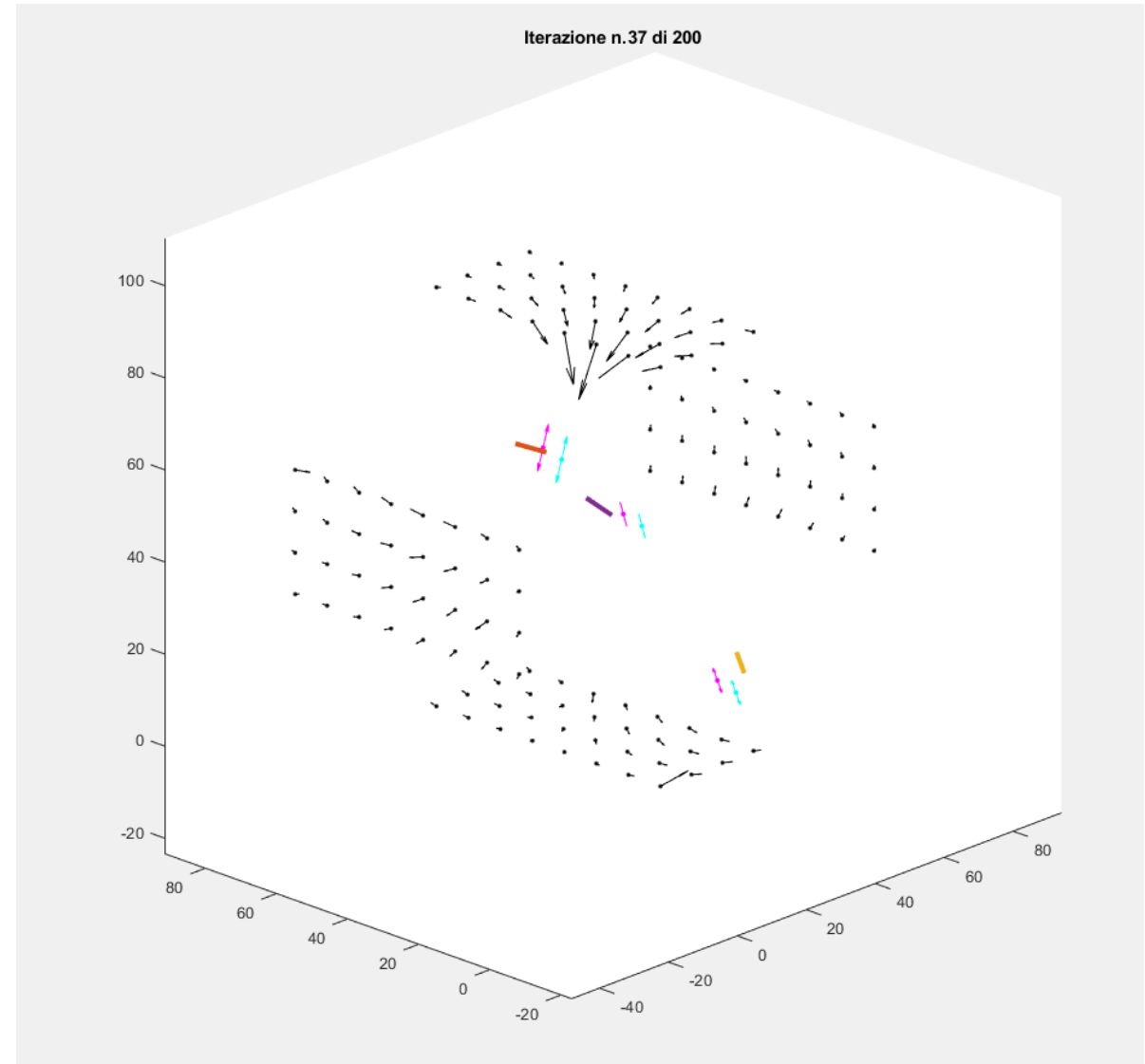
La distanza contribuisce linearmente alla funzione costo

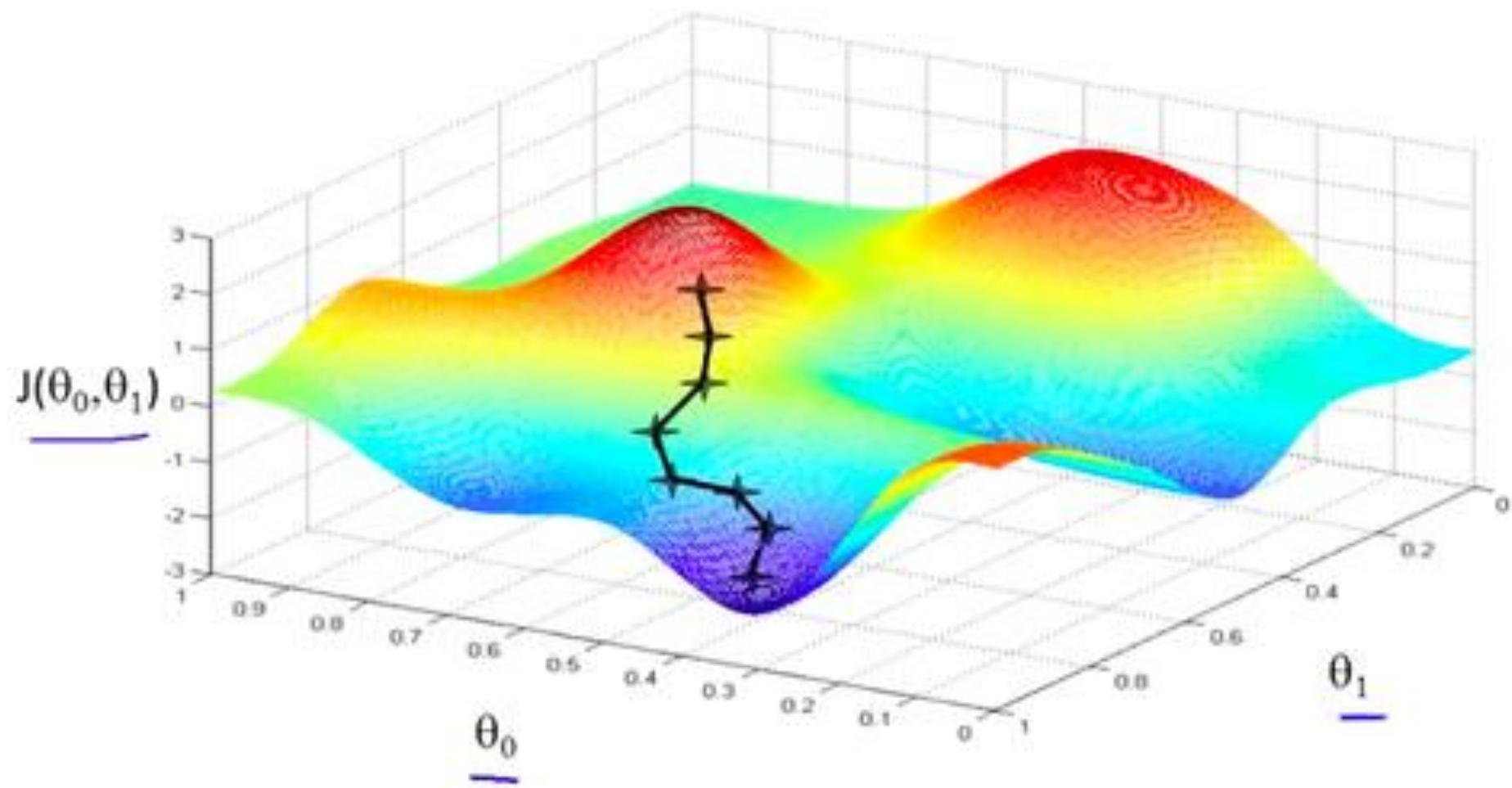
2. Distanza $> errMax$:

