

POLITECNICO DI MILANO

FACOLTÀ DI INGEGNERIA INDUSTRIALE

Corso di Laurea in Ingegneria Aerospaziale



SVILUPPO DI UN SISTEMA DI NAVIGAZIONE E CONTROLLO PER MANOVRE DI PROSSIMITÀ BASATO SU SENSORE VISIVO

Candidato:

Diego Boroni

Matricola 724814

Relatore:

Prof. Michèle LAVAGNA

Anno Accademico 2010/2011

Ai miei genitori

Il presente lavoro di tesi riguarda lo sviluppo di un sistema integrato di navigazione e controllo per operazioni di prossimità autonome fra due satelliti in orbita intorno alla terra.

Posizione e assetto relativo sono ricostruiti usando i dati provenienti da un sensore visivo; a tal proposito sono stati implementati quattro diversi algoritmi, due iterativi e due non-iterativi, ed il filtro di Kalman non-lineare per la stima della posizione e orientazione relativa dei due satelliti.

Il controllo ottimo del moto e dell'assetto relativo viene realizzato mediante l'utilizzo della tecnica di controllo non-lineare $\theta - D$. Questo metodo è basato sulla formulazione pseudo-lineare non-unica (conosciuta come parametrizzazione SDC o linearizzazione estesa) della dinamica che governa il problema, e minimizzando un indice delle prestazioni; quindi, la legge di controllo sub-ottima è ottenuta approssimando la soluzione della ARE con una serie di potenze troncata.

Infine, il sistema integrato viene implementato nella simulazione numerica della fase finale di una manovra di docking per verificare l'accuratezza del sistema di controllo non-lineare e studiare l'uso del sensore visivo per la navigazione.

Parole chiave: stima della posizione e assetto, manovre di prossimità, filtro di Kalman esteso (EKF), controllo ottimo non-lineare, sistema integrato

ABSTRACT

This thesis focuses on development of integrated system consisting of navigation and control for autonomous proximity operations between two spacecraft orbiting around Earth.

Relative position and attitude are estimated using information from a vision sensor; four solution, two non-iterative and two iterative, and the extended Kalman filter are proposed to solve the pose estimation problem.

The optimal control of translational and rotational motions is achieved using the non-linear approach, referred to T- D technique. This method involves manipulating governing dynamic equations into a pseudo-linear non-unique form (referred to SDC parameterization or extended linearization), in which system matrices are given as a function of the current state, and minimizing a quadratic-like performance index; then, a sub-optimal control law is obtained by approximation of the solution of ARE by a truncated series expansion..

Finally, a docking final-phase are simulated in order to verify the accuracy of the proposed non-linear controller and investigate the use of a vision sensor for navigation purpose.

Key word: pose estimation, proximity operations, extended Kalman filter (EKF), non-linear optimal control, integrated system

RINGRAZIAMENTI

1	Introduzione	1
1.1	Stato dell'arte	1
1.2	Scopo del lavoro	2
1.3	Struttura della tesi	2
2	Stima della posizione e orientazione di un oggetto nello spazio	3
2.1	Sistemi di riferimento, notazione e trasformazioni omogenee	4
2.1.1	Sistemi di riferimento	4
2.1.2	Notazione e trasformazioni	5
2.2	La fotocamera	6
2.2.1	Il modello prospettico o <i>pinhole</i>	7
2.2.2	Il modello prospettico debole o <i>weak-perspective model</i>	8
2.2.3	Il sensore <i>CCD</i> (<i>Charged Coupled Design</i>)	8
2.2.4	Il rumore	8
2.2.5	Parametri estrinseci	9
2.2.6	Parametri intrinseci	9
2.3	Soluzione non iterativa: approccio diretto al problema	12
2.3.1	Impostazione del problema	12
2.3.2	Soluzione approssimata	12
2.3.3	Soluzione con correzione	14
2.3.4	Simulazione numerica	15
2.3.5	Conclusioni sull'algoritmo	16
2.4	Soluzione non iterativa: algoritmo di Lepetit (EP_nP)	24
2.4.1	Impostazione del problema	24
2.4.2	Soluzione del sistema	25
2.4.3	Ottimizzazione con Gauss-Newton	26
2.4.4	Simulazione numerica	26
2.4.5	Conclusioni sull'algoritmo	27

2.5	Soluzione iterativa: algoritmo di Haralick	36
2.5.1	3D-3D estimation	36
2.5.2	Algoritmo di Haralick	37
2.5.3	Simulazione numerica	38
2.5.4	Conclusioni sull'algoritmo	39
2.6	Soluzione iterativa: algoritmo a 2 stadi	50
2.6.1	Equazione di collinearità	50
2.6.2	Calcolo della distanza ottima	50
2.6.3	Flow-chart delle operazioni da eseguire	51
2.6.4	Algoritmo di Umeyama	52
2.6.5	Simulazione numerica	53
2.6.6	Conclusioni sull'algoritmo	54
2.7	Confronto fra l'algoritmo a 2 stadi e quello di Lepetit	65
3	Modello del satellite e dell'ambiente	67
3.1	Definizione dei sistemi di riferimento	67
3.1.1	Riferimento inerziale, I	67
3.1.2	Riferimento orbitale, O	68
3.1.3	Riferimento assi corpo, B	69
3.2	Meccanica orbitale	69
3.2.1	Equazione di Keplero	70
3.2.2	Determinazione dei parametri Kepleriani	70
3.2.3	Disturbi orbitali: resistenza atmosferica	72
3.2.4	Disturbi orbitali: gradiente di gravità	72
3.2.5	Equazioni di Hill	73
3.2.6	Determinazione dell'accelerazione angolare orbitale	74
3.3	Dinamica dell'assetto	75
3.3.1	Equazioni di Eulero	75
3.3.2	Parametrizzazione dell'assetto con i quaternioni	76
3.3.3	Disturbi sull'assetto: resistenza atmosferica	76
3.3.4	Disturbi sull'assetto: gradiente di gravità	77
4	Controllo ottimo non-lineare e filtro di Kalman esteso (EKF)	79
4.1	Evoluzione del controllo ottimo di sistemi non-lineari	79
4.1.1	Controllo ottimo non lineare: il metodo θ - D	80
4.1.2	Definizione del problema	82
4.1.3	Simulazione numerica	84
4.2	Filtro di Kalman non-lineare discreto	88
4.2.1	La matrice di sensitività	90
4.2.2	Modello dei giroscopi	91
4.2.3	Soluzione analitica dell'equazione dell'assetto relativo a tempo-discreto	92
4.2.4	EKF per l'assetto	92

4.2.5	EKF per la determinazione dell'orbita e dell'assetto	95
4.2.6	Simulazioni numeriche	97
5	Risultati	101
5.1	Definizione dello scenario di simulazione	101
5.2	Risultati numerici	101
6	Conclusioni e sviluppi futuri	103
	Appendici	103
A	Matrici pseudo-lineari per il controllo	105
B	Matrici linearizzate per il filtro di Kalman	107
	Bibliografia	109

ELENCO DELLE FIGURE

2.1	rappresentazione dei tre sistemi di riferimento: <i>scena, fotocamera</i> <i>e oggetto</i>	5
2.2	Modello prospettico	7
2.3	vista del piano immagine e dei relativi sistemi di riferimento . . .	10
2.4	rappresentazione dei sistemi di riferimento adottati e dei punti caratteristici (in giallo)	13
2.5	Algoritmo di Mukundan; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, a=0,5 m, b=0,6 m, f=50 mm, $\sigma_{err} = 0$ m	17
2.6	Algoritmo di Mukundan; valori reali $\phi = -30^\circ$, $\theta = 6^\circ$, $\psi = 35^\circ$, a=0,5 m, b=0,6 m, f=50 mm, $\sigma_{err} = 0$ m	18
2.7	Algoritmo di Mukundan; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, a=0,5 m, b=0,6 m, f=50 mm, $\sigma_{err} = 10^{-4}$ m	19
2.8	Algoritmo di Mukundan; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, a=0,5 m, b=0,6 m, f=50 mm, $\sigma_{err} = 10^{-5}$ m	20
2.9	Algoritmo di Mukundan; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, a=0,5 m, b=0,6 m, f=50 mm, $\sigma_{err} = 10^{-6}$ m	21
2.10	Algoritmo di Mukundan; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, a=0,5 m, b=0,6 m, f=100 mm, $\sigma_{err} = 10^{-4}$ m	22
2.11	Algoritmo di Mukundan; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, a=0,5 m, b=0,6 m, f=5 mm, $\sigma_{err} = 10^{-4}$ m	23
2.12	Algoritmo EPnP; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, f=50 mm, $\sigma_{err} = 0$ m, 4 punti caratteristici	28
2.13	Algoritmo EPnP; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, f=50 mm, $\sigma_{err} = 0$ m, 5 punti caratteristici	29
2.14	Algoritmo EPnP; valori reali $\phi = -30^\circ$, $\theta = 6^\circ$, $\psi = 35^\circ$, f=50 mm, $\sigma_{err} = 0$ m, 5 punti caratteristici	30
2.15	Algoritmo EPnP; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, f=50 mm, $\sigma_{err} = 10^{-4}$ m, 5 punti caratteristici	31

2.16	Algoritmo EPnP; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, f=50 mm, $\sigma_{err} = 10^{-5}$ m, 5 punti caratteristici	32
2.17	Algoritmo EPnP; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, f=50 mm, $\sigma_{err} = 10^{-6}$ m, 5 punti caratteristici	33
2.18	Algoritmo EPnP; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, f=100 mm, $\sigma_{err} = 10^{-4}$ m, 5 punti caratteristici	34
2.19	Algoritmo EPnP; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, f=5 mm, $\sigma_{err} = 10^{-4}$ m, 5 punti caratteristici	35
2.20	Massimo errore nel calcolo di R e \mathbf{t} con algoritmo 3D-3D pose estimation di Haralick	37
2.21	Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, f=50 mm, $\sigma_{err} = 0$ m	40
2.22	Algoritmo di Haralick; valori reali $\phi = -30^\circ$, $\theta = 6^\circ$, $\psi = 35^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, f=50 mm, $\sigma_{err} = 0$ m	41
2.23	Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [1 \ 2 \ 50]^T$ m, f=50 mm, $\sigma_{err} = 0$ m	42
2.24	Algoritmo di Haralick; valori reali $\phi = -30^\circ$, $\theta = 6^\circ$, $\psi = 35^\circ$, $\mathbf{t} = [1 \ 2 \ 50]^T$ m, f=50 mm, $\sigma_{err} = 0$ m	43
2.25	Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 4 \ 0, 5 \ 3]^T$ m, f=50 mm, $\sigma_{err} = 10^{-4}$ m	44
2.26	Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 4 \ 0, 5 \ 3]^T$ m, f=50 mm, $\sigma_{err} = 10^{-5}$ m	45
2.27	Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 4 \ 0, 5 \ 3]^T$ m, f=50 mm, $\sigma_{err} = 10^{-6}$ m	46
2.28	Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, f=100 mm, $\sigma_{err} = 10^{-4}$ m	47
2.29	Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, f=5 mm, $\sigma_{err} = 10^{-4}$ m	48
2.30	Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 4 \ 0, 5 \ 3]^T$ m, f=50 mm, $\sigma_{err} = 10^{-6}$ m, punti caratteristici nel piano z=0 m	49
2.31	Massimo errore nel calcolo di R e \mathbf{t} con l'algoritmo di Umeyama	53
2.32	Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, f=50 mm, $\sigma_{err}=0$ m	55
2.33	Algoritmo 2-stage; valori reali $\phi = -30^\circ$, $\theta = 6^\circ$, $\psi = 35^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, f=50 mm, $\sigma_{err}=0$ m	56
2.34	Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [1 \ 2 \ 50]^T$ m, f=50 mm, $\sigma_{err}=0$ m	57
2.35	Algoritmo 2-stage; valori reali $\phi = 30^\circ$, $\theta = -6^\circ$, $\psi = 35^\circ$, $\mathbf{t} = [1 \ 2 \ 50]^T$ m, f=50 mm, $\sigma_{err}=0$ m	58
2.36	Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, f=50 mm, $\sigma_{err} = 10^{-4}$ m	59

2.37	Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, $f=50$ mm, $\sigma_{err} = 10^{-5}$ m	60
2.38	Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, $f=50$ mm, $\sigma_{err} = 10^{-6}$ m	61
2.39	Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, $f=100$ mm, $\sigma_{err} = 10^{-4}$ m	62
2.40	Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, $f=5$ mm, $\sigma_{err} = 10^{-4}$ m	63
2.41	Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, $f=50$ mm, $\sigma_{err} = 10^{-6}$ m, punti caratteristici nel piano $z=0$ m	64
3.1	sistemi di riferimento per la meccanica orbitale	68
3.2	Visualizzazione dei 6 parametri Kepleriani	70
4.1	Accelerazioni nel riferimento orbitale e coppie di controllo	85
4.2	Posizione nel riferimento corpo target e velocità angolare relativa	86
4.3	Quaternioni relativi	87
4.4	schema del filtro di Kalman	88
4.5	Errore nella stima della distanza e velocità relativa con il filtro di Kalman	98
4.6	Errore nella stima dei termini di drift dei giroscopi con il filtro di Kalman	99
4.7	Errore nella stima dei quaternioni relativi con il filtro di Kalman	100

ELENCO DELLE TABELLE

2.1	Confronto algoritmo <i>2-stadi</i> e <i>EPnP</i> per la determinazione della posa; t_z variabile fra 2 e 50 m, $t_x = 0,5$ e $t_y = 0,4$ m	65
2.2	Soluzioni ottenute con l'algoritmo <i>EPnP</i> in funzione del numero di iterazioni del metodo di <i>Gauss - Newton</i>	66

1.1 Stato dell'arte

Manovre di prossimità quali il *docking* e il *rendezvous* eseguite autonomamente dai satelliti costituiscono una tecnologia importante per le missioni spaziali correnti e soprattutto per quelle future; alcuni esempi odierni sono *ATV* (*Automated transfer vehicle*) dell'*ESA* (*European Space Agency*) e il Russo *Progress* (con equipaggio, *docking* semi-autonomo) per rifornimenti e supporto alla Stazione Spaziale Internazionale (ISS); possibili applicazioni future sono rimozione dei rifiuti spaziali, rifornimento di combustibile a satelliti già in orbita e in generale a missioni di esplorazione interplanetaria. Le prime missioni ad effettuare il *rendezvous* e il *docking* furono la Gemini (dalla 6A) e Apollo (dalla 8) per gli Stati Uniti (con equipaggio), mentre le Cosmos per l'Unione Sovietica (senza equipaggio); ai nostri giorni vengono svolte ancora molte missioni con l'obiettivo di formare il know-how necessario per progetti futuri, alcuni esempi sono:

- *KIKU-7* o *ETS-VII* (*Engineering Test Satellite No. 7*), dell'agenzia aerospaziale Giapponese (JAXA); anno di lancio 1997
- *DART* (*Demonstration of Autonomous Rendezvous Technology*), dell'agenzia spaziale Americana (NASA); anno di lancio 2005
- *Orbital Express*, della DARPA (Defense Advanced Research Projects Agency), Stati Uniti; anno di lancio 2007
- *ESS* (*Experimental Spacecraft System*) missioni *XSS-10* e *XSS-11*, AFRL (Air Force Research Laboratory) degli Stati Uniti d'America

1.2 Scopo del lavoro

Nella fase finale del docking le variabili relative fra i due satelliti quali la velocità, la posizione, l'orientazione e la velocità angolare devono essere controllate in maniera precisa per assicurare le corrette condizioni d'interfaccia dei meccanismi preposti all'aggancio. Lo scopo del lavoro è appunto studiare un sistema integrato di misura e controllo sfruttabile in operazioni di prossimità quali il docking.

Per quanto riguarda il controllo, essendo il problema intrinsecamente non lineare, si è optato per una recente strategia di controllo non-lineare, chiamata metodo $\theta - D$ (Xin, Balakrishnan, Stansbery, Ohlmeyer 2004; Xin, Balakrishnan 2005); esso costituisce una variazione del metodo *SDRE* e garantisce un risparmio computazionale.

Per quanto concerne il sistema di misura si è focalizzata l'attenzione su nuovi algoritmi proposti nel campo della *pose estimation* (Lepetit, Moreno-Noguer Fua 2008; ZHANG Shijie, CAO XiBin, Zhang Fan, HE Liang 2010) effettuando un confronto con l'approccio classico, il filtro di Kalman esteso.

1.3 Struttura della tesi

Nel secondo capitolo vengono presentati gli algoritmi per la stima della posizione e orientazione di un oggetto nello spazio (*pose estimation*); in generale essi non vengono ideati per l'applicazione nel campo spaziale ma hanno validità generale, cioè funzionano qualora vi siano due oggetti in vista, l'uno marcato con punti caratteristici di posizione nota e l'altro dotato di un dispositivo ottico (quale una fotocamera, sistema VISNAV, etc.); gli algoritmi vengono quindi studiati numericamente simulando una possibile prova di laboratorio.

Nel terzo capitolo vengono introdotti i modelli matematici utilizzati per modellare la meccanica orbitale, la dinamica dell'assetto e relativa parametrizzazione. Per quanto riguarda i disturbi si sono considerati quelli derivanti dall'aerodinamica, le forze legate all'asfericità del pianeta e le coppie causate dalla natura tridimensionale del satellite.

Nel quarto capitolo è stata introdotta la tecnica di controllo ottimo non-lineare $\theta - D$ e il filtro di Kalman non lineare; per entrambi è stata effettuata una simulazione numerica per verificarne il funzionamento.

Nel quinto capitolo viene effettuata una simulazione con controllo e algoritmi di *pose estimation*/filtro di Kalman per studiarne il comportamento accoppiato, come poi succede nella pratica; questo capitolo ha quindi lo scopo di verificare la robustezza del controllore e del sistema di misura integrati.

CAPITOLO 2

ESTIMA DELLA POSIZIONE E ORIENTAZIONE DI UN OGGETTO NELLO SPAZIO

La determinazione di una trasformazione che descriva la posizione e l'assetto relativo tra due sistemi di riferimento, partendo dalla corrispondenza tra i punti nello spazio fisico 3D e la loro proiezione sul piano delle immagini 2D, è un problema che trova applicazione in numerosi settori dell'ingegneria (robotica, aerospaziale, etc) grazie all'impiego sempre più diffuso dei sensori visivi (telecamere, fotocamere, sistemi laser?). Quest'ultimi, in effetti, offrono una buona accuratezza con costi relativamente contenuti. Negli anni sono stati sviluppati numerosi algoritmi per ottenere la soluzione del problema, in generale questi possono essere divisi in *I) non-iterativi* e *II) iterativi*. I primi sono adatti ad applicazioni realtime, visto che offrono una soluzione analitica del problema, ma sono generalmente sensibili ai disturbi. Lepetit et al. [**Lepetit**] hanno proposto un nuovo metodo non iterativo che offre un'accuratezza paragonabile ai recenti metodi iterativi e una buona robustezza (si veda paragrafo 2.4). I metodi iterativi, al contrario, sebbene in generale mostrino un'accuratezza superiore (almeno quando convergono) rispetto ai metodi non iterativi, richiedono un elevato costo computazionale, a volte incompatibile con le reali applicazioni.

Nella sfera di questi ultimi un approccio classico, utilizzato in *fotogrammetria*, è quello di formulare il problema ai minimi quadrati e di risolverlo con un algoritmo non lineare ottimizzato come il metodo di Gauss-Newton oppure il metodo di Levenberg-Marquardt; partendo dalla trasformazione rigida che lega i *punti caratteristici* \mathbf{p}_i agli stessi punti ma espressi nel riferimento camera

$$\mathbf{s}_i = R\mathbf{p}_i + \mathbf{t} \quad (2.1)$$

dove \mathbf{s}_i sono i *punti caratteristici* espressi nel riferimento camera, R è la matrice di rotazione e t il vettore traslazione nel riferimento camera; in particolare

$$R = \begin{pmatrix} \mathbf{r}_1^t \\ \mathbf{r}_2^t \\ \mathbf{r}_3^t \end{pmatrix}, \mathbf{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

E' quindi possibile, sfruttando il modello *pin-hole* che verrà introdotto nel paragrafo 2.2.1, esprimere i *punti caratteristici* proiettati sul piano immagine in funzione della matrice di rotazione e del vettore traslazione e, facendo la differenza rispetto a quelli acquisiti dalla fotocamera, formulare la funzione obiettivo da minimizzare

$$F = \sum_{i=0}^n \left[\left(\hat{u}_i - \frac{\mathbf{r}_1^t \mathbf{p}_i + t_x}{\mathbf{r}_3^t \mathbf{p}_i + t_z} \right)^2 + \left(\hat{v}_i - \frac{\mathbf{r}_2^t \mathbf{p}_i + t_y}{\mathbf{r}_3^t \mathbf{p}_i + t_z} \right)^2 \right] \quad (2.2)$$

dove si sono indicati con \hat{u}_i e \hat{v}_i le coordinate sul piano immagine dei *punti caratteristici* misurati; quindi questi metodi cercano R e \mathbf{t} tali da minimizzare la differenza con i valori reali.

Il metodo di Gauss-Newton assicura un'ottima convergenza solo se il punto di partenza è molto vicino alla soluzione reale (empiricamente si parla del 10% rispetto alla distanza e di 15 per ogni asse) altrimenti potrebbe non convergere, mentre il metodo Levenberg-Marquardt assicura convergenza sempre ma più in parte lontano dalla soluzione e più lentamente converge.

Haralick et al. [haralick] ha proposto un nuovo metodo basato sul concetto di *la profondità*, che permette di eliminare le non-linearità proprie dei problemi prospettici 2.2. Tuttavia il metodo mostra una lenta convergenza alla soluzione esatta, rendendolo inadatto per applicazioni *real-time*. Fortunatamente altri autori hanno proposto delle modifiche che hanno reso il metodo efficiente e quindi utilizzabile, ad esempio, per applicazioni spaziali.

Nei paragrafi che seguono verranno presentati due algoritmi iterativi e due non-iterativi, e ne verranno confrontate le prestazioni. Il primo dei non-iterativi è stato sviluppato appositamente per il *docking*, mentre il secondo, di carattere più generale, è quello proposto da Lepetit et al.; per gli iterativi viene presentato quello di Haralick et al. e una sua versione modificata utilizzando l'algoritmo di Umeyama [umeyama] proposta da un'equipe Cinese [china1].

2.1 Sistemi di riferimento, notazione e trasformazioni omogenee

2.1.1 Sistemi di riferimento

Vengono introdotti i sistemi di riferimento utilizzati nel capitolo e visibili in figura 2.1; in particolare

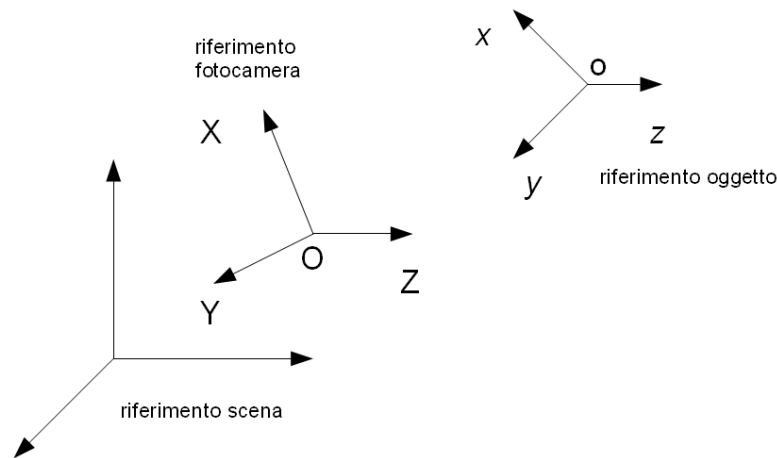


Figura 2.1: rappresentazione dei tre sistemi di riferimento: *scena*, *fotocamera* e *oggetto*

- *riferimento fotocamera*: è solidale con la fotocamera, che a sua volta è fissa su un corpo (in applicazioni spaziali ad esempio può essere il *chaser*), e viene indicato con $O(X, Y, Z)$, la sua origine coincide con il centro di proiezione; ogni punto sul piano immagine ha coordinate $p_i = (u_i, v_i, f)$, con u_i e v_i calcolati con il modello pin-hole (vedi Cap. 2.2.1), e f distanza focale (vedi Cap. 2.2).
- *riferimento oggetto*: è posto *solidale* con il corpo di cui si vuole conoscere l'assetto (in applicazioni spaziali ad esempio può essere il *target*) e la posizione ed è indicato con $o(x, y, z)$, i *punti caratteristici* sono noti in questo riferimento.
- *riferimento immagine*: è bidimensionale ed è collocato in un punto preciso del piano immagine (in generale l'angolo superiore sinistro del sensore *CCD*), le sue componenti sono espresse in *pixel*.
- *riferimento scena*: anche chiamato riferimento mondo, è un sistema di riferimento esterno alla fotocamera e serve a definirne l'orientazione nello spazio; viene utilizzato durante la fase di calibrazione.

2.1.2 Notazione e trasformazioni

Come già anticipato nell'introduzione al capitolo, con \mathbf{s}_i vengono indicati i *punti caratteristici* espressi nel riferimento camera $O(X, Y, Z)$, con \mathbf{p}_i i *punti caratteristici*

stici espressi nel riferimento $o(x, y, z)$, con R la matrice di rotazione e \mathbf{t} il vettore che descrive la posizione relativa dei due riferimenti; tutti questi elementi sono necessari a definire la trasformazione non omogenea 2.1. Ogni trasformazione può venire espressa anche nella forma omogenea

$$\tilde{\mathbf{s}}_i = T\tilde{\mathbf{p}}_i \quad (2.3)$$

semplicemente aggiungendo un'equazione

$$\begin{pmatrix} s_x \\ s_y \\ s_z \\ 1 \end{pmatrix}_i = \underbrace{\begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}}_T \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix}_i \quad (2.4)$$

dove T rappresenta la nuova matrice di trasformazione.

2.2 La fotocamera

La fotocamera è lo strumento che permette di tradurre la realtà in immagini digitali bidimensionali proprio come l'occhio umano. È composta da una lente che convoglia la luce incidente verso un elemento fotosensibile il quale, proporzionalmente all'intensità luminosa, genera dei segnali elettrici; questi vengono poi interpretati da un calcolatore che li traduce in un array di numeri ognuno dei quali rappresenta il livello dell'intensità luminosa incidente in quel particolare punto.

Affinché l'immagine di un punto sia correttamente definita è necessario che tutta la luce riflessa da esso e catturata dalla fotocamera venga fatta convergere verso un unico punto, se ciò avviene si dice che l'immagine è messa *a fuoco*, altrimenti è necessario riuscire a realizzare la condizione in altri modi:

1. introdurre delle ottiche (sistemi di lenti) ad apertura variabile che riescono a far convergere tutti i raggi in un unico punto
2. ridurre l'apertura della fotocamera ad un singolo punto, *pin-hole*, permettendo ad un solo raggio per volta di penetrare e creando quindi una corrispondenza biunivoca fra raggio incidente e punto proiettato

La seconda soluzione è quella che viene normalmente adottata per creare dei modelli di fotocamere verosimili e facilmente gestibili per sviluppare algoritmi di *computer vision* e *pose estimation*; si noti che tale approccio non è applicabile nella realtà perché il tempo di esposizione affinché il sensore riceva una quantità di luce sufficiente sarebbe troppo lungo e quindi di scarsa utilità. Le fotocamere reali infatti, per ovviare ai suddetti problemi, montano delle ottiche con tempi d'esposizione, lunghezza focale e apertura variabili.

2.2.1 Il modello prospettico o *pinhole*

Il modello *pinhole*, o *prospettico*, è caratterizzato dai seguenti elementi:

- *piano immagine*, dove gli oggetti vengono proiettati; è il piano in cui giace il sensore della fotocamera (CCD); fisicamente si trova all'interno della fotocamera e le immagini vi vengono proiettate capovolte, nei modelli viene utilizzato un piano immagine virtuale posto fuori la fotocamera e dove l'oggetto proiettato non è capovolto (figura 2.2); nel seguito si farà sempre riferimento a questa formulazione omettendo l'aggettivo virtuale

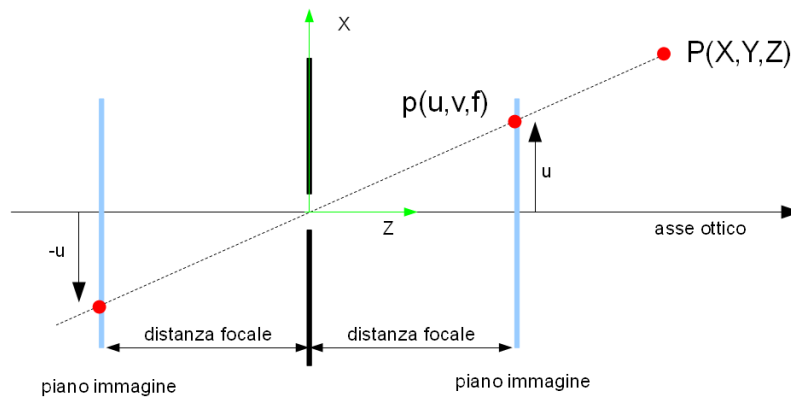


Figura 2.2: Modello prospettico

- *asse ottico*, è un asse immaginario passante per il centro delle lenti e perpendicolare al *piano immagine*
- *distanza focale* o più semplicemente *fuoco*, è il punto sull'*asse ottico* in cui convergono i raggi¹

Se consideriamo un punto $\mathbf{P}=(X,Y,Z)$ la sua proiezione sul piano immagine vale

$$u = f \frac{X}{Z} \quad (2.5)$$

$$v = f \frac{Y}{Z} \quad (2.6)$$

Le relazioni 2.9-2.10 possono essere riscritte come,

$$\mathbf{p} = \begin{pmatrix} fX/Z \\ fY/Z \\ f \end{pmatrix} = f \begin{pmatrix} X/Z \\ Y/Z \\ 1 \end{pmatrix}$$

¹per maggiori informazioni fare riferimento alla *teoria delle lenti sottili*

A causa del termine $1/Z$ il modello *prospettico* non è lineare; introducendo un'ipotesi semplificativa è possibile ovviare a tale problema.

2.2.2 Il modello prospettico debole o *weak-perspective model*

Quando la distanza relativa lungo Z tra due punti è molto inferiore alla distanza, sempre lungo Z , tra l'origine del riferimento e la distanza media dei punti \bar{Z} , allora è possibile modificare il modello come segue

$$u = f \frac{X}{\bar{Z}} = \left(\frac{f}{\bar{Z}} \right) X \quad (2.7)$$

$$v = f \frac{Y}{\bar{Z}} = \left(\frac{f}{\bar{Z}} \right) Y \quad (2.8)$$

Analiticamente può essere interpretato come una successione di due trasformazioni, la prima che proietta i punti sui raggi paralleli all'*asse ottico*

$$u = X$$

$$v = Y$$

e la seconda che esegue una scalatura isotropica di un fattore f/\bar{Z} . Il modello è chiaramente lineare.

2.2.3 Il sensore *CCD (Charged Coupled Design)*

Come già accennato nei paragrafi precedenti il trasduttore della luce in segnali elettrici è il sensore *CCD*; esso è costituito da $n \times m$ fotosensori, ciascuno dei quali converte l'intensità della radiazione luminosa incidente in un potenziale elettrico. La matrice del *CCD* viene convertita in una immagine digitale, ovvero in una matrice $N \times M$ (per esempio 1024×768) di valori interi (per esempio $0 \dots 255$). Gli elementi della matrice prendono il nome di *pixel (picture elements)*.

2.2.4 Il rumore

Ogniquale si ha a che fare con segnali elettrici occorre considerare la presenza del rumore, la cui origine è legata a diversi fattori, quali la temperatura del sensore, la presenza di amplificatori, etc. Nello specifico, la presenza di rumore può determinare una riduzione della qualità dell'immagine digitale (soprattutto in termini di intensità del colore), ossia eventuali imprecisioni nella stima della posizione e dell'orientazione relativa. Inoltre va aggiunto il fatto che l'immagine è effettivamente suddivisa in elementi finiti, i *pixel*, e ciò può ridurre la precisione

con cui si determina la posizione di un punto qualora questo si trovi a cavallo tra due *pixel*. Al fine di tenere in considerazione i fenomeni sopra citati, le eq. 2.9-2.10 sono state riscritte come segue

$$u = f \frac{X}{Z} + err_u = f \left(\frac{X}{Z} + \frac{err_u}{f} \right) \quad (2.9)$$

$$v = f \frac{Y}{Z} + err_v = f \left(\frac{Y}{Z} + \frac{err_v}{f} \right) \quad (2.10)$$

dove err_u e err_v rappresentano il rumore bianco, Gaussiano a media nulla e con varianza dell'ordine del *pixel* (10^{-6} metri). Si noti come nel modello introdotto il fuoco giochi il ruolo fondamentale, infatti, aumentando o diminuendo varia il peso relativo del termine di rumore; ciò potrebbe non essere del tutto verosimile e andrebbe investigato dal punto di vista sperimentale, in ogni caso, per gli scopi del presente lavoro, il modello in questione risulta adatto.

Nelle simulazioni degli algoritmi si è comunque cercato di capire l'entità di questo fenomeno analizzando diverse lunghezze focali, una classica e probabilmente verosimile di 50 mm e due agli estremi opposti, 5 mm e 100 mm ; per quanto riguarda il rumore si sono considerate varianze dell'ordine del pixel in su, quindi 10^{-6} , 10^{-5} e 10^{-4} metri, equivalentemente 1 μm , 10 μm e 100 μm .

2.2.5 Parametri estrinseci

Sono i parametri che definiscono l'orientazione del *riferimento fotocamera* rispetto ad un riferimento noto (*scena*) in cui la fotocamera si trova; la trasformazione che permette di passare dall'uno all'altro non è sempre nota a priori e deve essere determinata con una procedura di *calibrazione*.

Indicando con \mathbf{T} il vettore che unisce i centri dei due riferimenti, con R la matrice di rotazione, e con \mathbf{P}_S (*scena*) e \mathbf{P}_F (*fotocamera*) le coordinate di un generico punto \mathbf{P} espresse nei due riferimenti, possiamo scrivere la relazione

$$\mathbf{P}_F = R(\mathbf{P}_S - \mathbf{T}) \quad (2.11)$$

che ci permette di introdurre la seguente matrice

$$M_{est} = \left[R \quad | \quad -R^T \mathbf{T} \right] \quad (2.12)$$

che descrive una trasformazione omogenea tra le coordinate dei punti caratteristici espressi nel riferimento *scena* e gli stessi espressi nel riferimento *fotocamera*.

2.2.6 Parametri intrinseci

Il modello prospettico non tiene conto della presenza di lenti e del processo di digitalizzazione dell'immagine; la matrice dei parametri intrinseci consente di

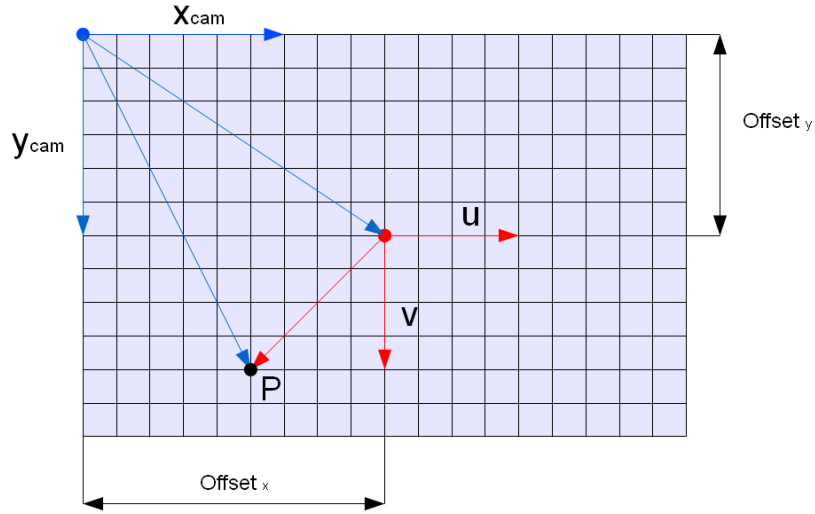


Figura 2.3: vista del piano immagine e dei relativi sistemi di riferimento

estendere il modello prospettico in modo che tenga conto anche di questi aspetti. Il primo aspetto da considerare è la conversione delle coordinate dei punti (u_i, v_i) , localizzati nel riferimento fotocamera, da millimetri a multipli del pixel; in sostanza si dividono le coordinate dei punti per le rispettive dimensioni del pixel (non è detto che i pixel siano quadrati); il secondo aspetto riguarda il posizionamento del riferimento *immagine* che differisce da quello *forocamera* ed è quindi necessario eseguire una traslazione, come visibile in figura 2.3. Indicando con $offset_x$ e $offset_y$ i valori di offset espressi in pixel e Δx e Δy le dimensioni del singolo pixel espresse in millimetri, valgono le seguenti espressioni

$$u = \frac{1}{\Delta x} \frac{f}{Z} X + offset_x = k_u \frac{f}{Z} X + offset_x \quad (2.13)$$

$$v = \frac{1}{\Delta y} \frac{f}{Z} Y + offset_y = k_v \frac{f}{Z} Y + offset_y \quad (2.14)$$

Per quanto riguarda le distorsioni geometriche causate dalla curvatura delle lenti è disponibile un modello non-lineare che ne tiene conto; esse sono infatti maggiori verso i bordi della lente e minori verso il centro di distorsione (in generale coincidente con il centro di proiezione), per questo motivo il modello di uso comune è radiale

$$u = u_d (1 + k_1 r^2 + k_2 r^4) \quad (2.15)$$

$$v = v_d (1 + k_1 r^2 + k_2 r^4) \quad (2.16)$$

$$r^2 = u_d^2 + v_d^2 \quad (2.17)$$

avendo indicato con (u_d, v_d) le coordinate del punto d'interesse; si nota chiaramente come al centro non vi sia distorsione ma aumenti con l'aumentare del raggio. In questo caso i parametri sono le costanti moltiplicative anche se nel seguito sono state considerate trascurabili.

I *parametri intrinseci* risultanti sono f , k_u , k_v , $offset_x$ e $offset_y$. Anche in questo caso è possibile formulare una matrice che racchiude tutti i parametri

$$M_{int} = \begin{bmatrix} k_u f & 0 & offset_x \\ 0 & k_v f & offset_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

e migliora il modello pin-hole incorporando anche le informazioni sulle lenti e sulla digitalizzazione.

2.3 Soluzione non iterativa: approccio diretto al problema

Il seguente algoritmo è stato proposto nel 1995 da R.Mukundan e K.R.Ramakrishnan, e si basa sulla scelta di tre particolari *punti caratteristici* che permettono, sotto un vincolo sulla distanza dei due corpi, di risolvere il problema della determinazione della posizione e dell'orientazione relativa tra due sistemi di riferimento in forma chiusa [mukundan].