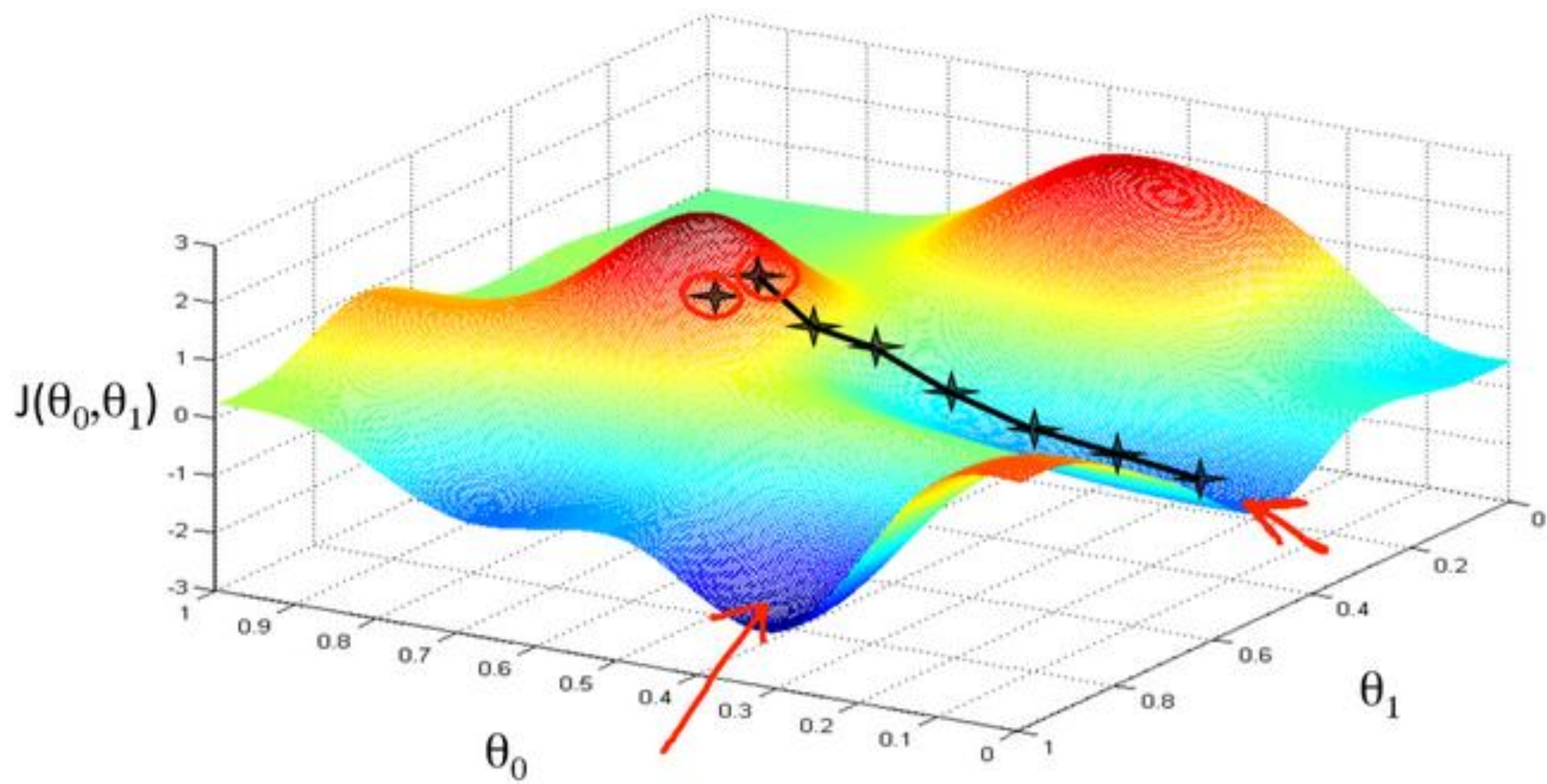
La distanza contribuisce quadraticamente alla funzione costo

Problema: minimi locali

Il problema principale degli algoritmi di minimizzazione è la presenza di minimi locali della funzione



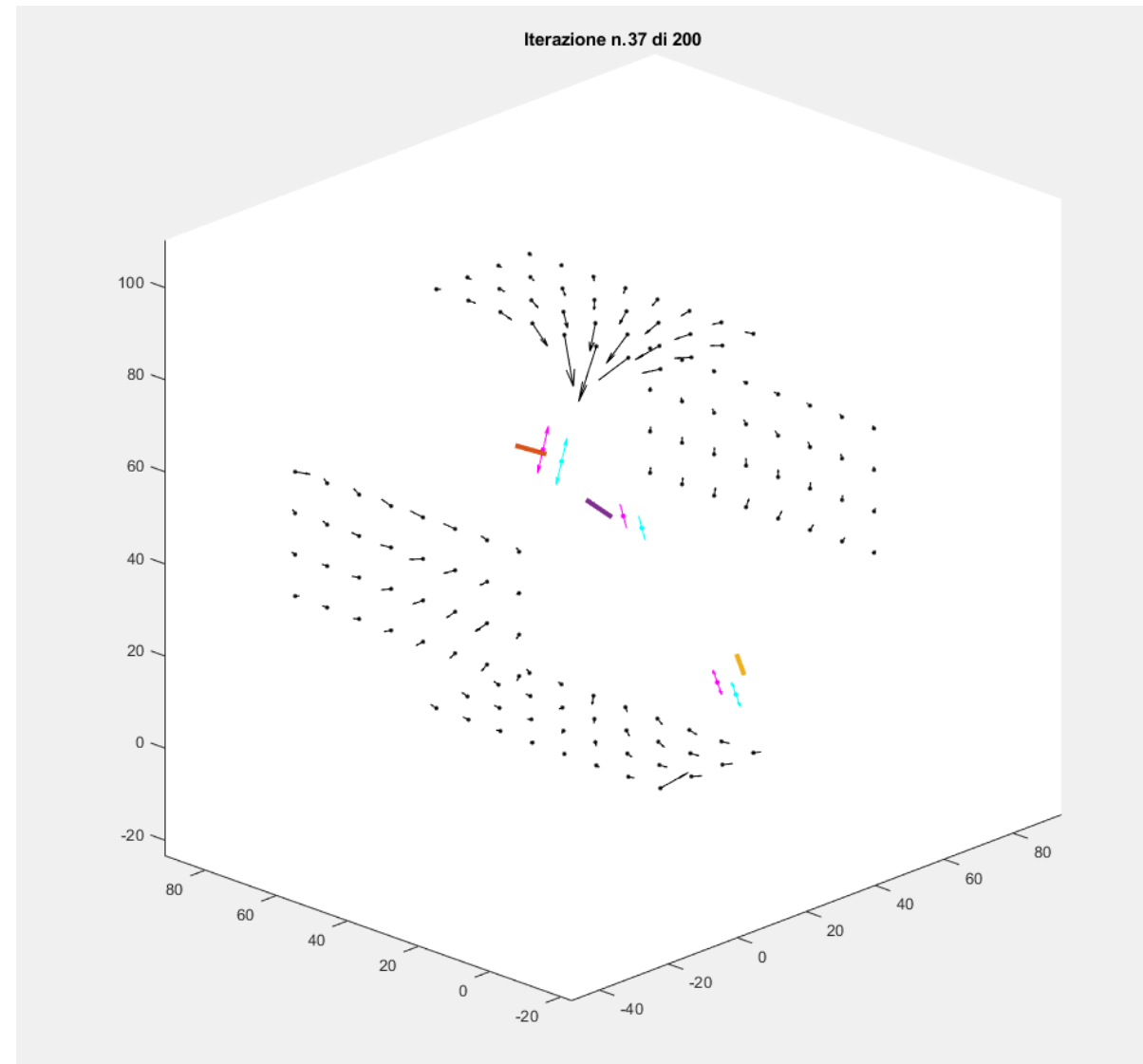




Risoluzione .1

CONTINUITA' DEI MOVIMENTI:

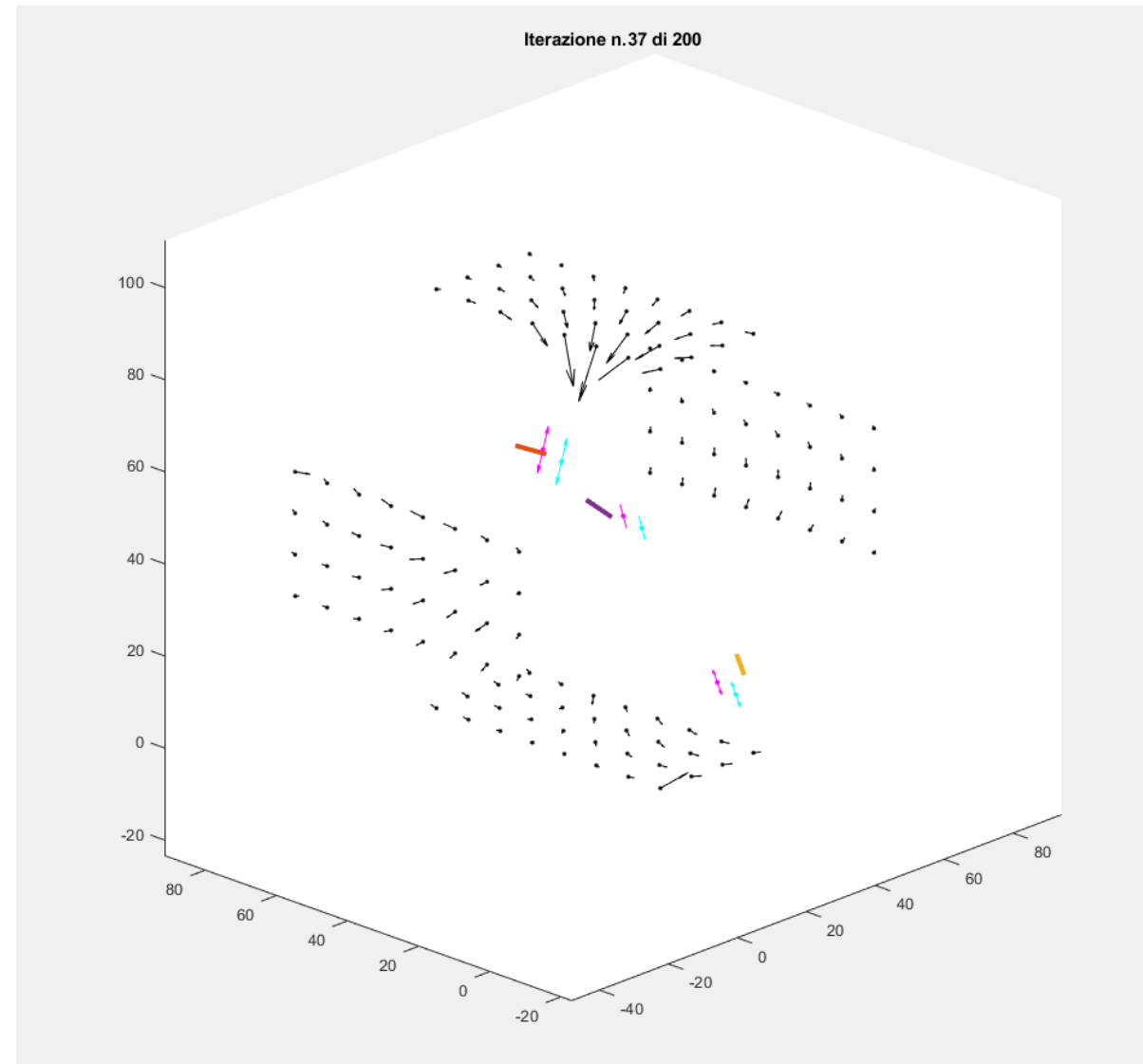
- Come posizione di partenza dell'algoritmo viene utilizzata la posizione precedente
- La frequenza di rilevazione è infatti di circa 20Hz
- La differenza tra una rilevazione e la successiva è dunque limitata