2.3.1 Impostazione del problema

L'assetto viene descritto mediante i quaternioni, la matrice di rotazione viene definita

$$R = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_3 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_3q_4 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \quad (2.19)$$

I quaternioni devono inoltre sempre soddisfare il vincolo di normalizzazione

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \quad (2.20)$$

I *punti caratteristici* sono disposti a formare una T e hanno le seguenti coordinate (vedi figura 2.4)

$$\mathbf{p}_1 = \begin{pmatrix} -a \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{p}_2 = \begin{pmatrix} 0 \\ -b \\ 0 \end{pmatrix}, \quad \mathbf{p}_3 = \begin{pmatrix} a \\ 0 \\ 0 \end{pmatrix}$$

L'equazione 2.1, valida per ognuno dei tre *punti caratteristici*, l'equazione 2.20 e le due equazioni 2.9 e 2.10 compongono un sistema di 7 equazioni nonlineari in 7 incognite ($q_1, q_2, q_3, q_4, t_x, t_y, t_z$), la cui soluzione può essere ottenuta mediante un metodo iterativo, quale Newton-Rhapsod.

2.3.2 Soluzione approssimata

In [mukundan], gli autori hanno utilizzato il modello prospettico debole (vedi Cap. 2.2.2) per ottenere una soluzione in forma chiusa al problema della determinazione della posizione e dell'assetto relativo. In particolare

$$t_y u_i = X_i f \quad (2.21)$$

$$t_y v_i = Z_i f \quad (2.22)$$

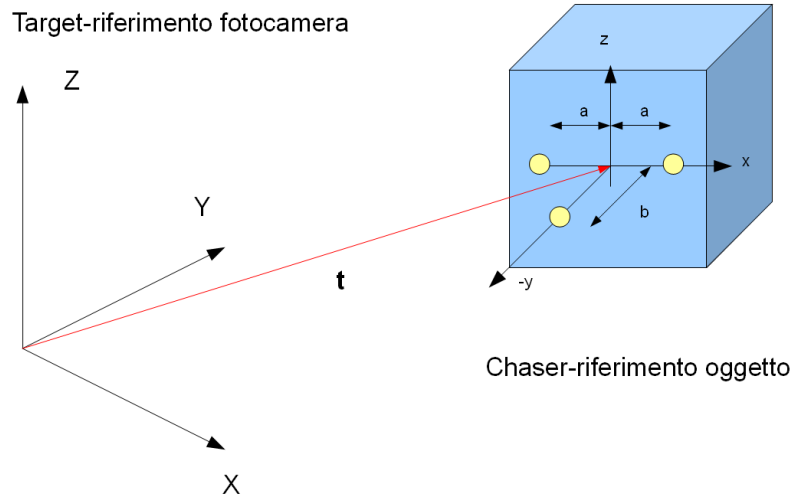


Figura 2.4: rappresentazione dei sistemi di riferimento adottati e dei punti caratteristici (in giallo)

avendo indicato con $(X_i Y_i Z_i)$ le componenti del generico *punto caratteristico* s_i nel riferimento fotocameracamera e con f la distanza focale. L'approssimazione fatta, se da un lato rischia di ridurre l'accuratezza nella stima della posizione e dell'assetto relativo, dall'altro ne semplifica l'architettura matematica, riducendo il sistema da risolvere in un sistema di 5 equazioni in 5 incognite

$$\begin{cases} q_1^2 - q_2^2 - q_3^2 + q_4^2 & = K_1 t_y \\ 2(q_1 q_3 + q_2 q_4) & = K_2 t_y \\ 2(q_1 q_2 + q_3 q_4) & = K_3 t_y \\ 2(q_2 q_3 - q_1 q_4) & = K_4 t_y \end{cases} \quad (2.23)$$

dove

$$\begin{aligned} K_1 &= (u_3 - u_1) / 2af \\ K_2 &= (v_3 - v_1) / 2af \\ K_3 &= (u_1 + u_3 - 2u_2) / 2bf \\ K_4 &= (v_1 + v_3 - 2v_2) / 2bf \end{aligned} \quad (2.24)$$

Le altre due componenti del vettore traslazione valgono

$$\begin{aligned} t_x &= (u_3 + u_1) / 2f \\ t_z &= (v_3 + v_1) / 2f \end{aligned} \quad (2.25)$$

Dalla 2.23 è possibile ricavare l'equazione

$$t_y^4 (K_2 K_3 - K_1 K_4)^2 - t_y^2 (K_1^2 + K_2^2 + K_3^2 + K_4^2) + 1 = 0 \quad (2.26)$$

Ponendo $\lambda = t_y^2$, la 2.26 produce due soluzioni, λ_1 e λ_2 . Come illustrato in [mukundan], se $\lambda_2 \geq \lambda_1$ valgono le seguenti relazioni

$$\lambda_1 \leq \frac{K_1^2 + K_3^2}{(K_2K_3 - K_1K_4)^2} \quad (2.27)$$

$$\lambda_2 \geq \frac{K_2^2 + K_4^2}{(K_2K_3 - K_1K_4)^2} \quad (2.28)$$

La 2.28 è soddisfatta solamente per angoli superiori ai 60° . La soluzione è quindi $t_y = \sqrt{\lambda_1}$ presa con segno positivo. Definendo la costante

$$\beta = \frac{(K_2K_3 - K_1K_4)t_y^2 - K_4t_y}{2} \quad (2.29)$$

si ricavano i quaternioni

$$q_4 = \pm \sqrt{\frac{(1 + K_1t_y) + \sqrt{(1 + K_1t_y)^2 - 4\beta^4}}{4}} \quad (2.30a)$$

$$q_1 = \frac{\beta}{2q_4} \quad (2.30b)$$

$$q_2 = t_y \frac{K_2q_4 - K_3q_1}{2(q_4^2 - q_1^2)} \quad (2.30c)$$

$$q_3 = t_y \frac{K_2q_1 - K_3q_4}{2(q_1^2 - q_4^2)} \quad (2.30d)$$

Quando i tre punti sono collineari occorre fare attenzione che l'equazione 2.26 diviene di secondo grado e risolverla di conseguenza.

2.3.3 Soluzione con correzione

Nell'ipotesi adottata dagli autori [mukundan] per semplificare il sistema il punto caratteristico che meno la rispetta è \mathbf{p}_2 che si trova ad una distanza $-b$, è quindi possibile modificare le equazioni 2.21 e 2.22 come segue

$$(t_y - b) u_2 = X_2 f \quad (2.31)$$

$$(t_y - b) v_2 = Z_2 f \quad (2.32)$$

Se si definiscono due ulteriori costanti

$$K_3' = u_2/f \quad (2.33)$$

$$K_4' = v_2/f \quad (2.34)$$

è possibile riscrivere una nuova formulazione per il parametro β

$$\beta = \frac{(K_2K_3 - K_1K_4)t_y^2 - K_4t_y + (K_2K'_3 - K_1K'_4)t_y - K'_4}{2} \quad (2.35)$$

e quindi i quaternioni valgono

$$q_4 = \pm \sqrt{\frac{(1 + K_1t_y) + \sqrt{(1 + K_1t_y)^2 - 4\beta^4}}{4}} \quad (2.36a)$$

$$q_1 = \frac{\beta}{2q_4} \quad (2.36b)$$

$$q_2 = \frac{t_y(K_2q_4 - K_3q_1) - K'_3q_1}{2(q_4^2 - q_1^2)} \quad (2.36c)$$

$$q_3 = t_y \frac{t_y(K_2q_1 - K_3q_4) - K'_3q_4}{2(q_1^2 - q_4^2)} \quad (2.36d)$$

2.3.4 Simulazione numerica

Al fine di verificare l'efficacia dell'algoritmo, sono state riprodotte tre possibili configurazioni d'assetto relativo e si è analizzato il comportamento dell'algoritmo in relazione a:

- variazione della distanza relativa
- presenza dei disturbi
- distanza focale
- posizione dei punti caratteristici

Variazione della distanza e dell'assetto relativo

Al fine di verificare quanto l'uso di un modello prospettico debole incida sull'accuratezza dell'algoritmo, sono state condotte diverse simulazioni corrispondenti a due set di rotazioni relative (r_1 e r_2) e sono stati riportati i risultati ottenuti in funzione della posizione relativa (range da 2 a 36 m). Nello specifico, la distanza focale è stata imposta costante e pari a 50 mm, e i punti caratteristici sono stati collocati ad una distanza dal centro del sistema di riferimento vincolato al target tale per cui $a = 0,5$ e $b = 0,6$.

Le rotazioni intorno agli assi \mathbf{x} , \mathbf{y} e \mathbf{z} sono descritte dai rispettivi tre angoli ϕ , θ e ψ ; sono stati analizzati due set di rotazioni

$$r_1 = \begin{pmatrix} 3^\circ \\ -2^\circ \\ 6^\circ \end{pmatrix} \quad r_2 = \begin{pmatrix} -30^\circ \\ 6^\circ \\ 35^\circ \end{pmatrix}$$

Come illustrato in figura 2.5, 2.6 l'accuratezza dell'algoritmo migliora all'aumentare della distanza, anche se in ogni caso gli errori, soprattutto sugli angoli, sono troppo elevati perché l'algoritmo possa essere impiegato in applicazioni di docking.

Introduzione dei disturbi

Per poter studiare la sensibilità ai disturbi, il modello prospettico debole è stato modificato come riportato in 2.9-2.10. In particolare sono stati considerati valori di deviazione standard pari a $\sigma = 10^{-4}, 10^{-5}, 10^{-6} m$. Dalle figure 2.7, 2.8, 2.9 si comprende come l'algoritmo non sia robusto, in particolare il grafico riportato in figura 2.7 presenta delle forti oscillazioni che in alcuni casi portano alla *failure* dell'algoritmo; quando la varianza è superiore a $10^{-4} m$ l'algoritmo non produce mai risultati. Per valori inferiori le oscillazioni, benché ridotte, sono comunque concentrate nella zona in cui l'algoritmo dovrebbe essere più preciso rendendolo inutilizzabile per applicazioni pratiche.

Variazione della distanza focale

Come già introdotto nel paragrafo 2.2.4 esiste un legame fra l'ampiezza del disturbo e il valore del fuoco, ciò che vogliamo capire è l'entità di questo legame e se può in qualche modo influenzare le scelte di progettazione; a tal riferimento si analizza un set di rotazioni con varianza del rumore pari a $10^{-4} m$ (caso peggiore) e si varia la distanza focale; dal confronto della figura 2.10 ($f = 100 mm$) con la 2.7 ($f = 50 mm$) si nota un leggero miglioramento dei risultati; invece, dal confronto con la figura 2.11 ($f = 5 mm$), si nota un peggioramento più marcato.

Variazione disposizione punti caratteristici

E' interessante vedere se la distanza dei *punti caratteristici* ha qualche influenza sui risultati, a tal proposito sono stati distanziati e ravvicinati, ma senza evidenziare una particolare influenza sulla soluzione; ciò che si è registrato è che alcuni valori portano alla *failure* l'algoritmo.

2.3.5 Conclusioni sull'algoritmo

Per quanto rappresenti un elegante soluzione al problema manca completamente di precisione e di robustezza, quindi inapplicabile in qualsiasi sistema reale; eventualmente può venir utilizzato per fornire i valori di *guess* per un algoritmo iterativo.

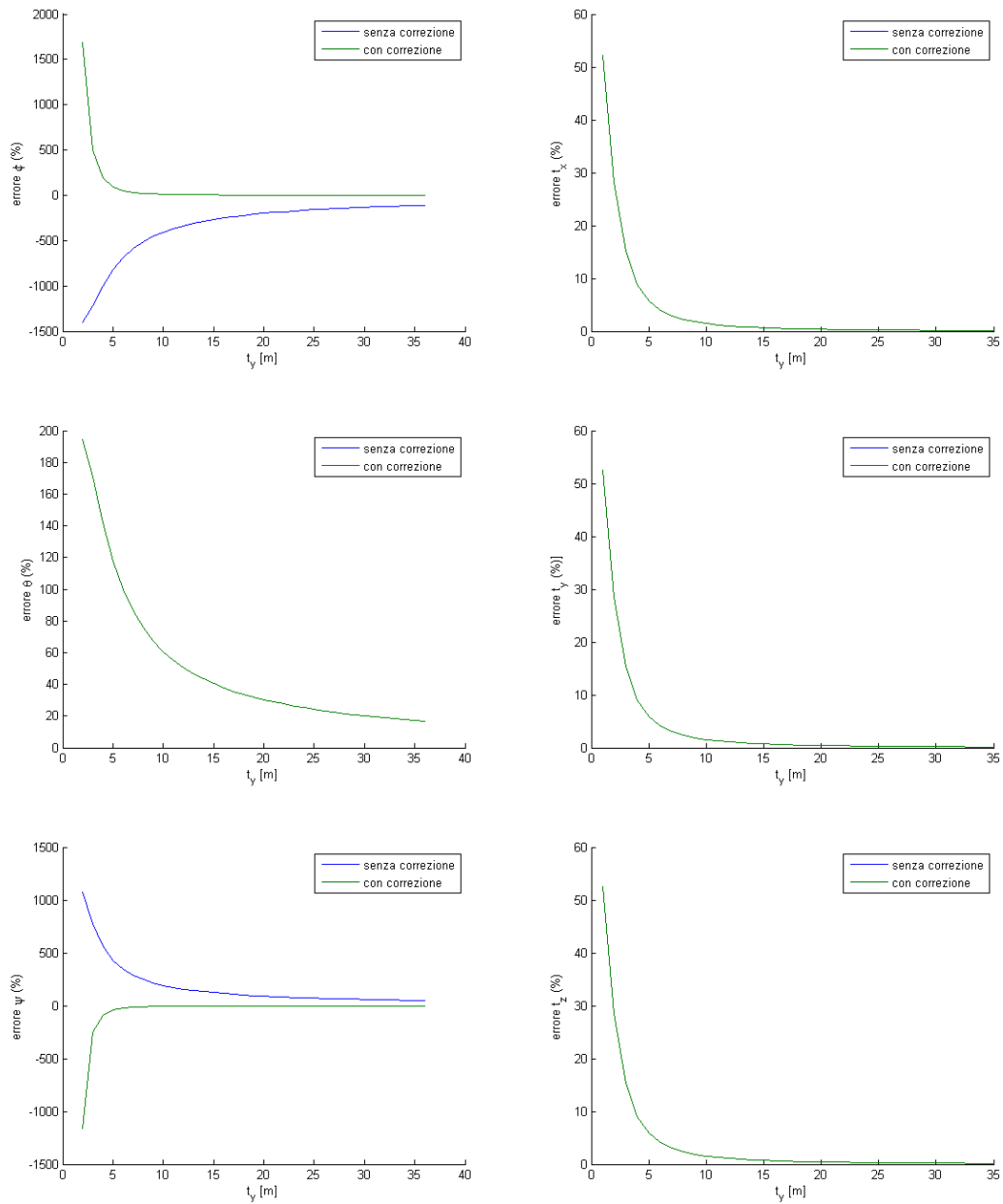


Figura 2.5: Algoritmo di Mukundan; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $a=0,5$ m, $b=0,6$ m, $f=50$ mm, $\sigma_{err} = 0$ m

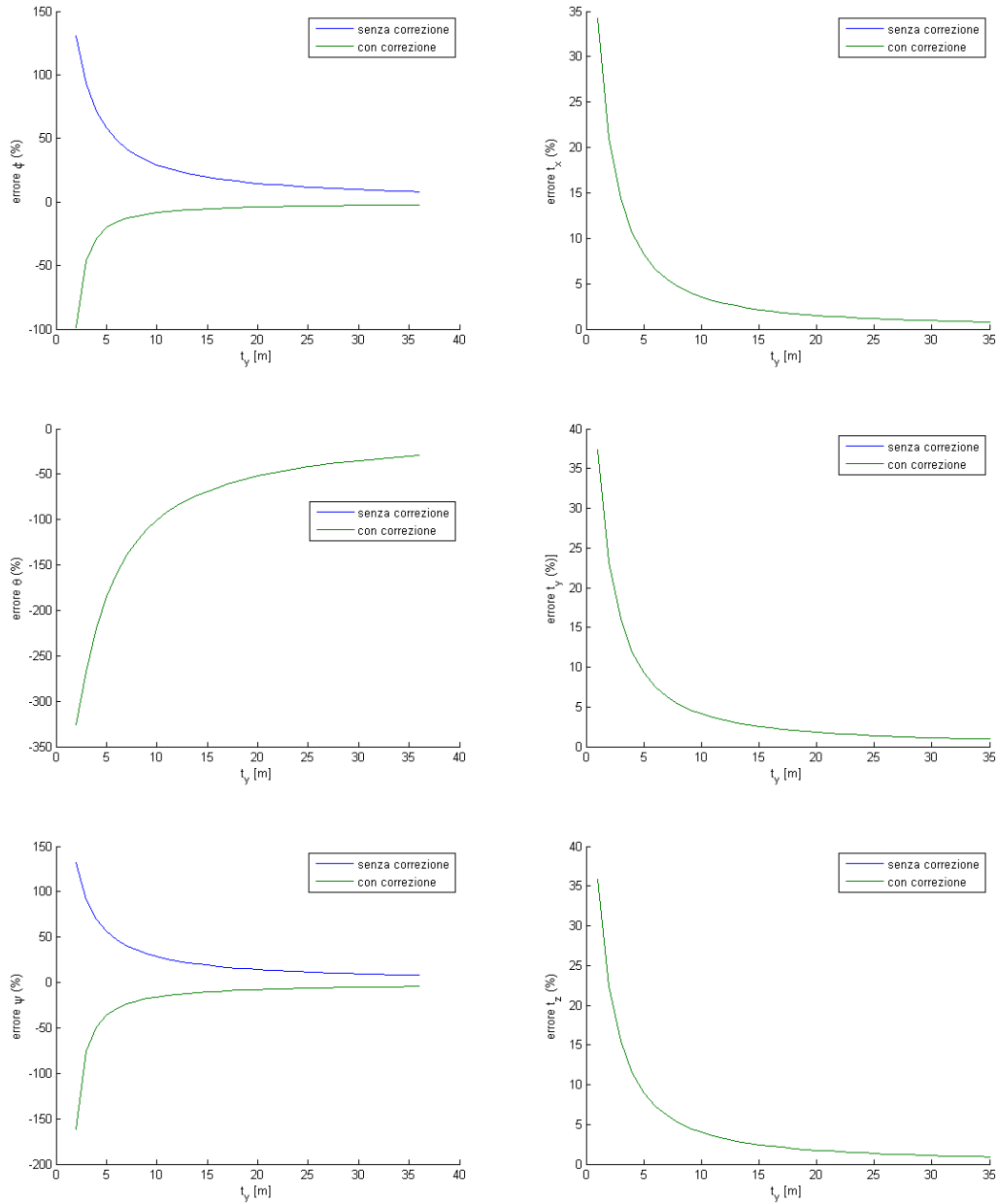


Figura 2.6: Algoritmo di Mukundan; valori reali $\phi = -30^\circ$, $\theta = 6^\circ$, $\psi = 35^\circ$, $a=0,5$ m, $b=0,6$ m, $f=50$ mm, $\sigma_{err} = 0$ m