Risoluzione .2

ELIMINAZIONE RILEVAZIONI EVIDENTEMENTE ERRATE

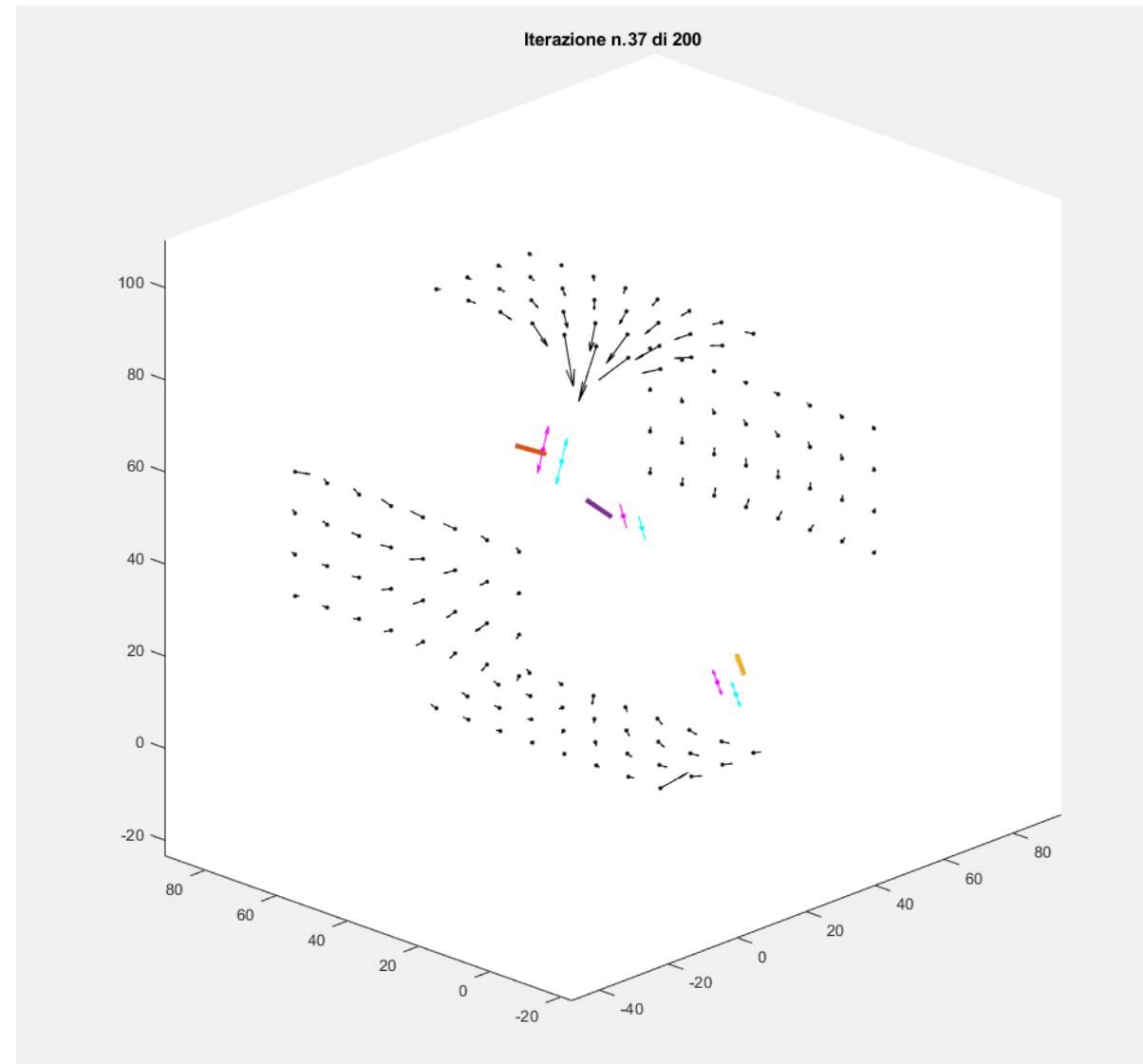
- Dato che ogni misurazione si basa sulla precedente, un errore in una misurazione rischia di propagarsi anche nelle misurazioni successive
- Vengono eliminate le misurazioni che:
 1. Si discostano troppo dalla precedente
 2. Mostrano magneti esterni al mockup
 3. Costo eccessivamente alto



Risoluzione .3

ALGORITMO "FMINCON" = FUNCTION MINIMUM CONSTRAINED

- Questo algoritmo ignora tutti i minimi che non rispettano un determinato vincolo
- Vincolo: "magneti all'interno delle traiettorie"

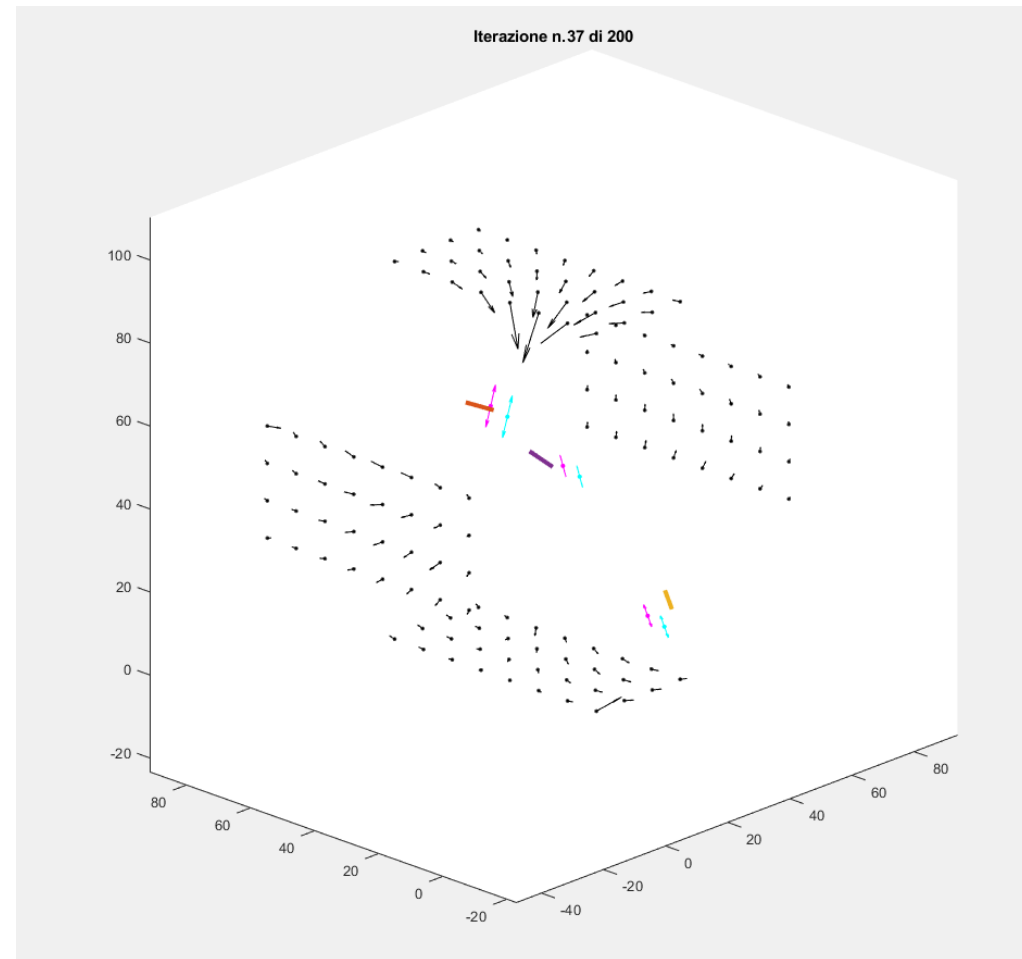


Risoluzione .3

ALGORITMO "FMINCON" = FUNCTION MINIMUM CONSTRAINED

- Questo algoritmo ignora tutti i minimi che non rispettano un determinato vincolo
- Vincolo: "magneti all'interno delle traiettorie"

$$\text{Constrain} : \{ d_i < errMax, \quad \forall i \in [1, nMag] \}$$



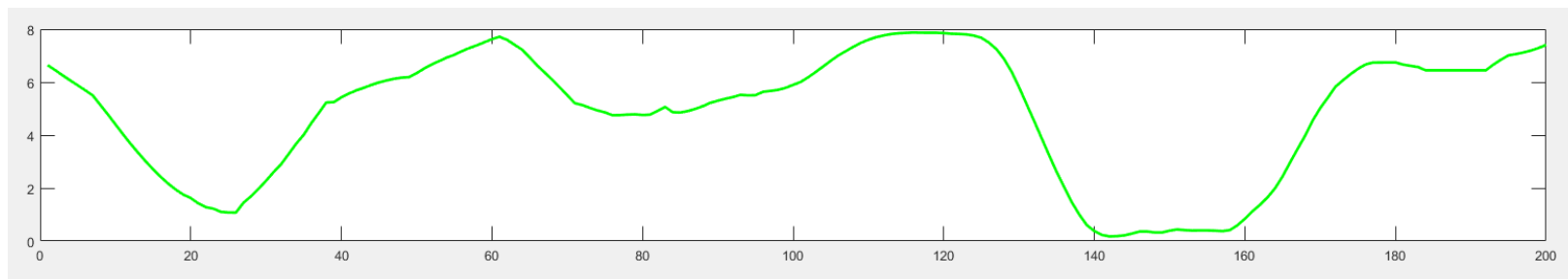
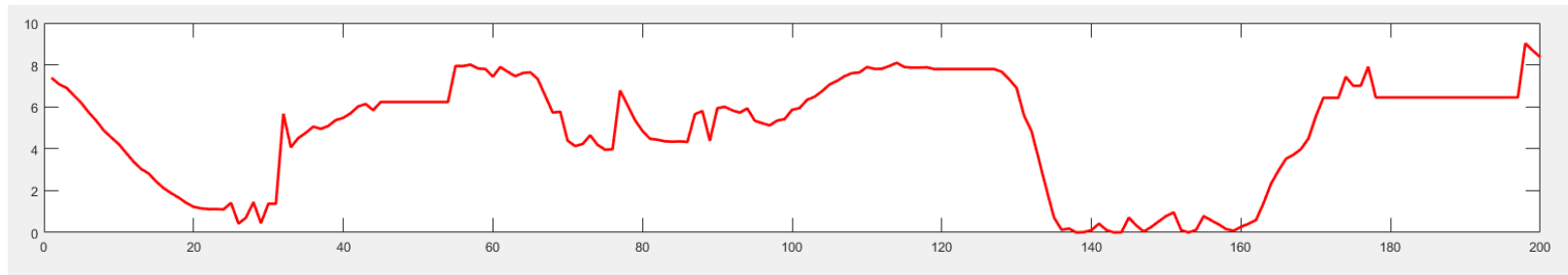
Fase di Run

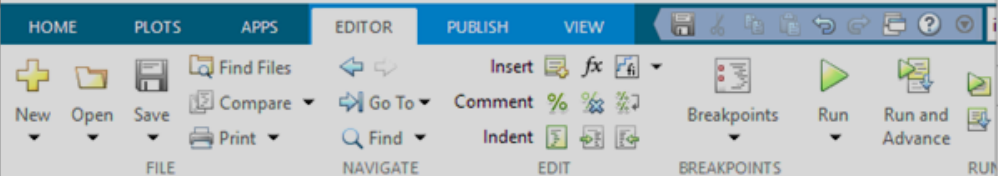
Questa parte del programma simula l'effettivo funzionamento della protesi:

1. Riceve in input i valori del campo magnetico rilevato dai sensori
2. Ottiene la posizione dei magneti
3. Controlla se tutti si trovano all'interno della traiettoria
4. Se la risposta è NO, cerca la rototraslazione che riporta ogni magnete nella sua traiettoria
5. Eliminazione rumore delle misurazioni

5. Eliminazione rumori di misurazione

- Media mobile sui valori ottenuti
- Questo permette di eliminare i rumori delle misurazioni





D:\wolff\Documents\università\Laboratorio feb 2021\Copy_of_real_data_analysis_2

Editor - D:\wolff\Documents\università\Laboratorio feb 2021\Copy_of_real_data_analysis_2\Disturbi Mecc...

variables_DMsym.m SETUP.m RUN.m find_rototranslation.m constraint

```

64 - D = 0.004; % Diameter of the magnets
65 - H = 0.002; % Height of the magnets
66 - M = 1.2706/(4*pi*1e-7); % Magnetization of the magnets [A/m]
67 - m = M*((D/2)^2*pi*H);
68
69 %% load data
70 % I valori di 'Data1_2' sono stati presi con i motori in movimen
71 % mockup fermo. Numero magneti = 3.
72 % Ai dati A già stato sottratto l'offset.

```

Command Window

	4	87	1.389108e-03	0.000e+00	4.035e-02	1.47
calc...	5	105	1.383313e-03	0.000e+00	1.977e-02	2.81
calc...	6	112	1.379036e-03	0.000e+00	1.000e+00	5.96
con...	7	119	1.376792e-03	0.000e+00	1.000e+00	1.24
cost...	8	126	1.375142e-03	0.000e+00	1.000e+00	1.88
dip...	9	136	1.374155e-03	0.000e+00	3.430e-01	1.42
dist...	10	147	1.374067e-03	0.000e+00	6.782e-03	1.649e-04
dist...	11	157	1.374065e-03	0.000e+00	6.782e-03	5.346e-05
erro...	12	247	1.374065e-03	0.000e+00	2.050e-11	3.470e-12

Optimization stopped because the relative changes in all elements of x are less than `options.StepTolerance` = 1.000000e-12, and the relative maximum constraint violation, 0.000000e+00, is less than `options.ConstraintTolerance` = 1.000000e-06.

Columns 1 through 5

"Rototranslation" = " -0.14193" "-1.8486" "0.011312" "0.92557"

Columns 6 through 7

"-0.12597" "-4.0915"

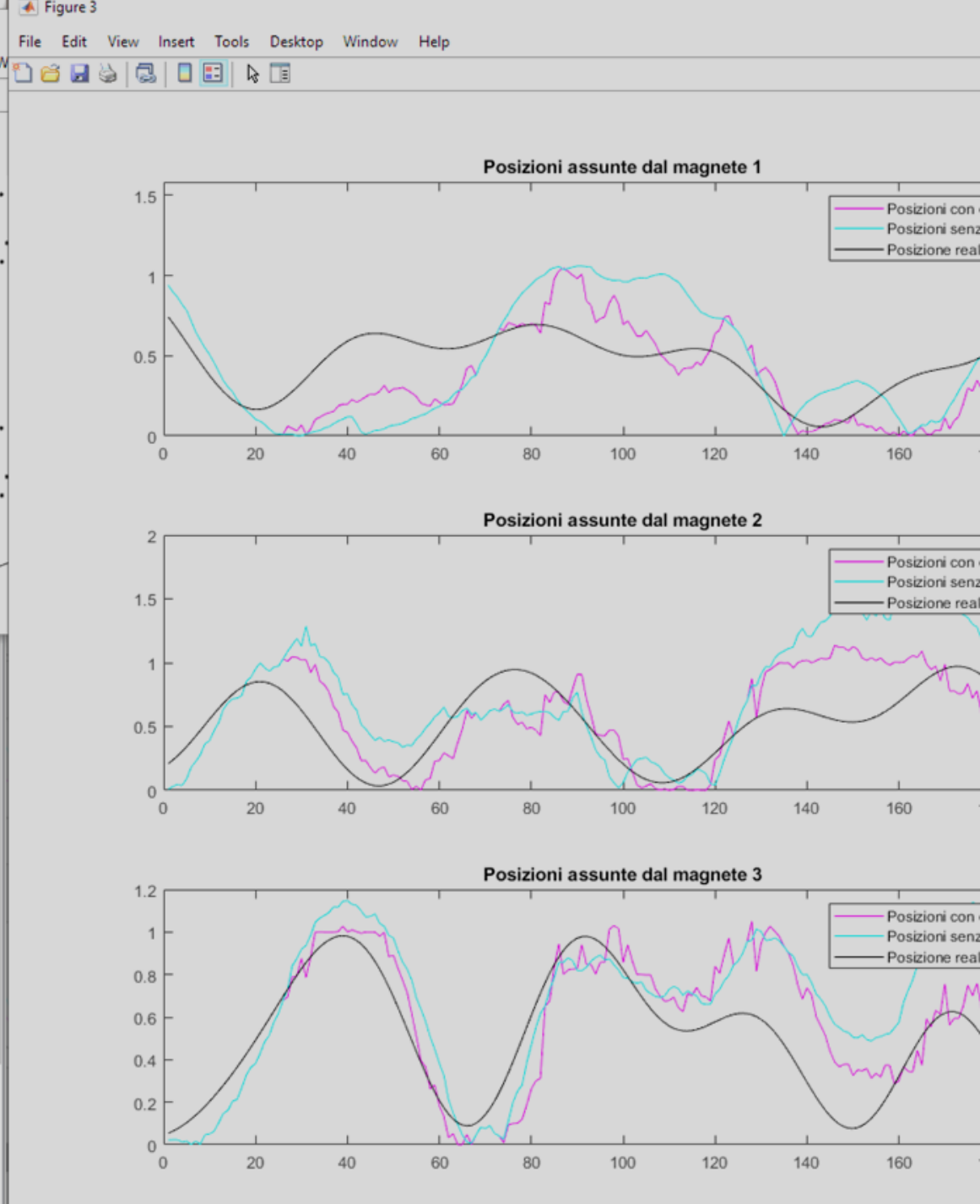
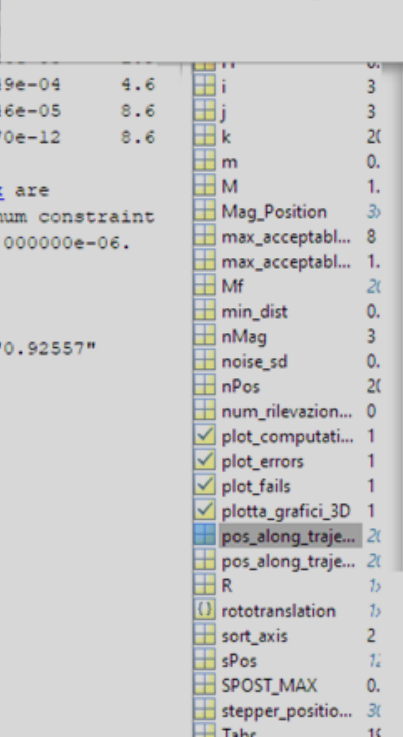
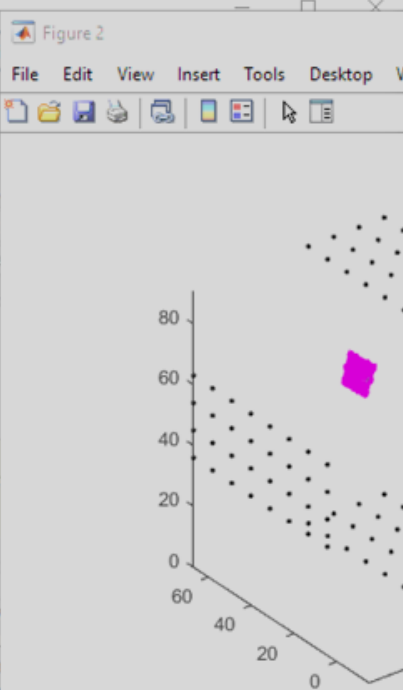
Costo = 6.838868e-03