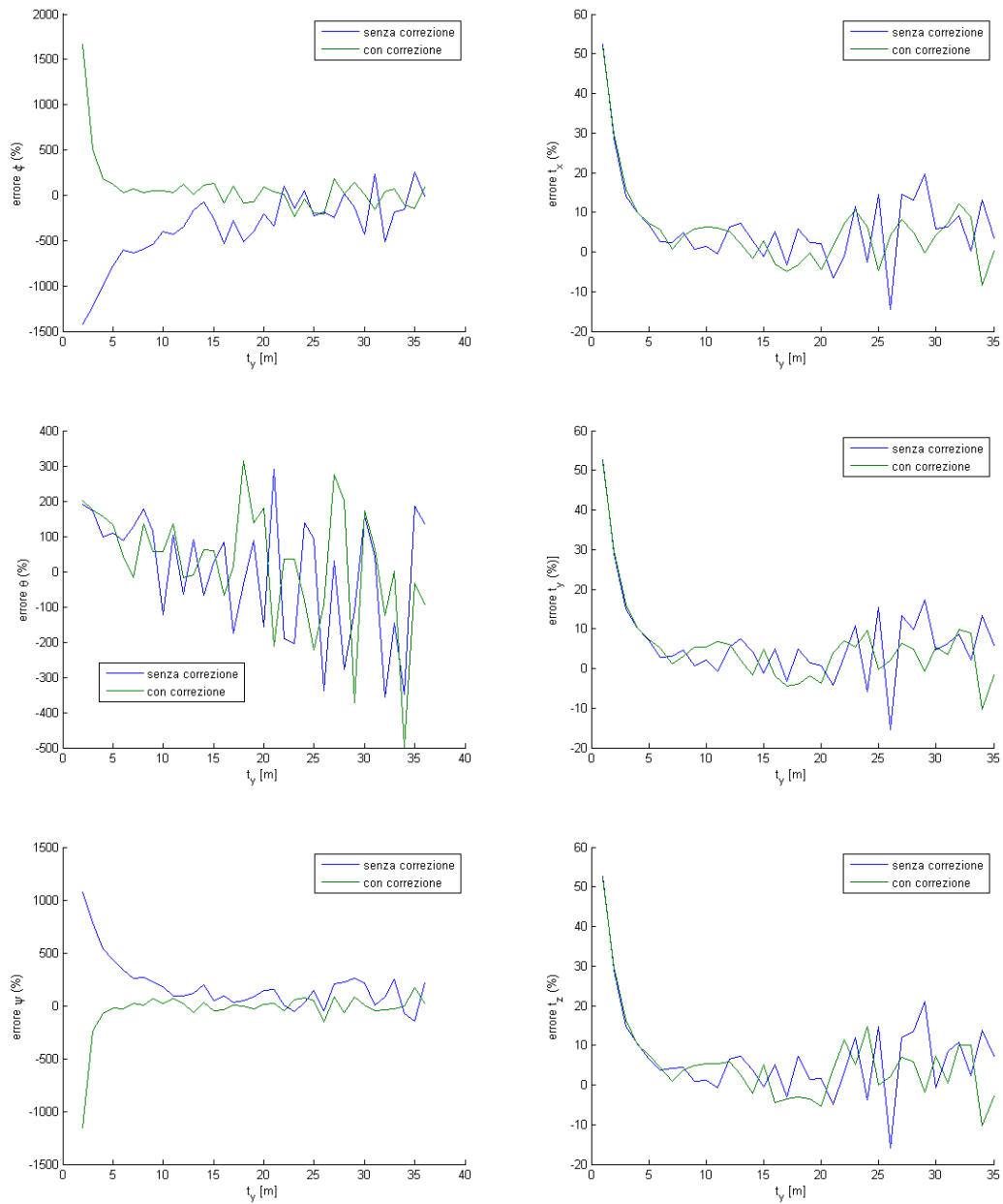


Figura 2.7: Algoritmo di Mukundan; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $a=0,5$ m, $b=0,6$ m, $f=50$ mm, $\sigma_{err} = 10^{-4}$ m

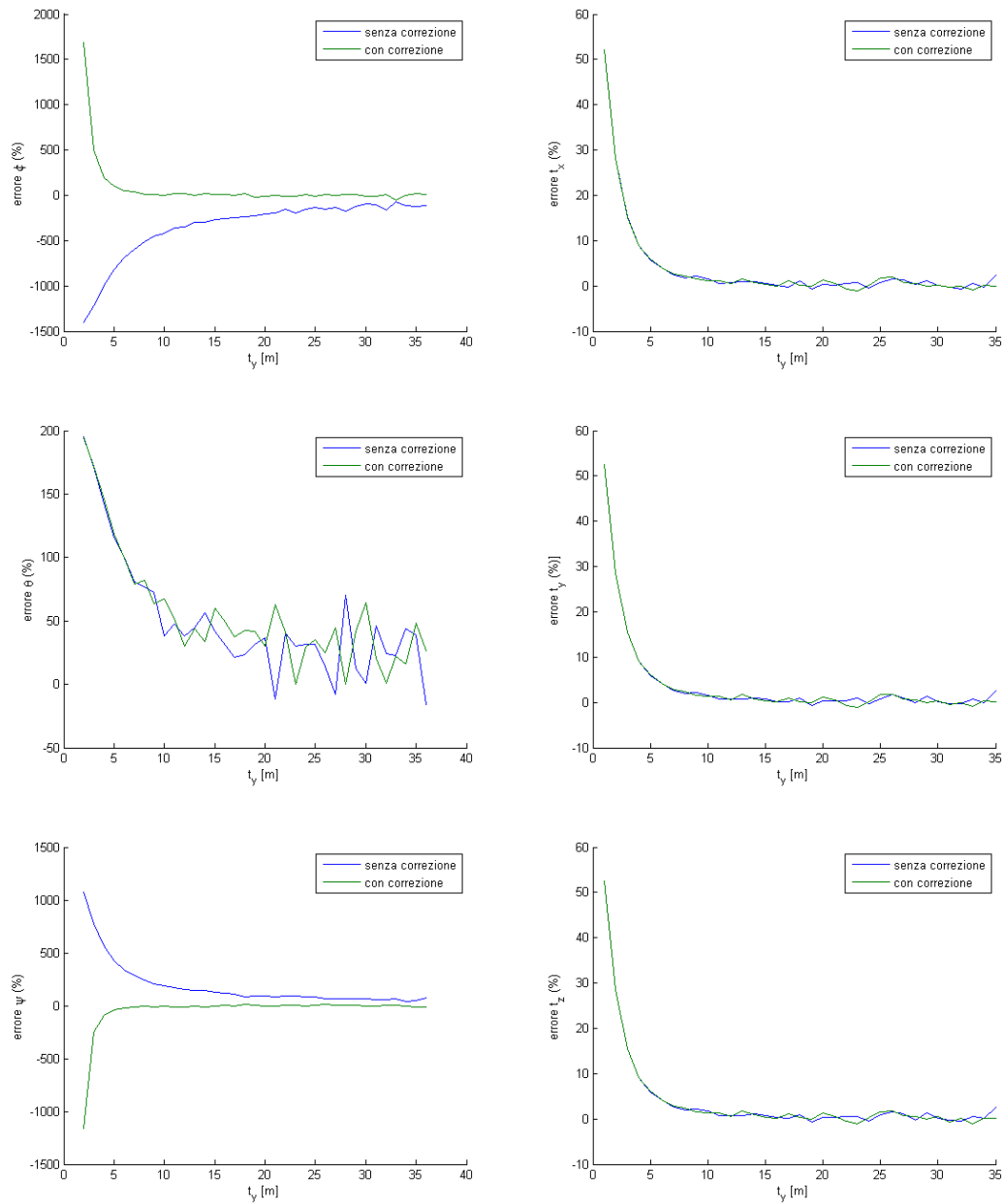


Figura 2.8: Algoritmo di Mukundan; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $a=0,5$ m, $b=0,6$ m, $f=50$ mm, $\sigma_{err} = 10^{-5}$ m

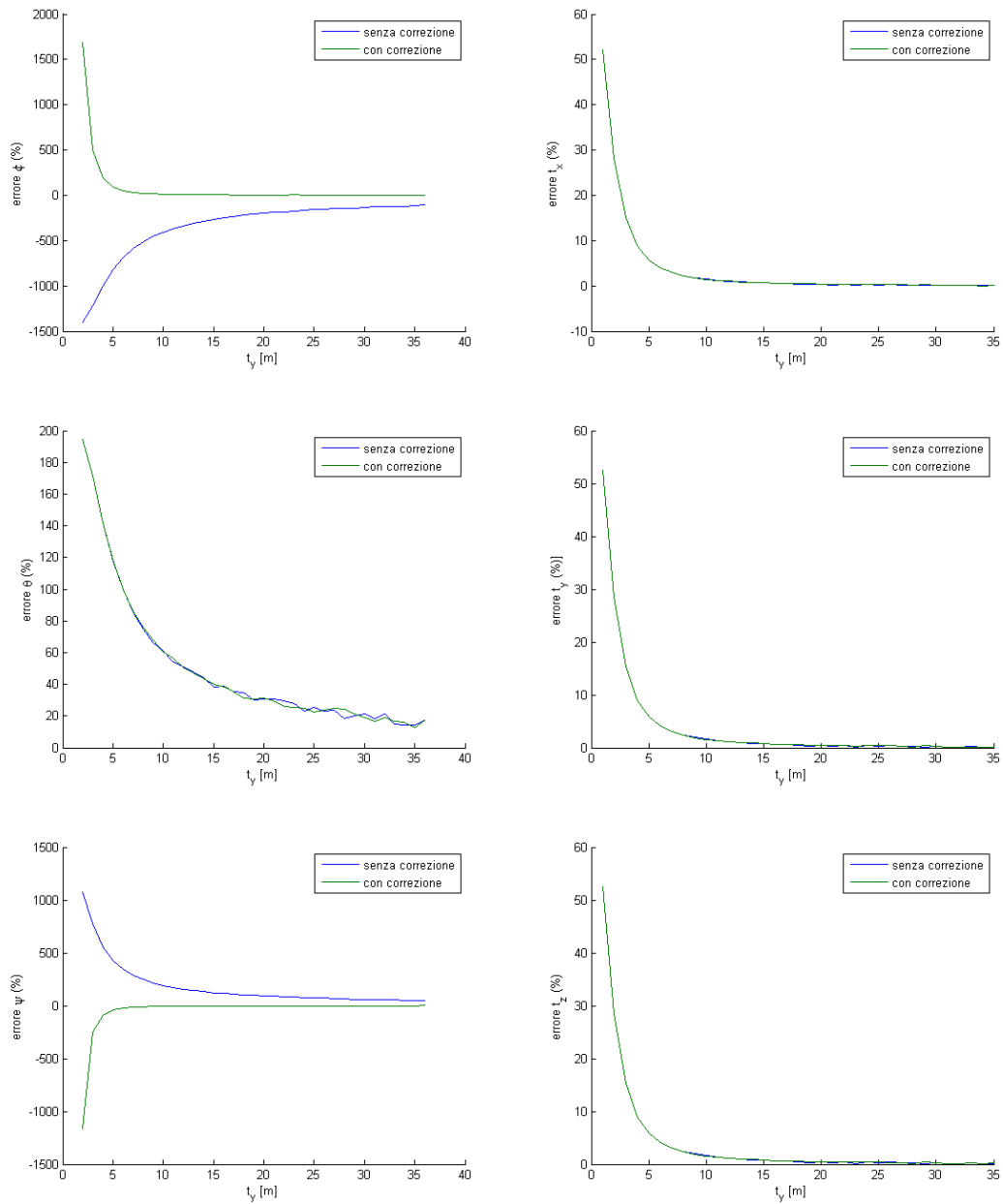


Figura 2.9: Algoritmo di Mukundan; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $a=0,5$ m, $b=0,6$ m, $f=50$ mm, $\sigma_{err} = 10^{-6}$ m

22CAPITOLO 2. Stima della posizione e orientazione di un oggetto nello spazio

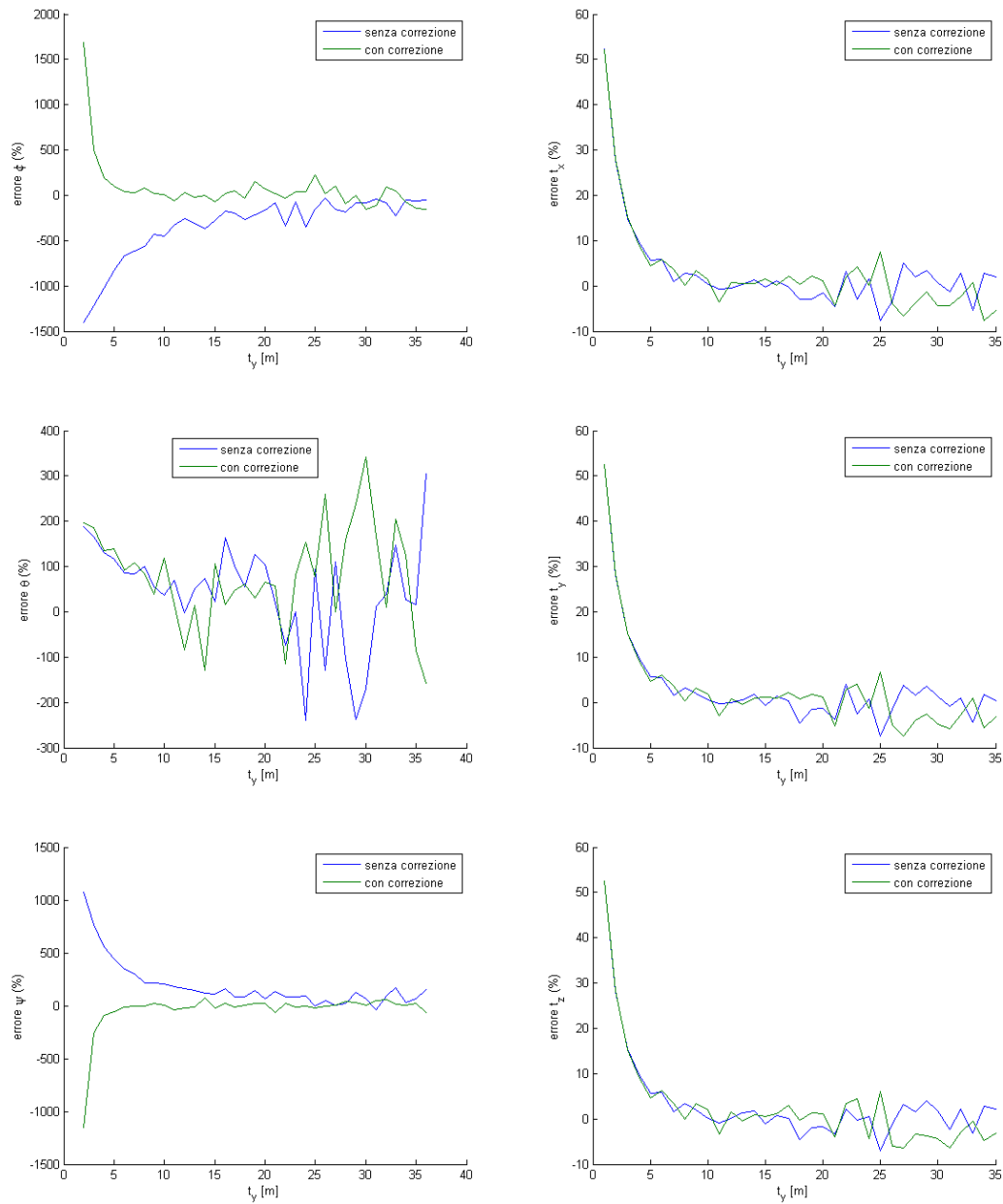


Figura 2.10: Algoritmo di Mukundan; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $a=0,5$ m, $b=0,6$ m, $f=100$ mm, $\sigma_{err} = 10^{-4}$ m

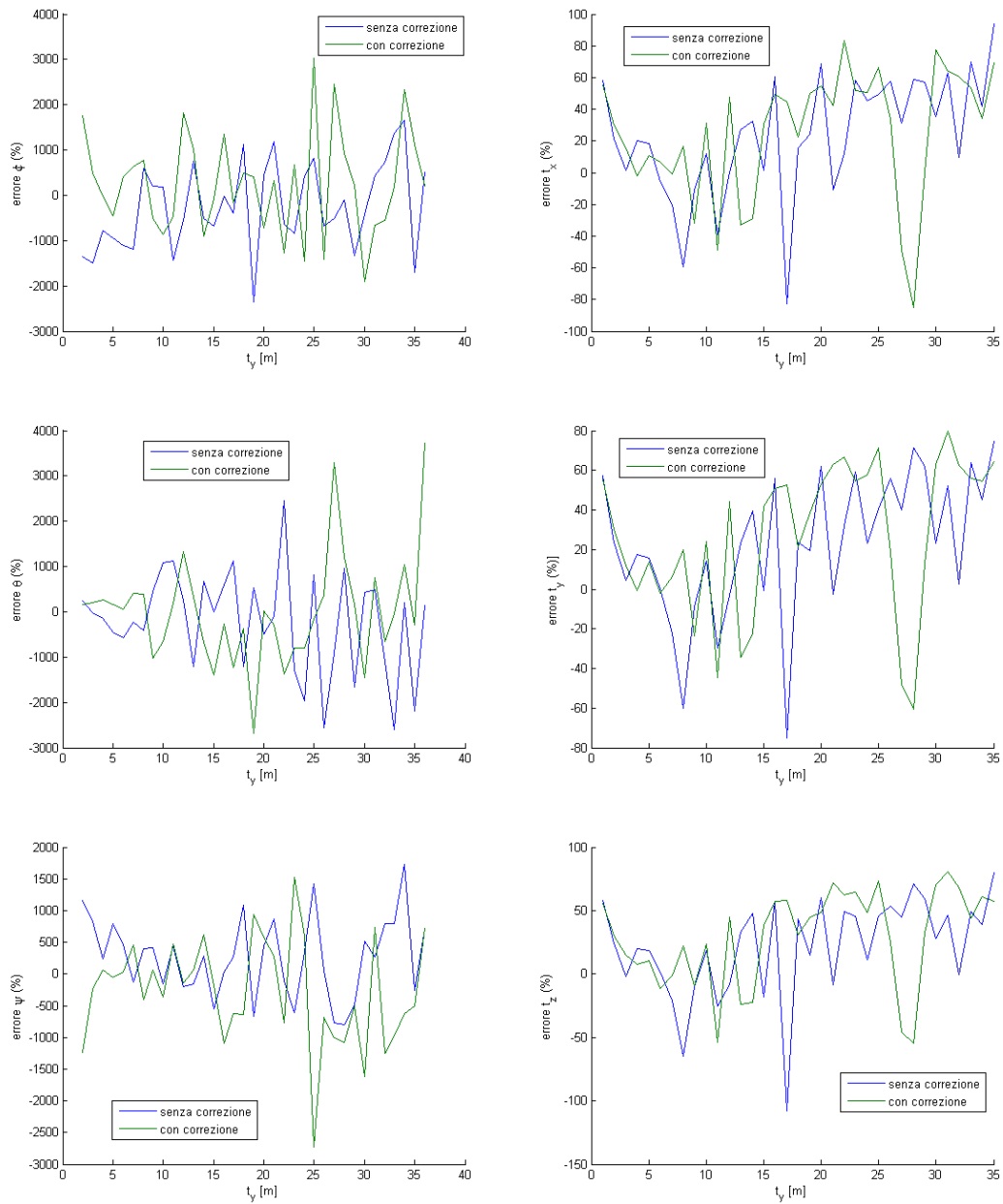


Figura 2.11: Algoritmo di Mukundan; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $a=0,5$ m, $b=0,6$ m, $f=5$ mm, $\sigma_{err} = 10^{-4}$ m

2.4 Soluzione non iterativa: algoritmo di Lepetit (EPnP)

Nel capitolo che segue verrà presentato un algoritmo non-iterativo proposto da Vincent Lepetit, Francesc Moreno-Noguer e Pascal Fua nel 2008 [Lepetit].

2.4.1 Impostazione del problema

Nella formulazione degli algoritmi iterativi, una svolta fondamentale, è avvenuta quando si è capito di effettuare un cambio di variabili e di considerare come incognite del problema le *profondità* dei *punti caratteristici* in coordinate camera; anche in questo algoritmo avviene un cambio di variabili ma non in favore delle *profondità* bensì di 4 *punti di controllo* virtuali che, in somma pesata, permettono di riesprimere le coordinate dei punti caratteristici.

Utilizzando l'apice per specificare il sistema di riferimento (*target* e *camera*) è possibile scrivere le seguenti relazioni

$$\mathbf{p}_i^t = \sum_{j=1}^4 \alpha_{ij} \mathbf{c}_j^t \quad (2.37)$$

$$\mathbf{p}_i^c = \sum_{j=1}^4 \alpha_{ij} \mathbf{c}_j^c \quad (2.38)$$

$$\sum_{j=1}^4 \alpha_{ij} = 1 \quad (2.39)$$

avendo indicato con \mathbf{p} i *punti caratteristici*, \mathbf{c} i *punti di controllo* e con α i *pesi*; questi ultimi coincidono con le coordinate omogenee del baricentro dei punti caratteristici.

Introducendo la matrice di calibrazione intrinseca M_{int} , la proiezione dei punti sul *piano immagine* (u_i, v_i) e i *parametri scalari di proiezione* w_i possiamo scrivere

$$w_i \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = M_{int} \mathbf{p}_i^c = M_{int} \sum_{j=1}^4 \alpha_{ij} \mathbf{c}_j^c \quad (2.40)$$

in forma estesa

$$w_i \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = \begin{bmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{bmatrix} \sum_{j=1}^4 \alpha_{ij} \begin{pmatrix} x_j^c \\ y_j^c \\ z_j^c \end{pmatrix} \quad (2.41)$$

Le incognite del problema sono le coordinate dei quattro punti di controllo, quindi 12 scalari, e degli n *parametri scalari di proiezione*, per un totale di $12+n$.

Sviluppando l'ultima equazione e sostituendola nelle altre si riesce a scrivere 2 equazioni per ogni *punto di controllo*

$$\sum_{j=1}^4 \alpha_{ij} f_u x_j^c + \alpha_{ij} (u_c - u_i) z_j^c = 0 \quad (2.42)$$

$$\sum_{j=1}^4 \alpha_{ij} f_v x_j^c + \alpha_{ij} (v_c - v_i) z_j^c = 0 \quad (2.43)$$

che in forma matriciale costituisce un sistema lineare del tipo

$$M\mathbf{x} = 0$$

il cui vettore delle incognite è della forma $\mathbf{x} = [\mathbf{c}_1^{cT} \ \mathbf{c}_2^{cT} \ \mathbf{c}_3^{cT} \ \mathbf{c}_4^{cT}]^T$; come si evince dalle formule appena scritte i *parametri proiettivi* non compaiono più nella formulazione del problema.

2.4.2 Soluzione del sistema

Gli autori, sempre in [Lepetit], propongono di rintracciare la soluzione nello spazio nullo della matrice M, chiamato in inglese *kernel*, e viene espressa nella forma

$$\mathbf{x} = \sum_{i=1}^N \beta_i \mathbf{v}_i \quad (2.44)$$

dove il set di vettori \mathbf{v}_i è costituito dagli *autovettori nulli* della matrice $M^T M$; occorre quindi scegliere i coefficienti β_i affinché la combinazione lineare rappresenti correttamente la soluzione; il valore N rappresenta la dimensione del sottospazio nullo e varia da 1 fino a 4.

L'algoritmo ricava la soluzione in tutti e quattro i casi e per ognuno calcola l'*errore di riproiezione*, ovvero applica il modello pin-hole al contrario per calcolare i valori dei punti sul piano immagine partendo da \mathbf{p}_i^w , R e \mathbf{t} appena stimati e ne calcola la distanza coi valori veri (*measure*)

$$\begin{aligned} \text{errore}^2 &= \sum_{i=1}^n \left((\tilde{u}_i - u_i)^2 + (\tilde{v}_i - v_i)^2 \right) \\ &= \sum_{i=1}^n \left(\left(\tilde{u}_i - [1 \ 0] A [R|\mathbf{t}] \begin{bmatrix} \mathbf{p}_i^w \\ 1 \end{bmatrix} \right)^2 + \left(\tilde{v}_i - [0 \ 1] A [R|\mathbf{t}] \begin{bmatrix} \mathbf{p}_i^w \\ 1 \end{bmatrix} \right)^2 \right) \end{aligned} \quad (2.45)$$

A monte di quest'ultima formula i valori di R e di \mathbf{t} sono stati calcolati con un algoritmo di *3D-3D pose estimation* simile a quello di Umeyama (vedi Cap. 2.6.4 oppure [umeyama]).

In [Lepetit] gli autori illustrano la soluzione alla 2.44 per $N = 1, 2, 3, 4$.

2.4.3 Ottimizzazione con Gauss-Newton

I quattro coefficienti β possono essere determinati in modo tale che minimizzino la distanza fra i punti di controllo espressi nel riferimento *fotocamera* e degli stessi espressi nel riferimento *oggetto*

$$errore = \sum \left(\left| \mathbf{c}_i^c - \mathbf{c}_j^c \right|^2 - \left| \mathbf{c}_i^w - \mathbf{c}_j^w \right|^2 \right) \quad (2.46)$$

I punti di controllo nel riferimento *oggetto* sono noti, mentre nel riferimento *fotocamera* vengono espressi come combinazione lineare dei β_i

$$\mathbf{c}_i^c = \sum_{j=1}^4 \beta_j \mathbf{v}_j \quad (2.47)$$

All'aumentare del numero di *punti caratteristici* non deve corrispondere anche un aumento dei *punti di controllo*, e quindi la funzione espressa in 2.46 non cambia dimensione; tutto ciò si riflette sul costo computazionale dell'ottimizzazione che può non crescere all'aumentare del numero di punti caratteristici considerati.

2.4.4 Simulazione numerica

Le simulazioni vengono effettuate considerando i seguenti *punti caratteristici*

$$p_1 = \begin{pmatrix} 0 \\ -0.2 \\ 0 \end{pmatrix} \quad p_2 = \begin{pmatrix} -0.2 \\ 0 \\ -0.3 \end{pmatrix} \quad p_3 = \begin{pmatrix} 0.2 \\ 0 \\ 0 \end{pmatrix} \quad p_4 = \begin{pmatrix} 0 \\ 0.2 \\ 0 \end{pmatrix}$$

dove ogni coordinata è espressa in metri. Le rotazioni intorno agli assi \mathbf{x} , \mathbf{y} e \mathbf{z} sono descritte dai rispettivi tre angoli ϕ , θ e ψ ; sono state analizzate due rotazioni già descritte al capitolo 2.3.4. Per ogni simulazione è stata effettuata anche la correzione con l'ottimizzazione di Gauss-Newton. Un confronto numerico accurato è consultabile al paragrafo 2.7.

Variazione assetto e distanza relativa

Questa serie di simulazioni prevede la variazione dell'assetto e della distanza relativa ma senza considerare i disturbi.

Una prima simulazione con 4 *punti caratteristici*, riportata in figura 2.12, ha dato esiti completamente negativi, sia con che senza l'ottimizzazione; si è quindi deciso di aggiungere un quinto punto di coordinate

$$p_5 = \begin{pmatrix} 0 \\ -0,2 \\ 0 \end{pmatrix} m$$

I risultati, ora soddisfacenti, riportati in figura 2.13; in particolare, con meno di cinque iterate del metodo di Gauss-Newton, l'errore manifesta un forte abbattimento; esulando dalla correzione, che rende il metodo iterativo, l'errore ottenuto dal solo EP n P è comunque contenuto a dimostrazione che il metodo interpreta correttamente il problema.

Introduzione disturbi

Vengono qui introdotti i rumori di misura caratterizzati da ($\sigma = 10^{-4}, 10^{-5}, 10^{-6} m$) sul primo set di rotazioni e traslazioni; i risultati sono riportati in figura 2.15, 2.16 e 2.17, si nota che la precisione diminuisce all'aumentare dell'entità del rumore e, soprattutto, l'errore sulla determinazione degli angoli è maggiore rispetto a quello del vettore \mathbf{t} ; si nota infine un peggioramento dei risultati dell'EP n P all'aumentare della distanza relativa.

Variazione distanza focale

Come già introdotto nel paragrafo 2.2.4 esiste un legame fra l'ampiezza del disturbo e il valore del fuoco, ciò che vogliamo capire è l'entità di questo legame e se può in qualche modo influenzare le scelte di progettazione; a tal riferimento si analizza un set di rotazioni con disturbo intenso e si varia la distanza focale; dal confronto delle figure 2.18 e 2.19 con la figura 2.15 si nota che un aumento del fuoco induce un miglioramento delle prestazioni mentre un aumento un peggioramento; questi andamenti, contrariamente agli altri algoritmi, sono molto meno marcati; si nota infine che i risultati dopo l'ottimizzazione sono sempre ottimi e soprattutto non manifestano una dipendenza dalla distanza \mathbf{t} .

Variazione distanza punti caratteristici

La distanza dei punti caratteristici non è particolarmente influente sui risultati se non nel caso in cui vengano presi tutti con la terza coordinata nulla; in particolare l'algoritmo fornisce che non forti picchi di errore sia nella versione non iterativa sia con la correzione di Gauss-Newton.

2.4.5 Conclusioni sull'algoritmo

E' sicuramente l'algoritmo più preciso fra quelli testati, è sufficientemente robusto ai disturbi e l'ottimizzatore converge in meno di cinque iterazioni; in quest'ultima configurazione, sebbene diventi iterativo, mostra una precisione eccellente e grande robustezza, nonché totale indipendenza dalla distanza relativa fra i due oggetti.

Nella configurazione semplice, quindi non iterativa, i risultati sono comunque ottimi e paragonabili, se non superiori, agli algoritmi iterativi.

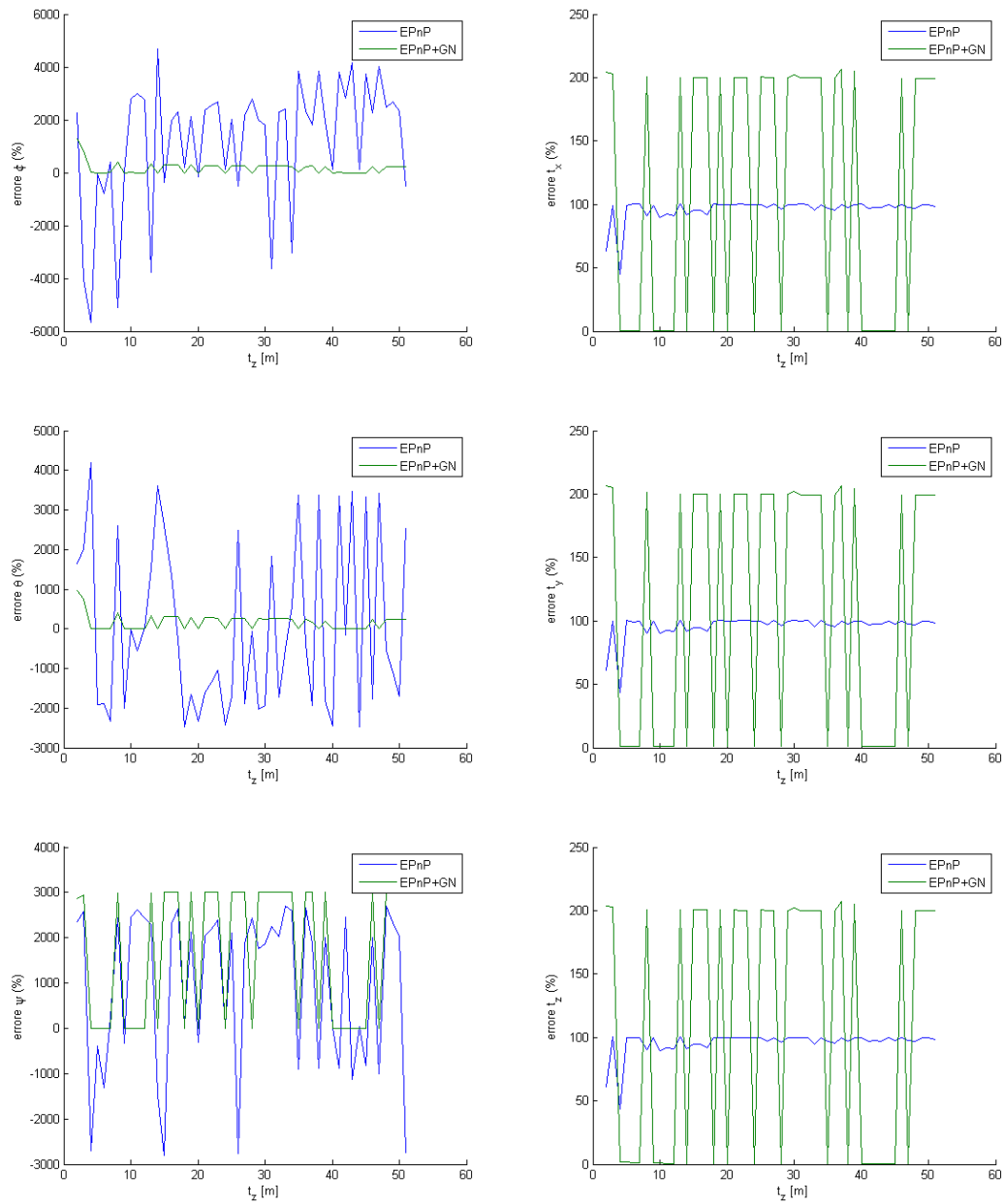


Figura 2.12: Algoritmo EPnP; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $f=50$ mm, $\sigma_{err} = 0$ m, 4 punti caratteristici

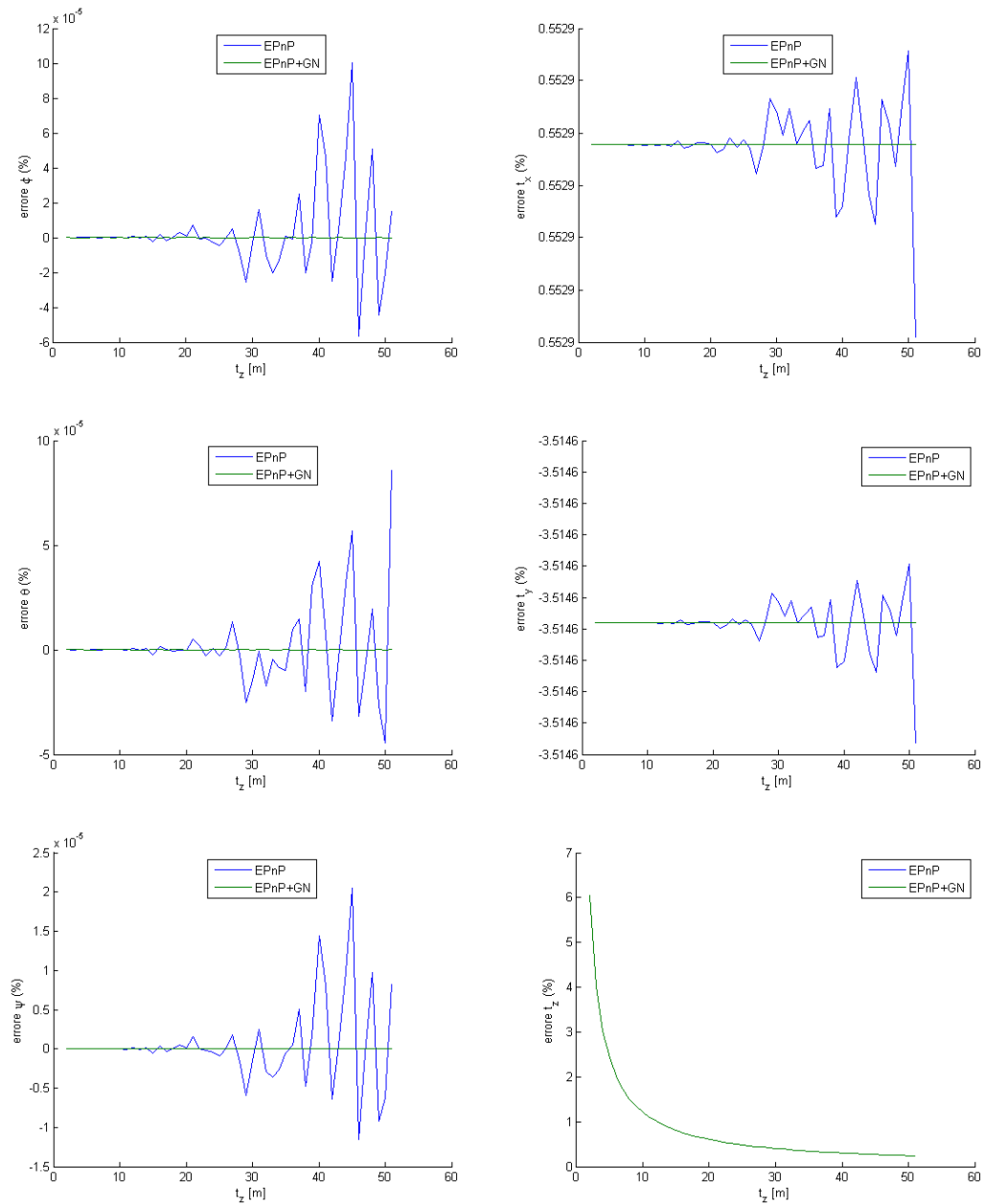


Figura 2.13: Algoritmo EP n P; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $f=50$ mm, $\sigma_{err} = 0$ m, 5 punti caratteristici