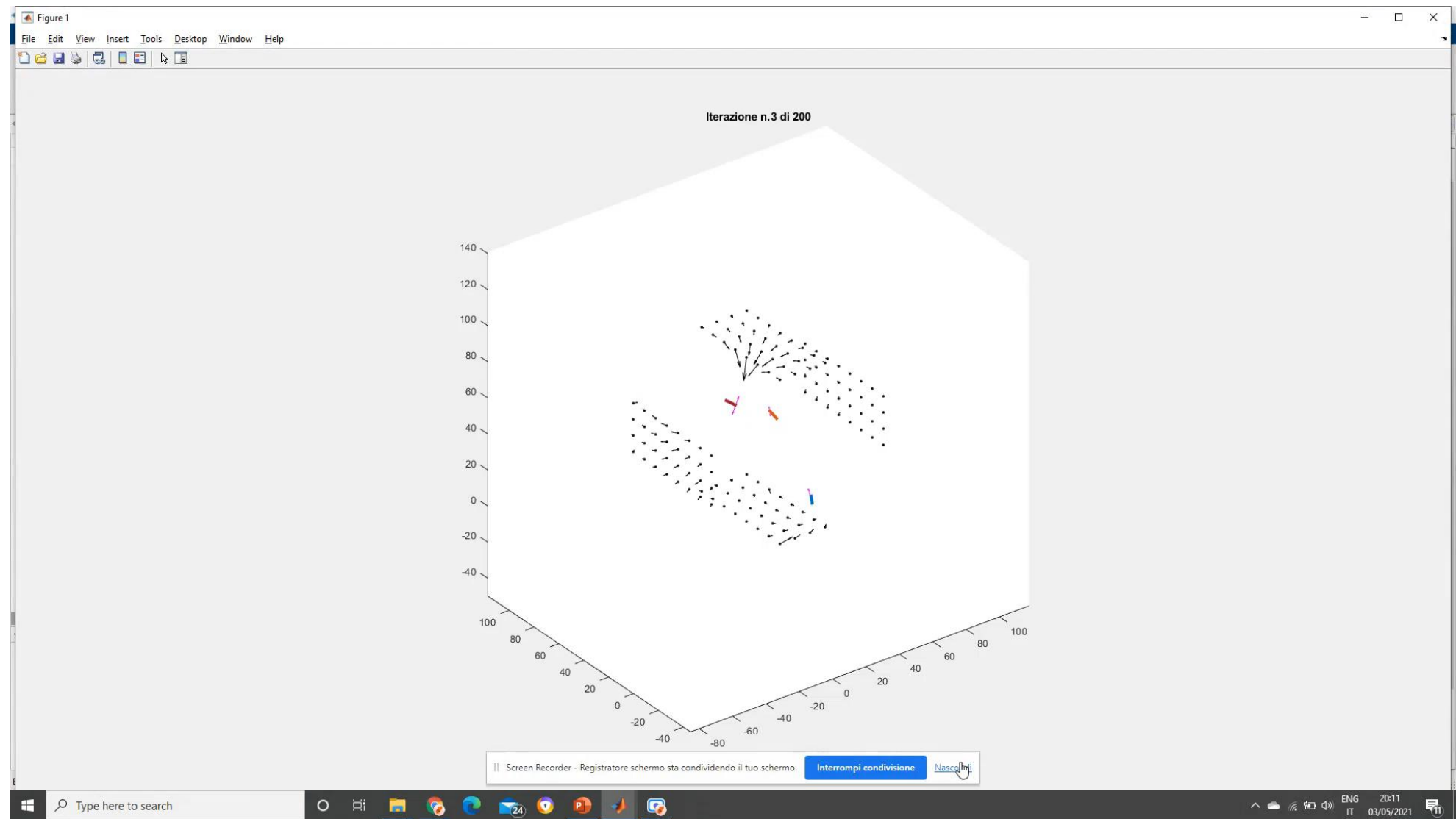
distanze_iniziali = [4.554797e-03, 4.869926e-03, 1.171716e-03]

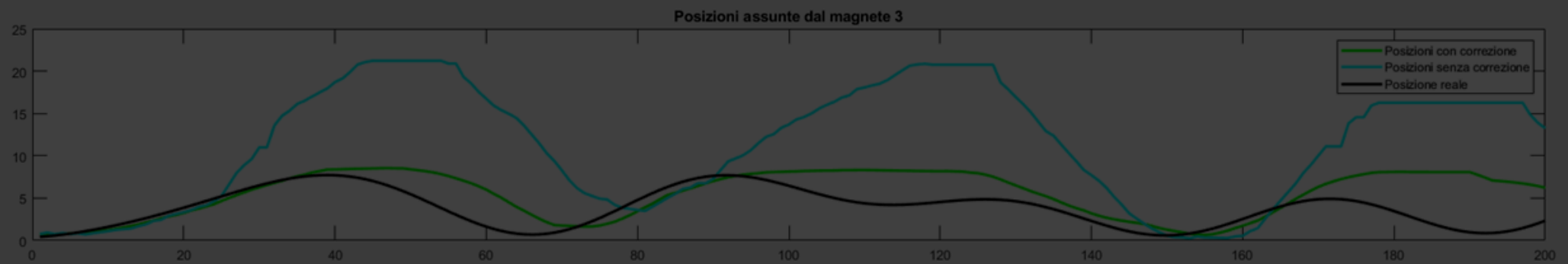
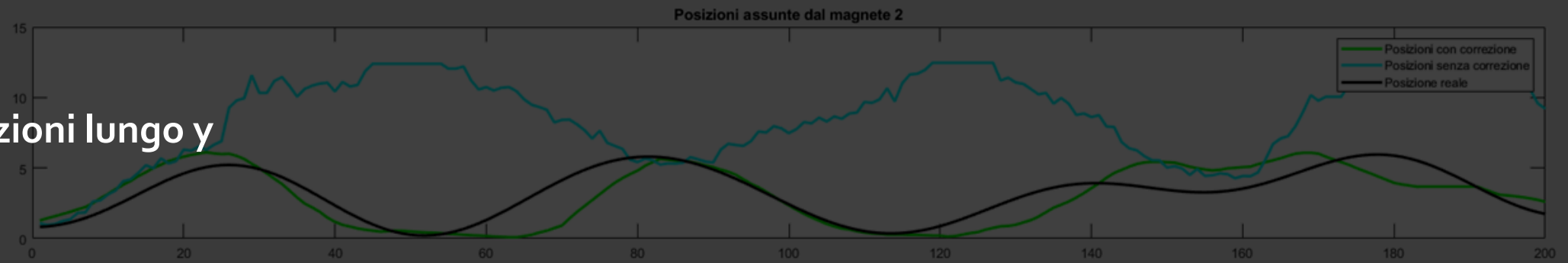
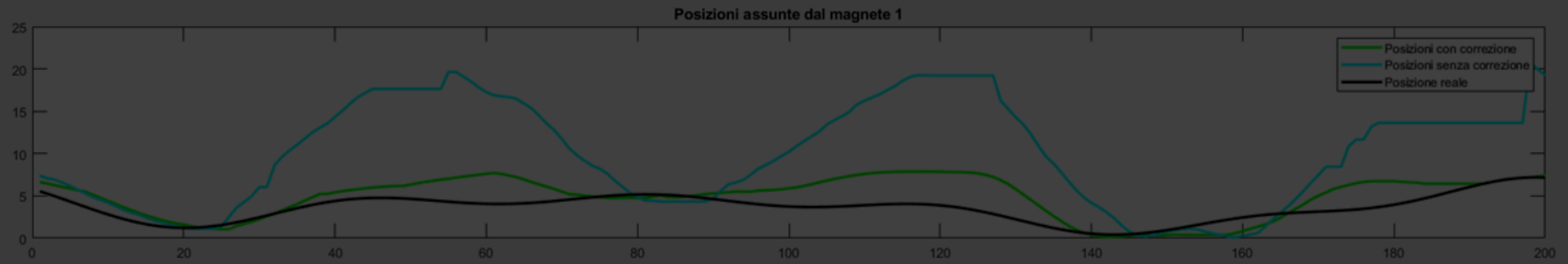
distanze_finali = [8.709604e-04, 5.028930e-04, 2.119595e-07]

RISULTO! ADESSO "everyone_in_trajectory" = 1!!!