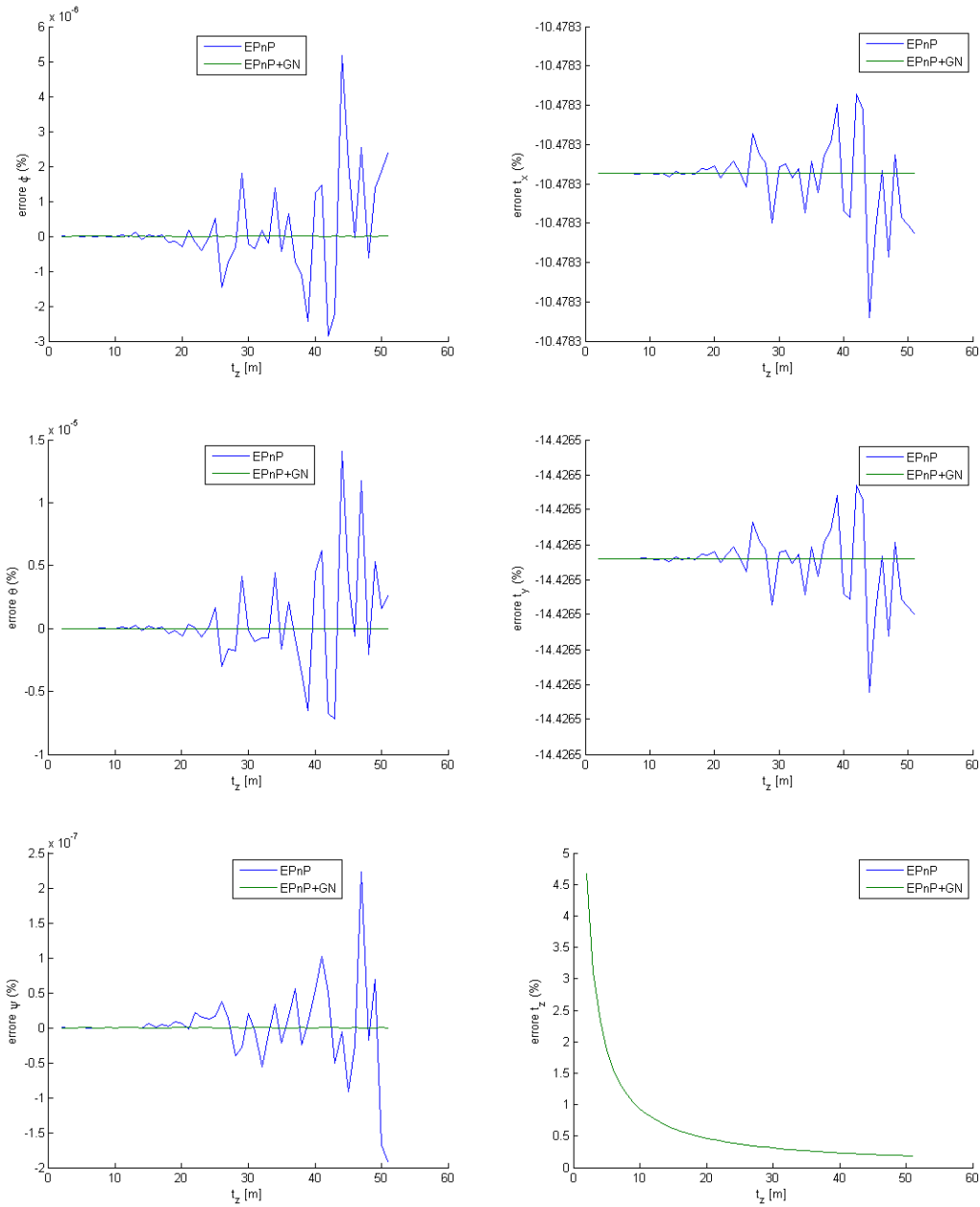


Figura 2.14: Algoritmo EPnP; valori reali $\phi = -30^\circ$, $\theta = 6^\circ$, $\psi = 35^\circ$, $f=50$ mm, $\sigma_{err} = 0$ m, 5 punti caratteristici

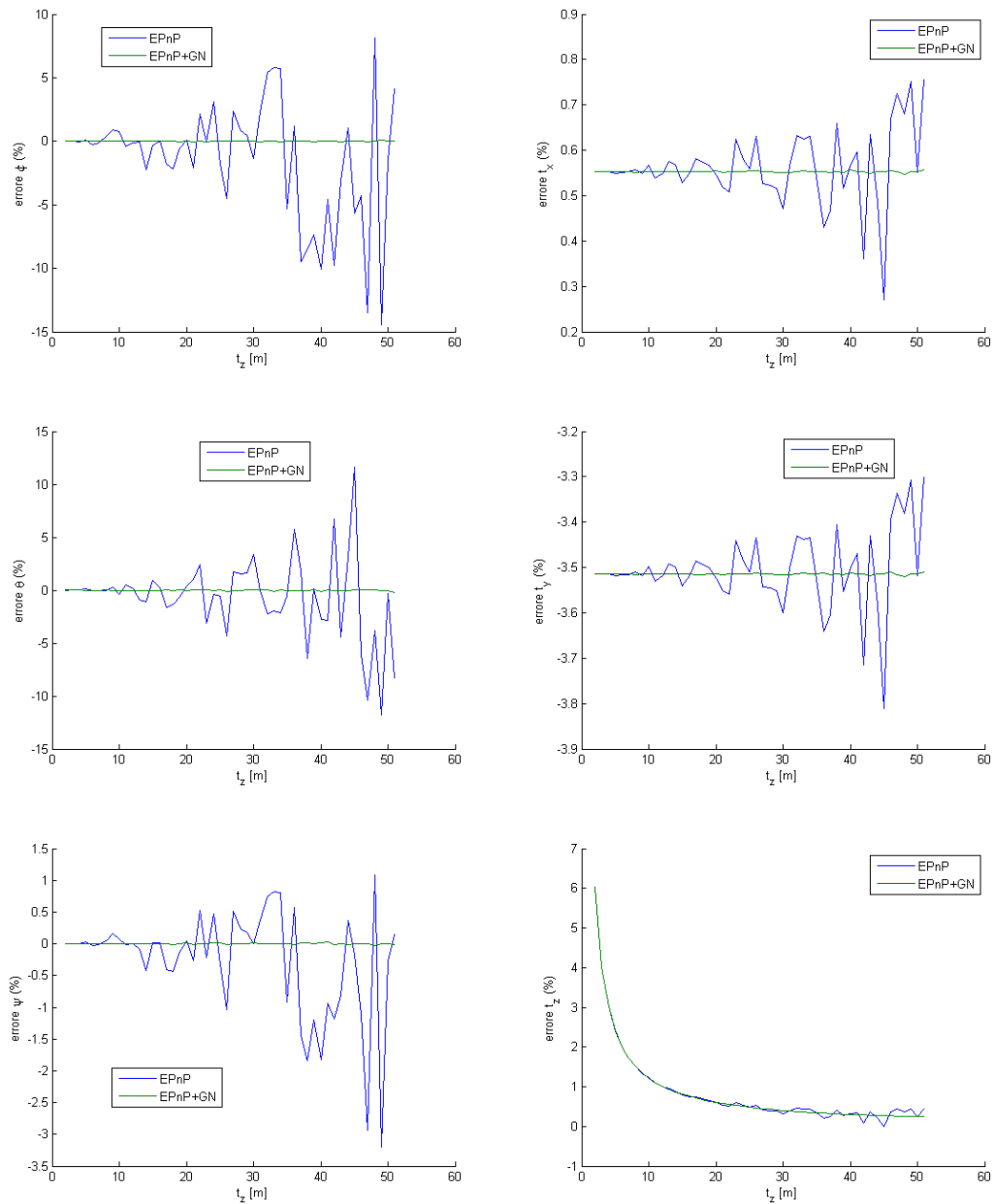


Figura 2.15: Algoritmo EP n P; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $f=50$ mm, $\sigma_{err} = 10^{-4}$ m, 5 punti caratteristici

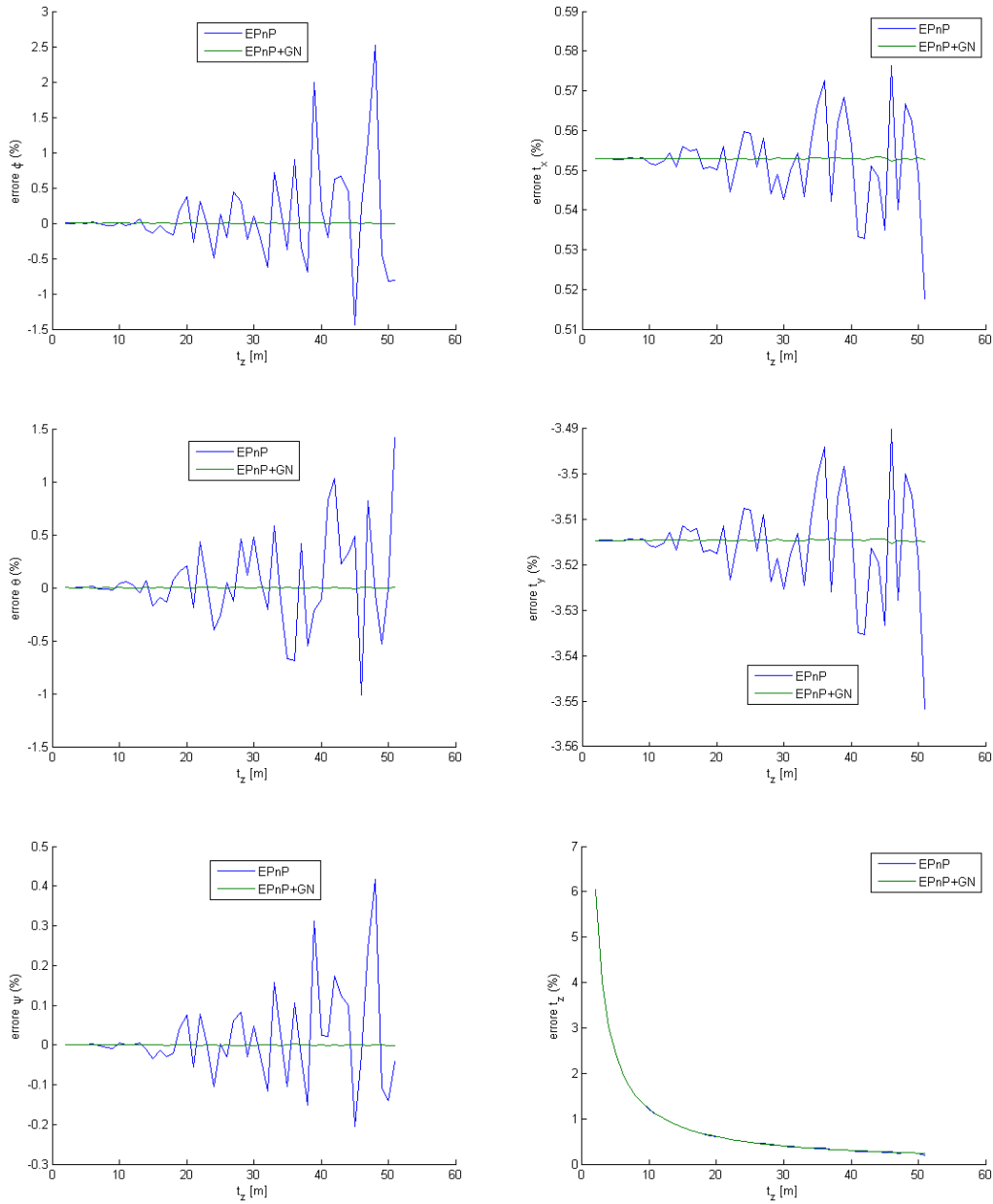


Figura 2.16: Algoritmo EPnP; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $f=50$ mm, $\sigma_{err} = 10^{-5}$ m, 5 punti caratteristici

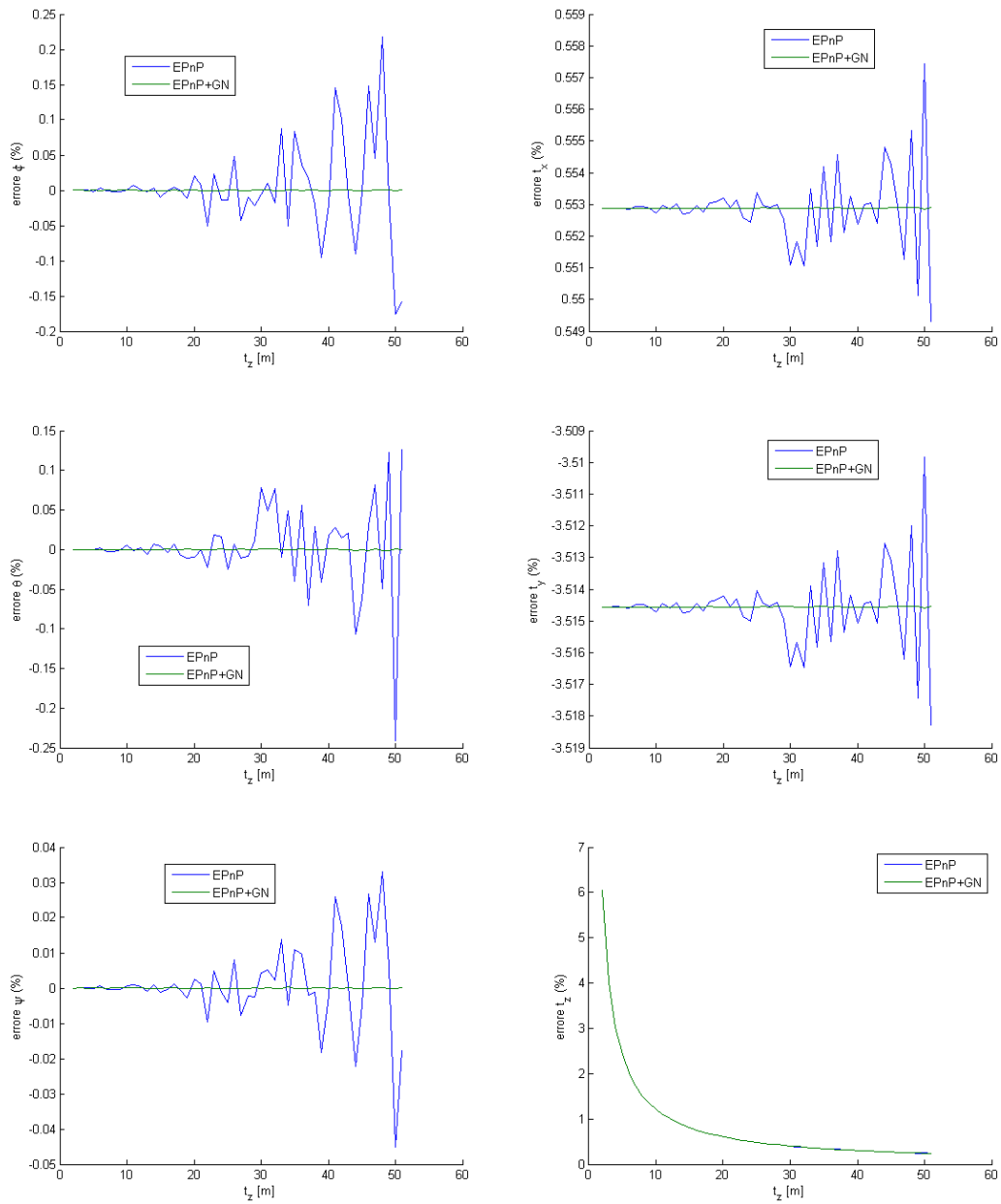


Figura 2.17: Algoritmo EP n P; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $f=50$ mm, $\sigma_{err} = 10^{-6}$ m, 5 punti caratteristici

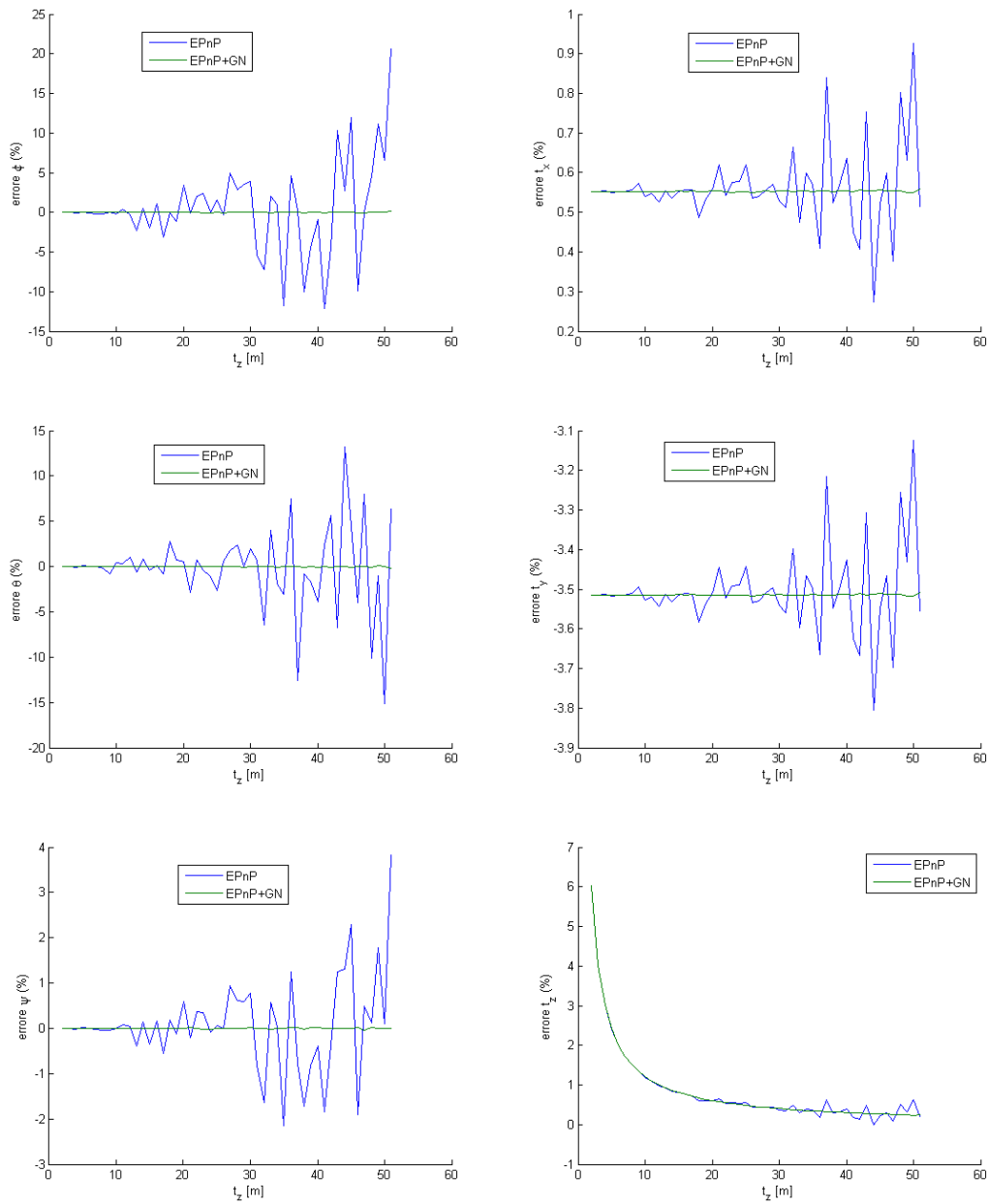


Figura 2.18: Algoritmo EPnP; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $f=100$ mm, $\sigma_{err} = 10^{-4}$ m, 5 punti caratteristici