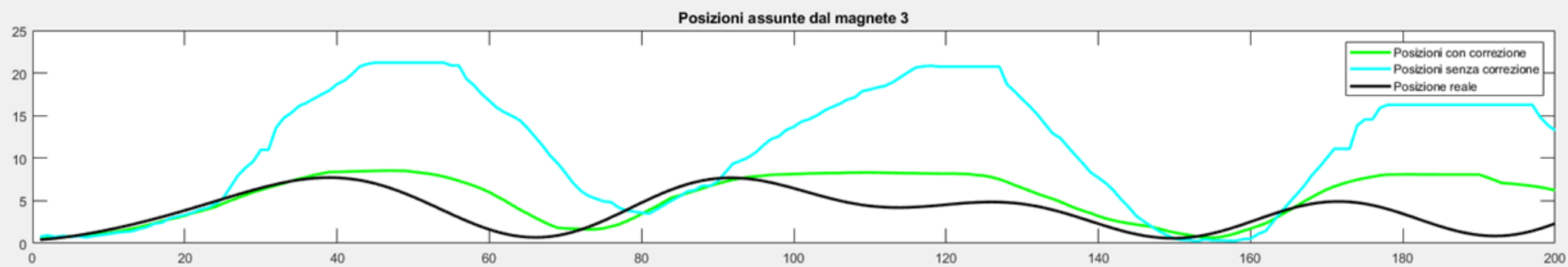
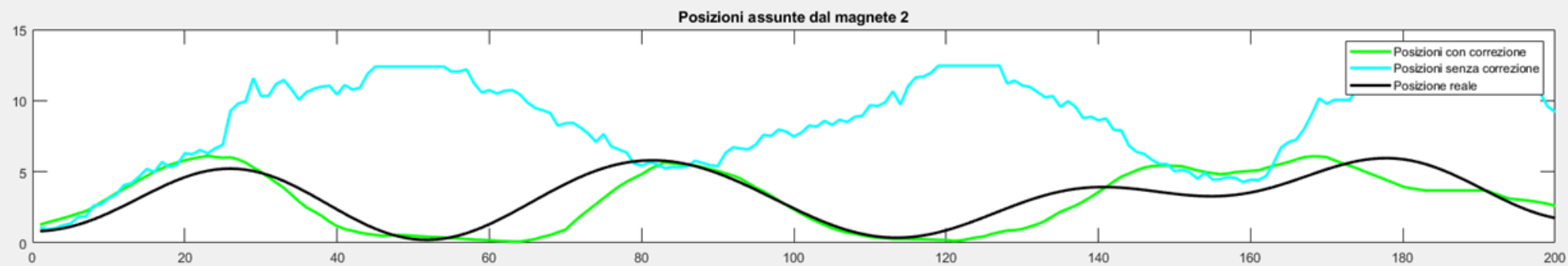
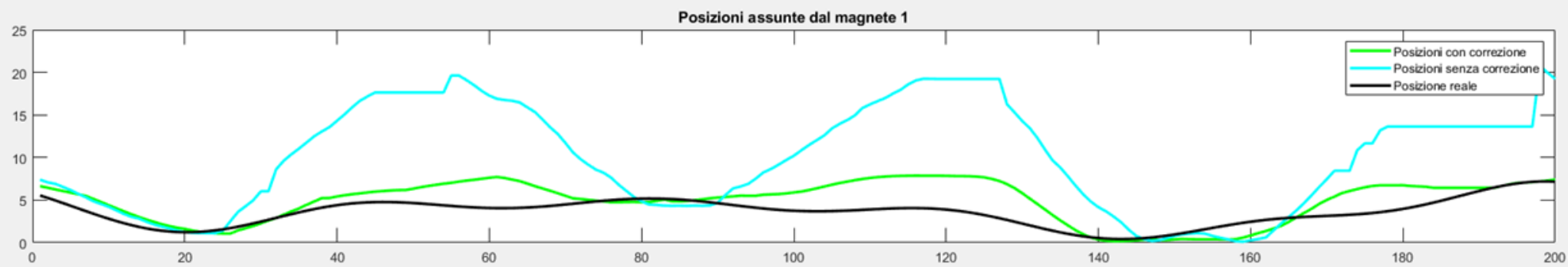
num_rilevazioni_radiare =



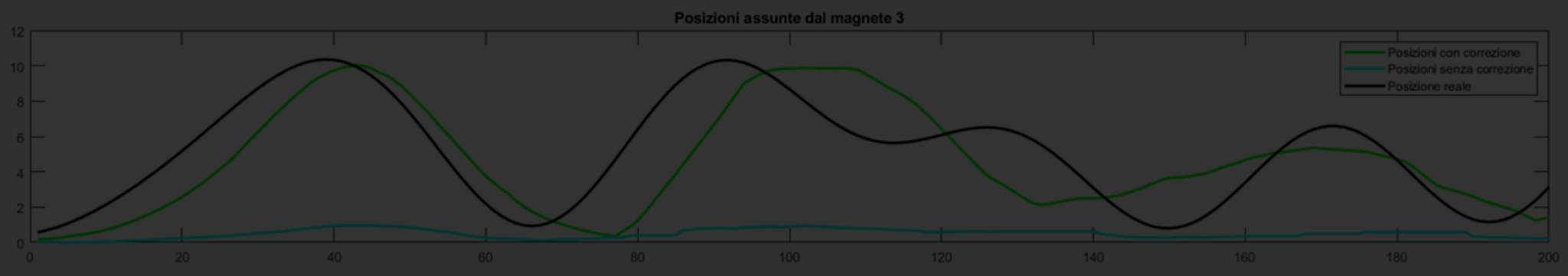
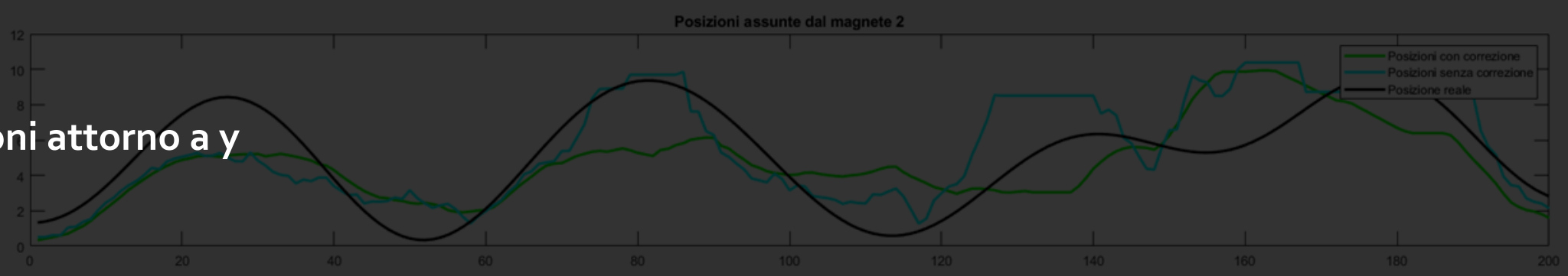
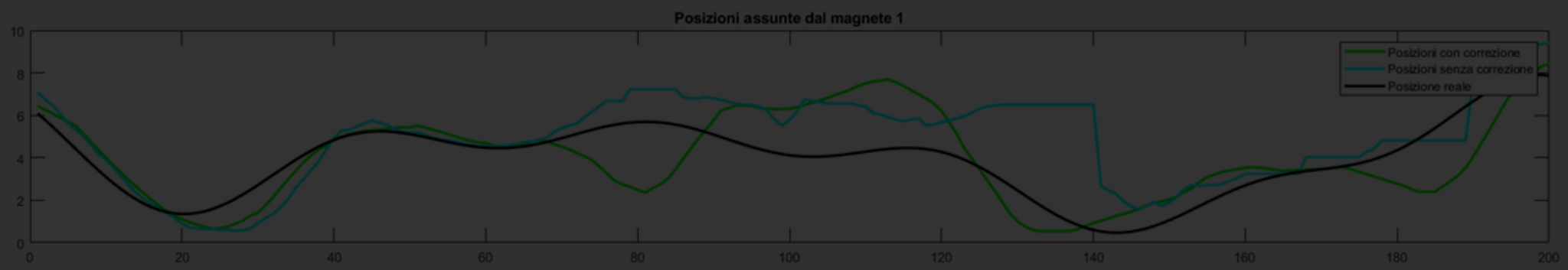


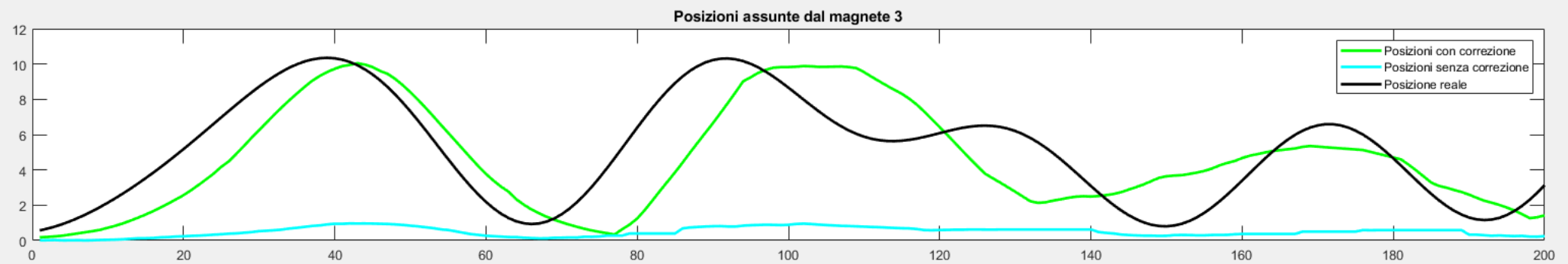
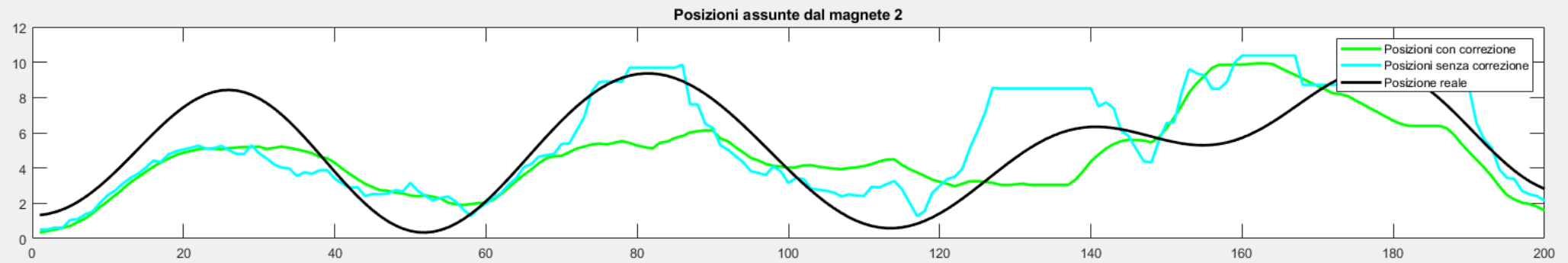
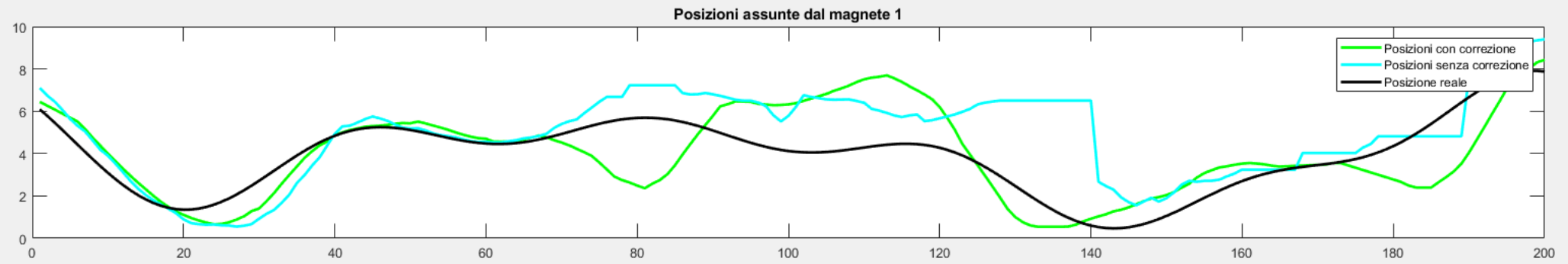


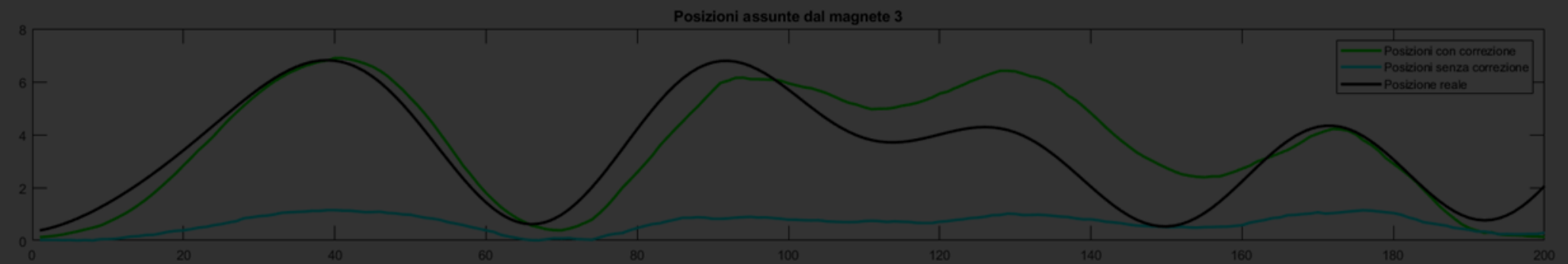
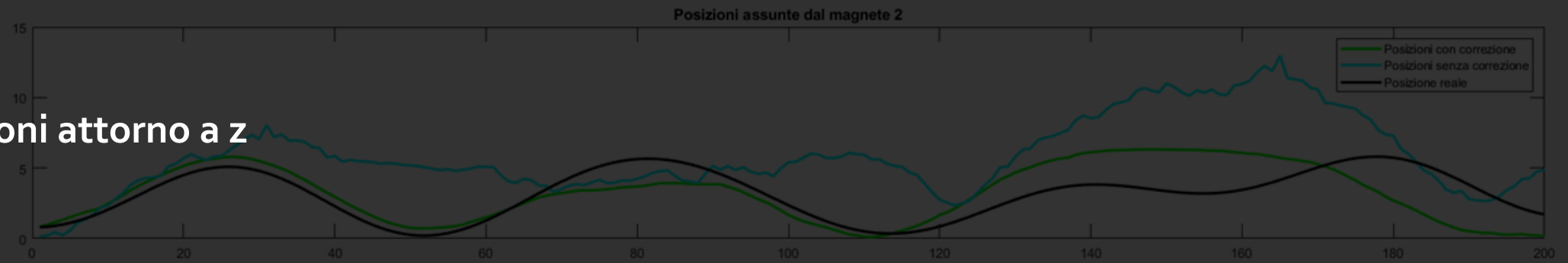
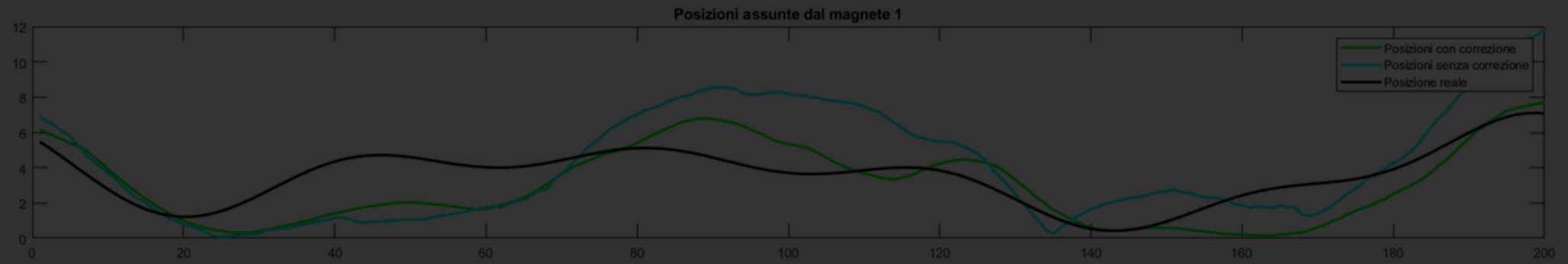
Traslazioni lungo y



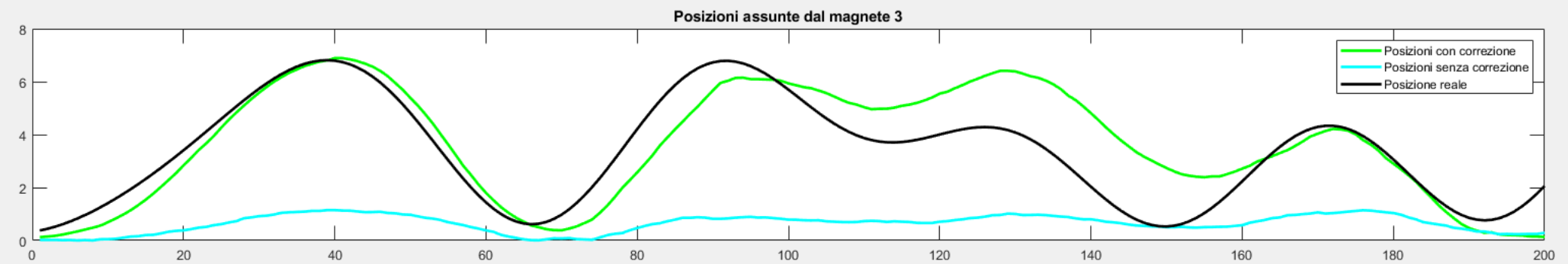
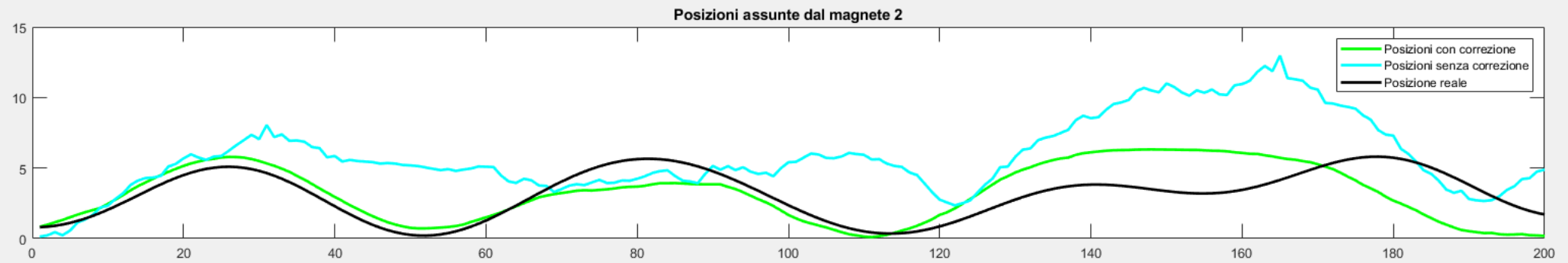
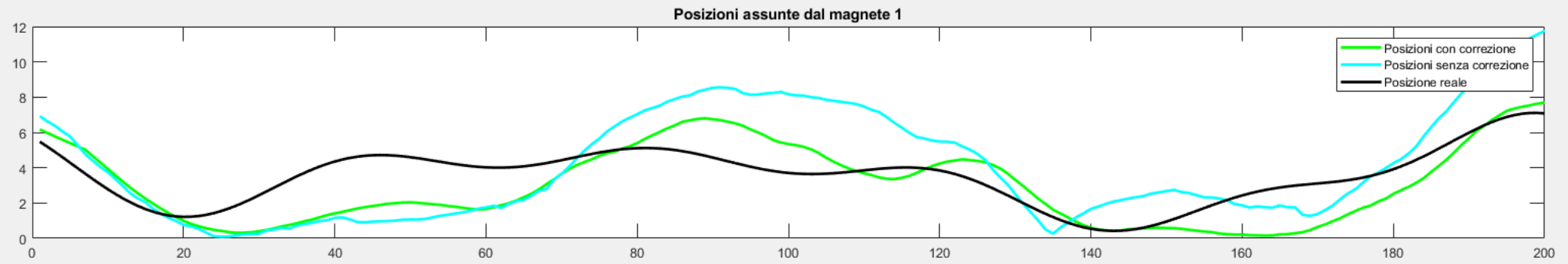
Rotazioni attorno a y