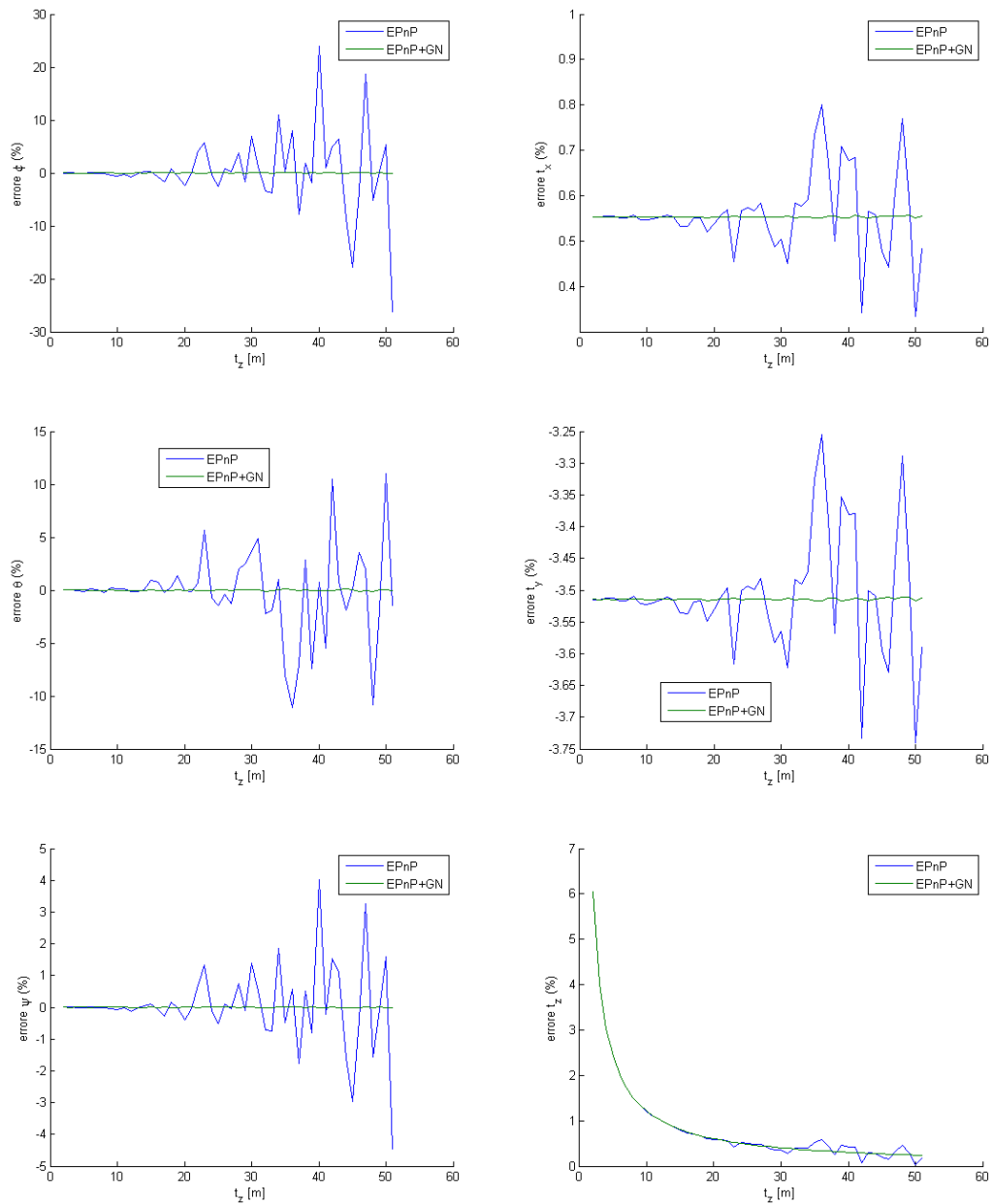


Figura 2.19: Algoritmo EP n P; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $f=5$ mm, $\sigma_{err} = 10^{-4}$ m, 5 punti caratteristici

2.5 Soluzione iterativa: algoritmo di Haralick

Sebbene, come si vedrà in seguito, questo algoritmo non sia particolarmente prestante dal punto di vista computazionale, viene presentato perché è stato il primo ad introdurre le *profondità* e rendere così il problema lineare; prima è però necessario introdurre un algoritmo di *3D-3D estimation* che viene utilizzato per ricavare la soluzione del problema.

2.5.1 3D-3D estimation

In riferimento al problema 2.2 l'algoritmo di *3D-3D estimation* permette di calcolare R e \mathbf{t} conoscendo \mathbf{q} e \mathbf{p} ; il problema viene impostato dagli autori [haralick] definendo la seguente funzione errore

$$e^2(R, \mathbf{t}) = \sum_{i=0}^n |\mathbf{s}_i - R\mathbf{p}_i - \mathbf{t}|^2 \quad (2.48)$$

costretto al vincolo che la matrice R sia ortogonale in quanto matrice di rotazione

$$RR^T = I \quad (2.49)$$

La minimizzazione rispetto \mathbf{t} porta alla soluzione del problema; si introducono

$$B = [\mathbf{b}_1 \ \mathbf{b}_2 \ \mathbf{b}_3] \quad (2.50)$$

dove

$$\mathbf{b}_k = \sum_{i=0}^n (s_i k - \bar{s}_k) (\mathbf{p}_i - \bar{\mathbf{p}}_k) \quad (2.51)$$

avendo indicato con l'indice k la componente del vettore, con l'indice i il punto cui ci si sta riferendo mentre \bar{q} e \bar{p} sono le medie dei punti nei due sistemi di riferimento

$$\bar{q} = \frac{\sum_{i=0}^n s_i}{n} \quad (2.52)$$

$$\bar{p} = \frac{\sum_{i=0}^n p_i}{n} \quad (2.53)$$

Mediante *decomposizione ai valori singolari* è infine possibile ricavare la matrice di rotazione

$$B = UDV \quad (2.54)$$

$$R = V^T U^T \quad (2.55)$$

e infine il vettore traslazione

$$\mathbf{t} = \bar{\mathbf{q}} - R\bar{\mathbf{p}} \quad (2.56)$$

Per verificare l'algoritmo è stato creato un programma che genera tre angoli di rotazione e un vettore di traslazione casuali, il problema viene risolto e si calcola

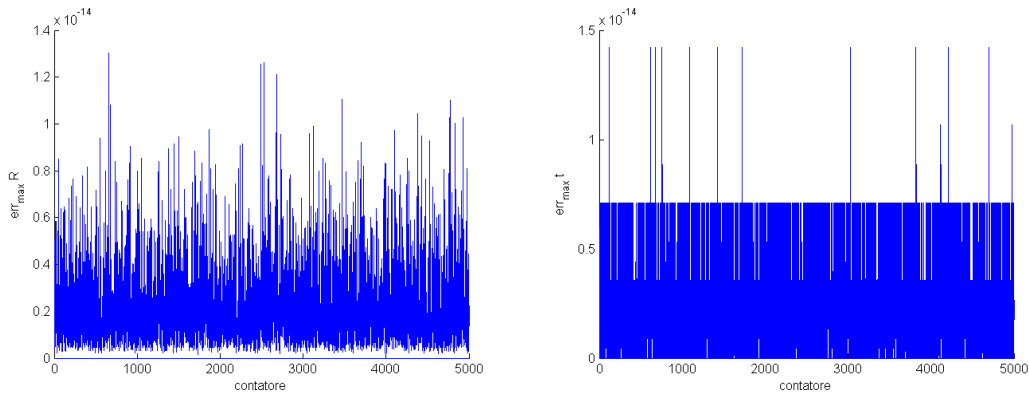


Figura 2.20: Massimo errore nel calcolo di R e \mathbf{t} con algoritmo 3D-3D pose estimation di Haralick

l'errore come differenza fra il valore vero e quello stimato; tale operazione viene eseguita qualche migliaio di volte ottenendo i grafici di figura 2.20; come si può notare la soluzione viene sempre ricavata con un errore dell'ordine dell'errore macchina.

2.5.2 Algoritmo di Haralick

Siano

$$\mathbf{u}_i = \begin{pmatrix} u_i \\ v_i \\ f \end{pmatrix} = f\mathbf{v}_i \quad (2.57)$$

i punti acquisiti dalla fotocamera; si esprimano i *punti caratteristici* nel riferimento camera come prodotto fra le *profondità* e il vettore/versore appena calcolato

$$\mathbf{q}_i = d_i\mathbf{v}_i \quad (2.58)$$

A questo punto è possibile impostare l'algoritmo in tre passi

1. ipotizzare delle *profondità* di primo tentativo; una scelta ragionevole può essere attribuire a ognuna il valore di quanto si stimi sia distante l'oggetto con i punti caratteristici
2. utilizzando l'algoritmo di 3D-3D pose estimation prima presentato ricavare la matrice R_k il vettore \mathbf{t}_k che minimizzano il seguente funzionale (k indica l'iterata)

$$e^2(R_k, \mathbf{t}_k) = \sum_{i=0}^n |R_k\mathbf{p}_i + \mathbf{t}_k - d_n\mathbf{v}_n|^2 \quad (2.59)$$

3. calcolare le *profondità* al passo successivo come segue

$$d_i^{k+1} = \frac{(R_k \mathbf{q}_i + \mathbf{t}_k)^T \mathbf{v}_i}{\mathbf{v}_i^T \mathbf{v}_i} \quad (2.60)$$

L'autore dimostra che l'algoritmo converge, il problema è che spesso tale convergenza è troppo lenta.

2.5.3 Simulazione numerica

Anche per questo algoritmo sono state analizzate due rotazioni già presentate nel capitolo 2.3.4 e due distanze relative significative

$$\mathbf{t}_1 = \begin{pmatrix} 0.5 \\ 0.4 \\ 3 \end{pmatrix} m, \quad \mathbf{t}_2 = \begin{pmatrix} 1 \\ 2 \\ 50 \end{pmatrix} m$$

La posizione dei *punti caratteristici* è la stessa già utilizzata per verificare l'algoritmo di Lepetit et al. (vedi Cap. 2.4.4). I valori di *guess* per le profondità sono stati assunti tutti uguali e pari alla norma del vettore \mathbf{t} ; è lecito pensare che in una missione reale sia più o meno conosciuto per mezzo di misure di altre sensori o per stime di missione. In ogni caso tale valore, a meno che non sia eccessivamente lontano da quello reale, non influenza particolarmente le prestazioni dell'algoritmo.

Variazione assetto e posizione relativa

Come per gli altri algoritmi si analizzano diverse situazioni in termini di distanza e assetto relativo e si valuta l'andamento della soluzione.

Osservando le figure 2.21, 2.22, 2.23 e 2.24 si nota come l'algoritmo converga e si avvicini alla soluzione reale, ma, in alcuni casi (figura 2.21 e 2.22), il numero delle iterazioni cresce di due ordini di grandezza manifestando una *velocità di convergenza* assolutamente troppo bassa; questo è uno dei motivi che ha decretato l'insuccesso dell'algoritmo.

Introduzione dei disturbi

Vengono qui introdotti i rumori di misura con varianze pari a $\sigma = 10^{-4}, 10^{-6} m$ sul primo set di rotazioni e traslazioni; i risultati sono riportati in figura 2.25 e 2.27, si nota come il numero delle iterazioni sia rimasto lo stesso e inoltre non insorgono oscillazioni; l'errore è più marcato sugli angoli che non sulle distanze e ovviamente diminuisce al diminuire del rumore.

Variazione della distanza focale

Il parametro distanza focale gioca un ruolo importante nel modello prospettico e può influenzare i risultati numerici (riferimento al Cap. 2.2.4); per tale motivo si analizza un set di rotazioni con rumore avente $\sigma = 10^{-4} m$ e si varia la distanza focale; dal confronto delle figure 2.28 e 2.29 con la figura 2.25 si nota che un aumento del fuoco non induce un miglioramento delle prestazioni, gli errori percentuali sugli angoli peggiorano mentre quelli sulle distanze migliorano leggermente ma non in maniera sostanziale, mentre un aumento della distanza focale decreta un netto peggioramento degli errori.

Variazione della posizione dei punti caratteristici

La distanza dei punti caratteristici non è particolarmente influente sui risultati se non nel caso in cui vengano presi tutti con la terza coordinata nulla, l'effetto, ben visibile in figura 2.30 da confrontare con la figura 2.27, è una riduzione dell'accuratezza principalmente nella determinazione degli angoli.

2.5.4 Conclusioni sull'algoritmo

E' sicuramente uno degli algoritmi storicamente più importanti grazie al nuovo approccio che sfrutta; d'altro canto è una soluzione incompleta che mostra pregi ma ancora delle grosse limitazioni. In particolare la convergenza è assicurata ma risulta essere troppo lenta, la robustezza è invece buona e può sicuramente essere migliorata introducendo un filtro di Kalman. Un ultimo punto non molto pratico è il doverlo inizializzare con le profondità, non sempre facilmente stimabili.

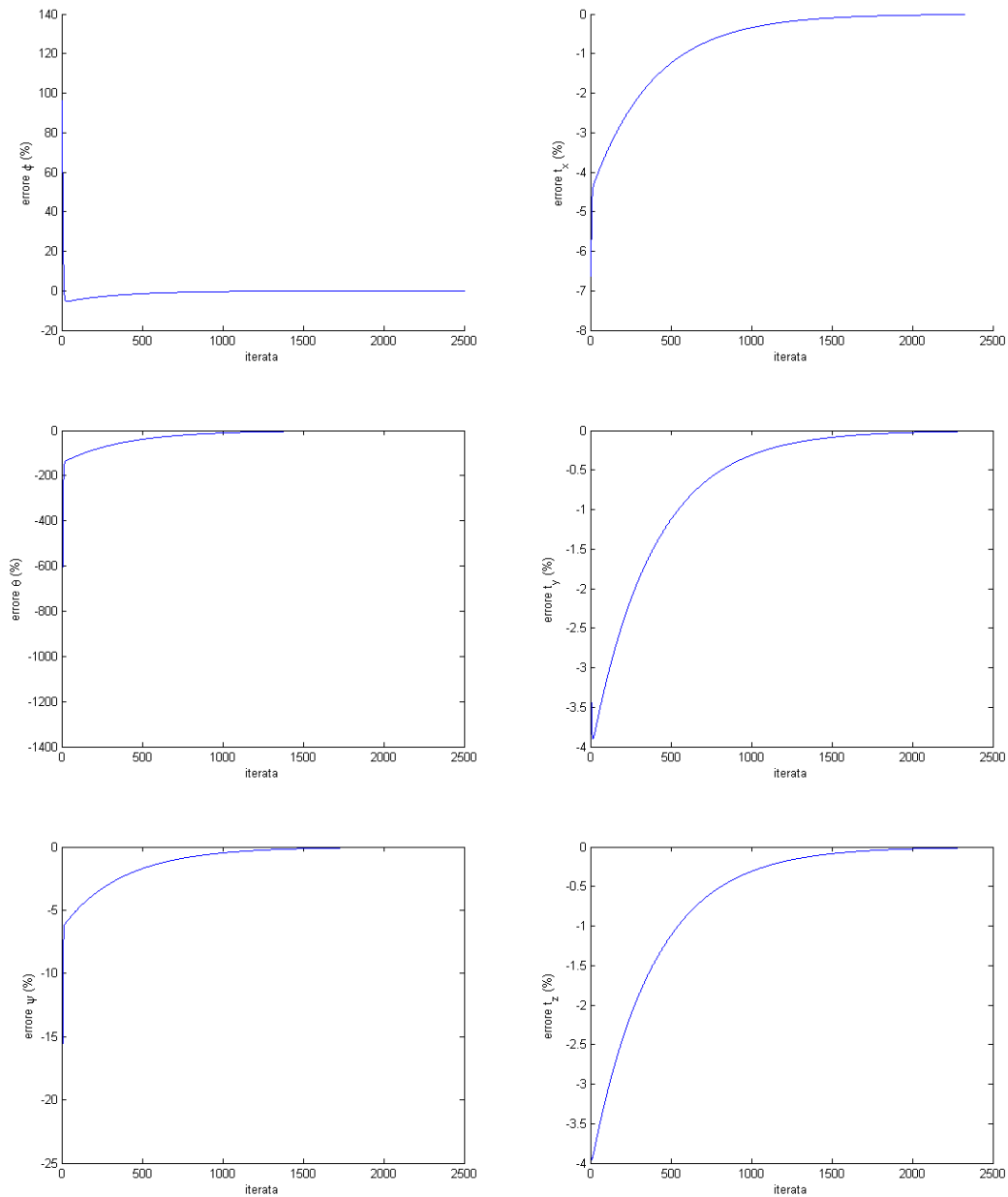


Figura 2.21: Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, $f=50$ mm, $\sigma_{err} = 0$ m

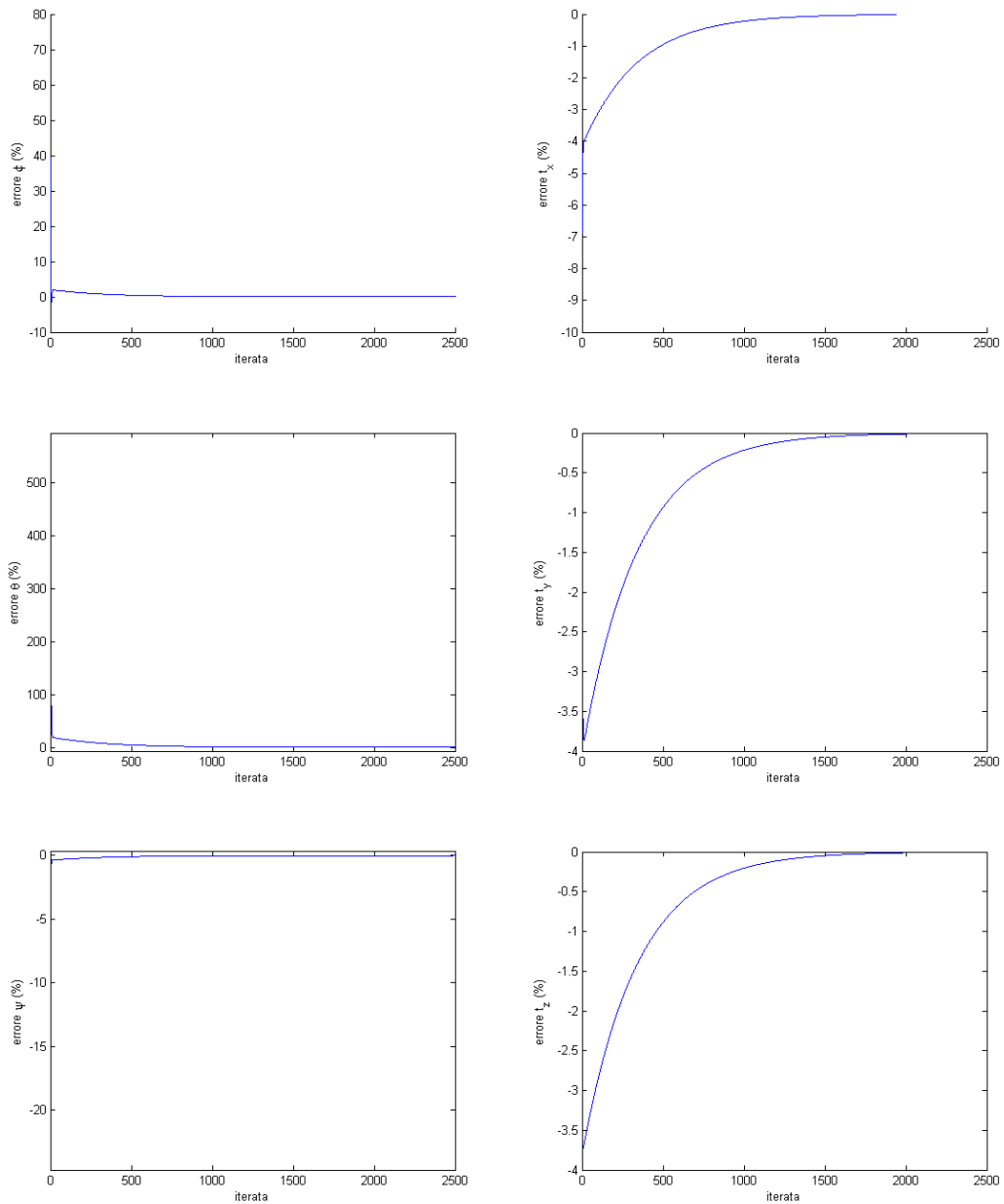


Figura 2.22: Algoritmo di Haralick; valori reali $\phi = -30^\circ$, $\theta = 6^\circ$, $\psi = 35^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, $f=50$ mm, $\sigma_{err} = 0$ m