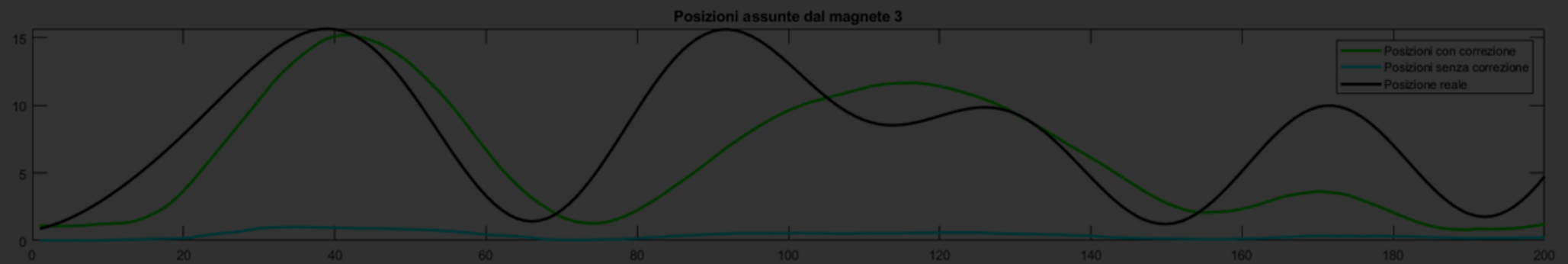
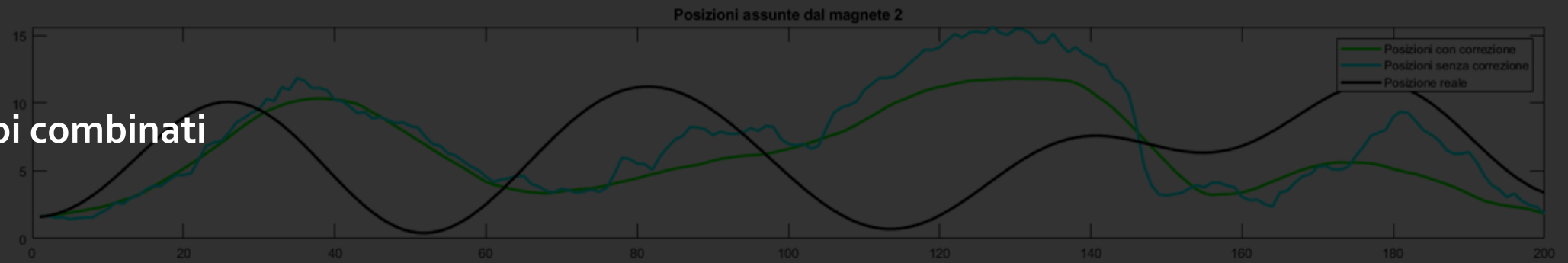
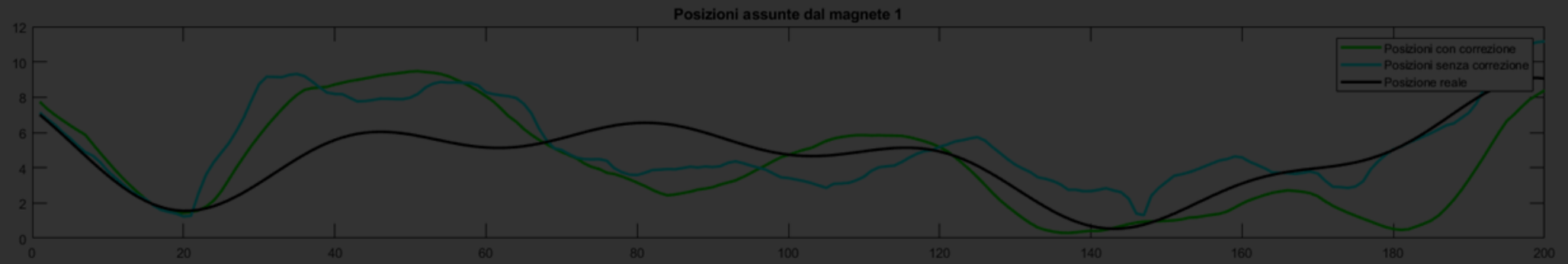




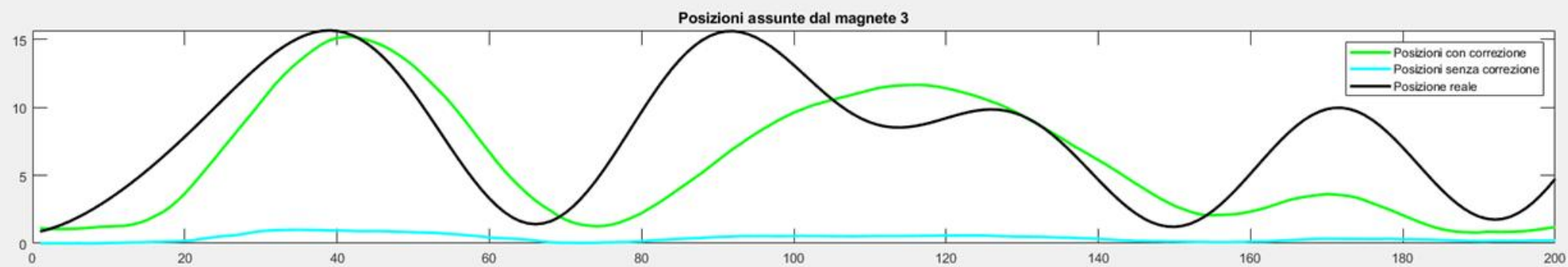
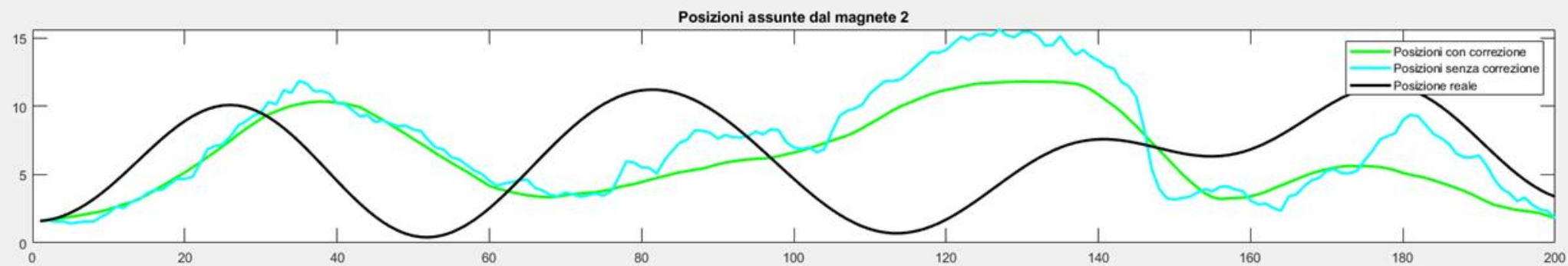
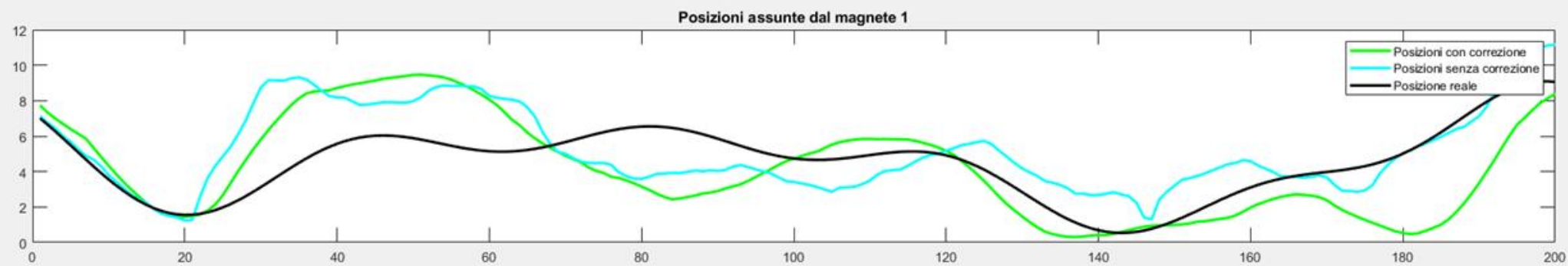


Rotazioni attorno a z





Disturbi combinati



Grazie per
l'attenzione