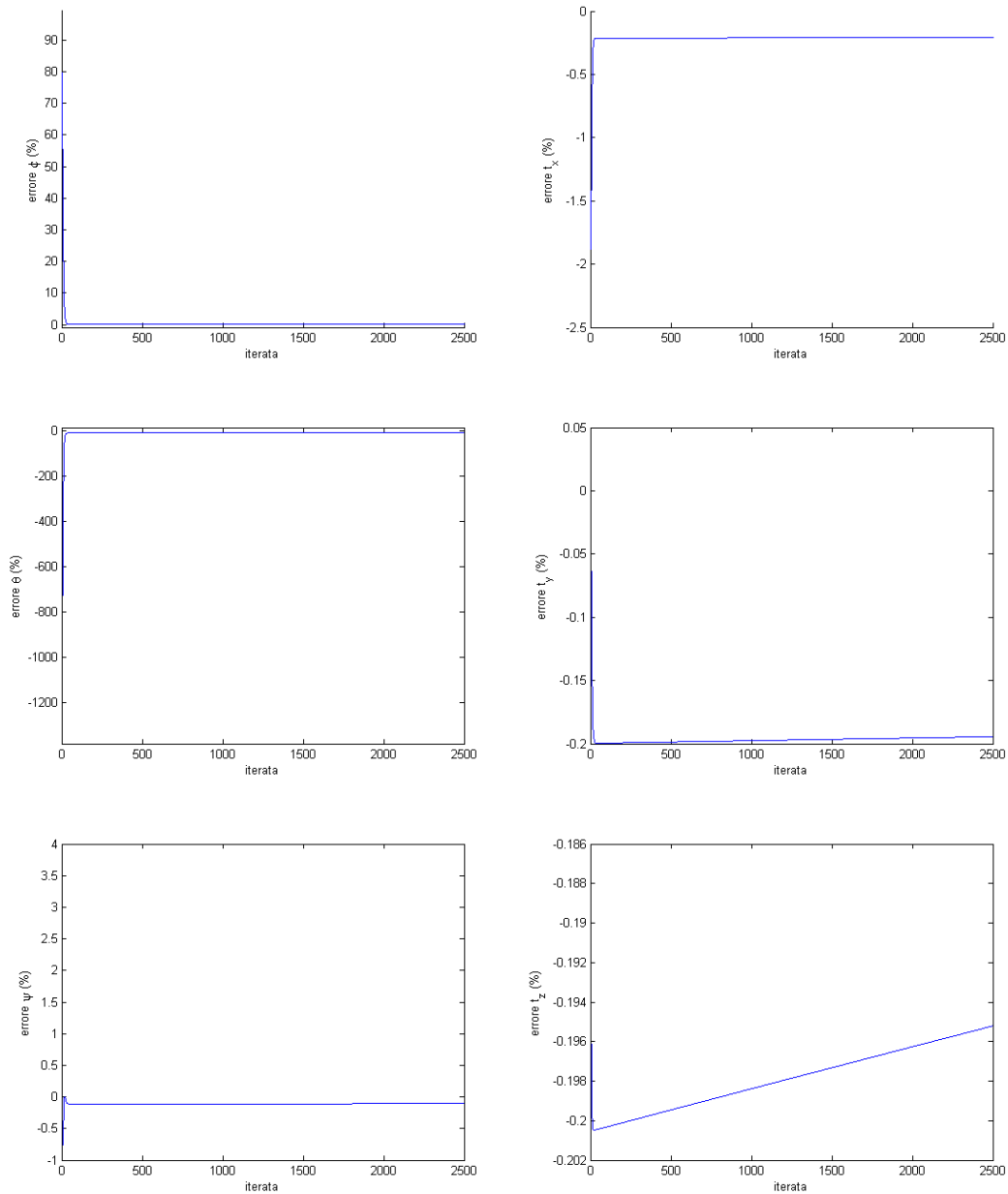


Figura 2.23: Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [1 \ 2 \ 50]^T$ m, $f=50$ mm, $\sigma_{err} = 0$ m

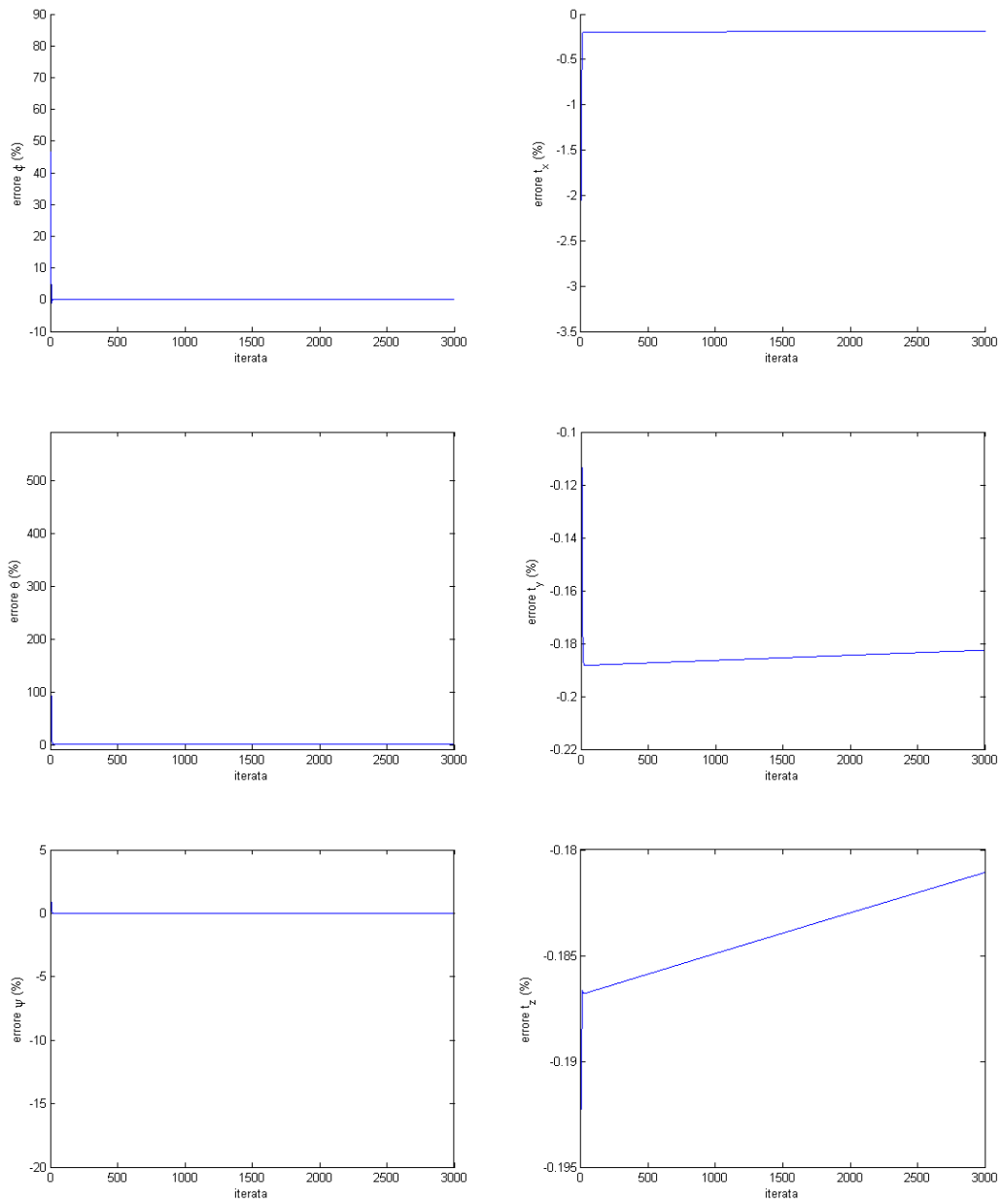


Figura 2.24: Algoritmo di Haralick; valori reali $\phi = -30^\circ$, $\theta = 6^\circ$, $\psi = 35^\circ$, $\mathbf{t} = [1 \ 2 \ 50]^T$ m, $f=50$ mm, $\sigma_{err} = 0$ m

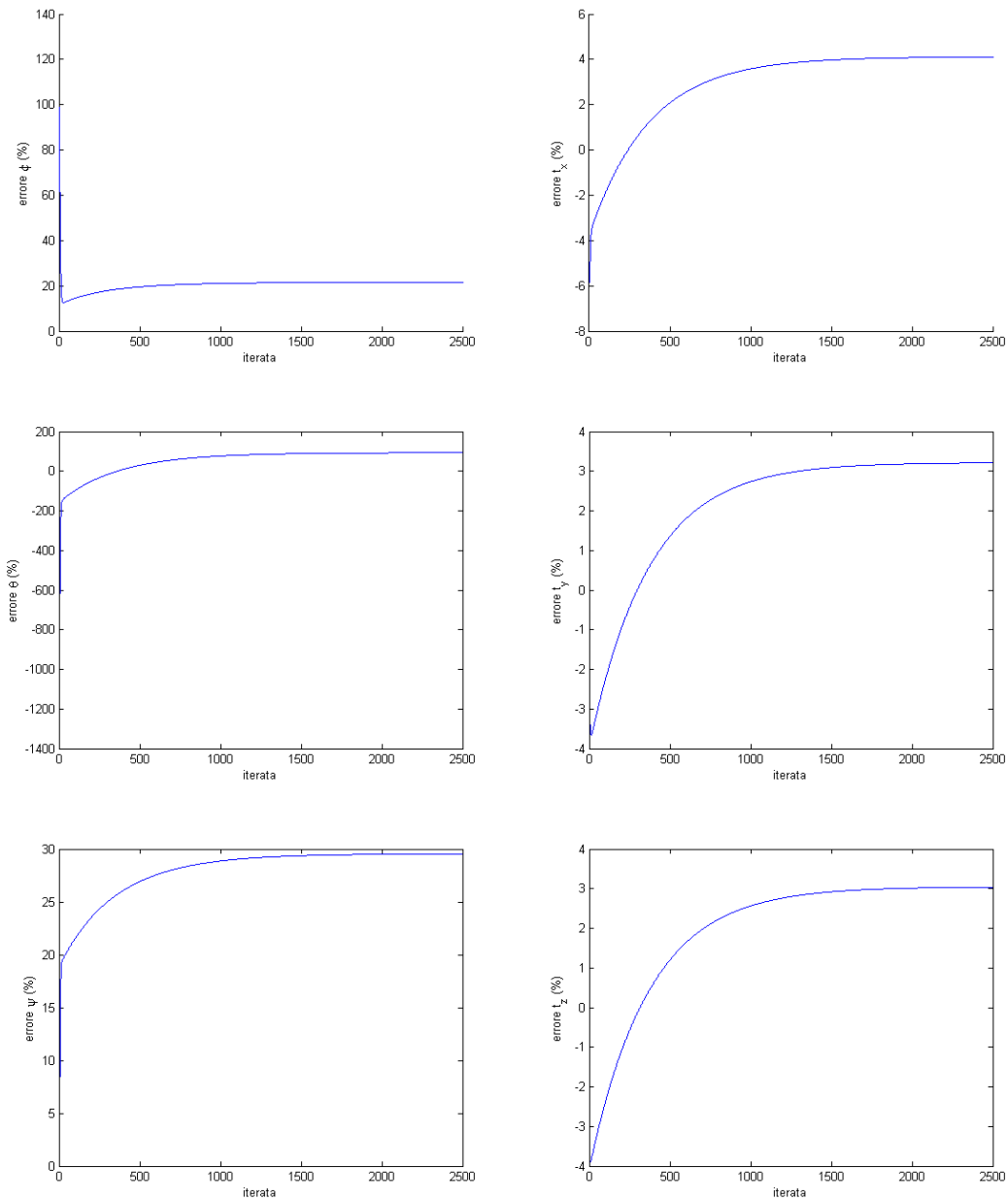


Figura 2.25: Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 4, 0, 5, 3]^T$ m, $f=50$ mm, $\sigma_{err} = 10^{-4}$ m

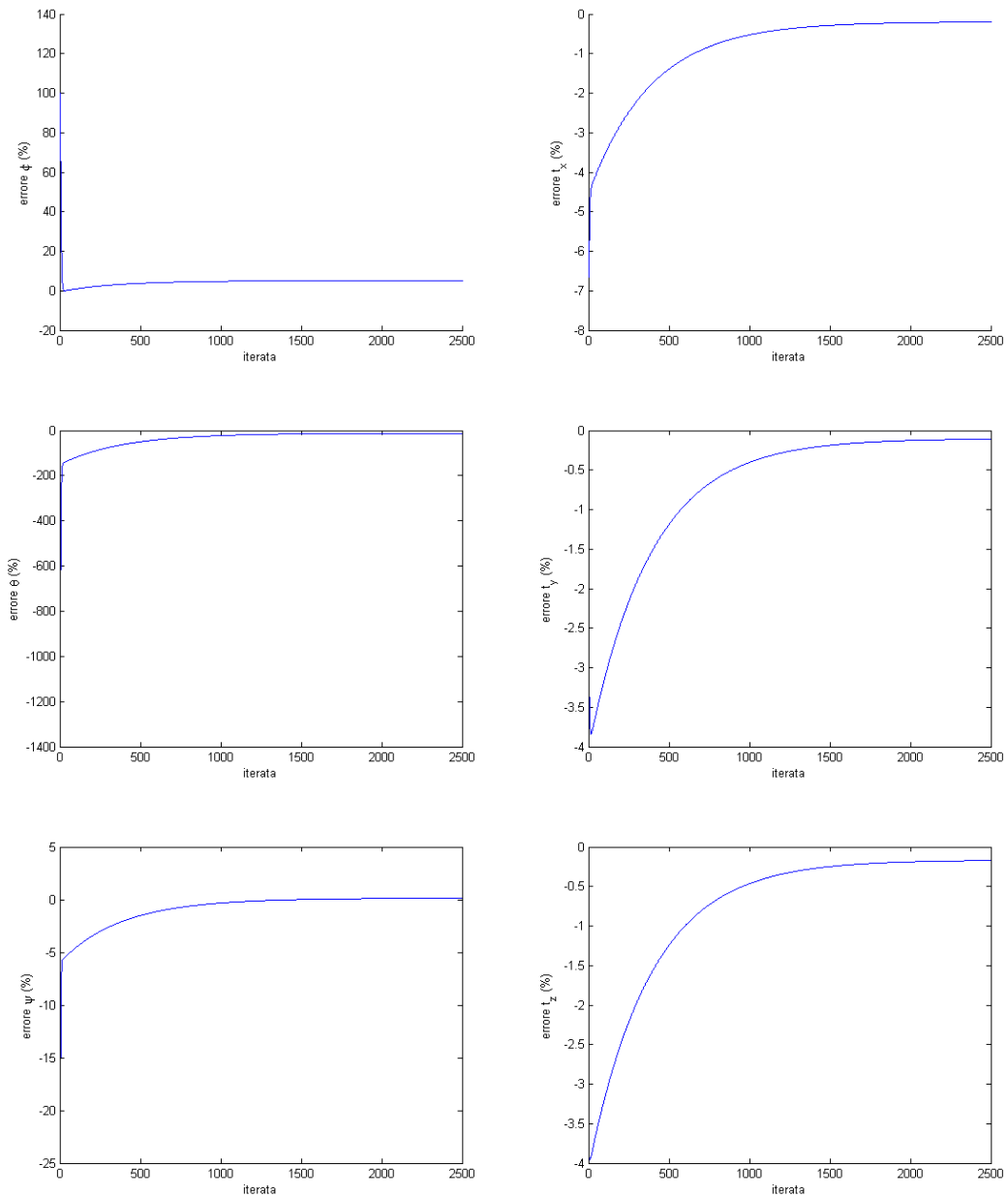


Figura 2.26: Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 4, 0, 5, 3]^T$ m, $f=50$ mm, $\sigma_{err} = 10^{-5}$ m

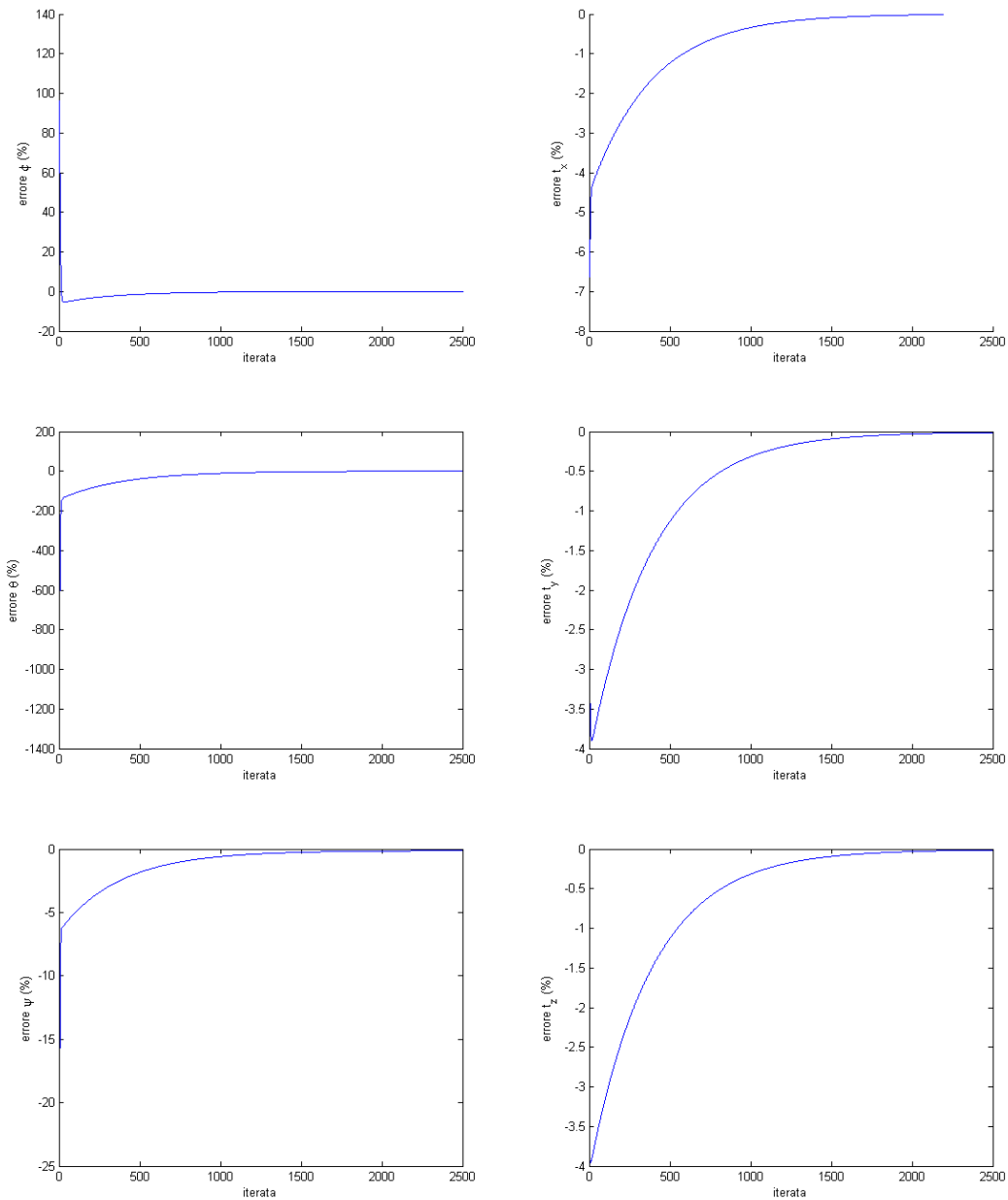


Figura 2.27: Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 4 \ 0, 5 \ 3]^T$ m, $f=50$ mm, $\sigma_{err} = 10^{-6}$ m

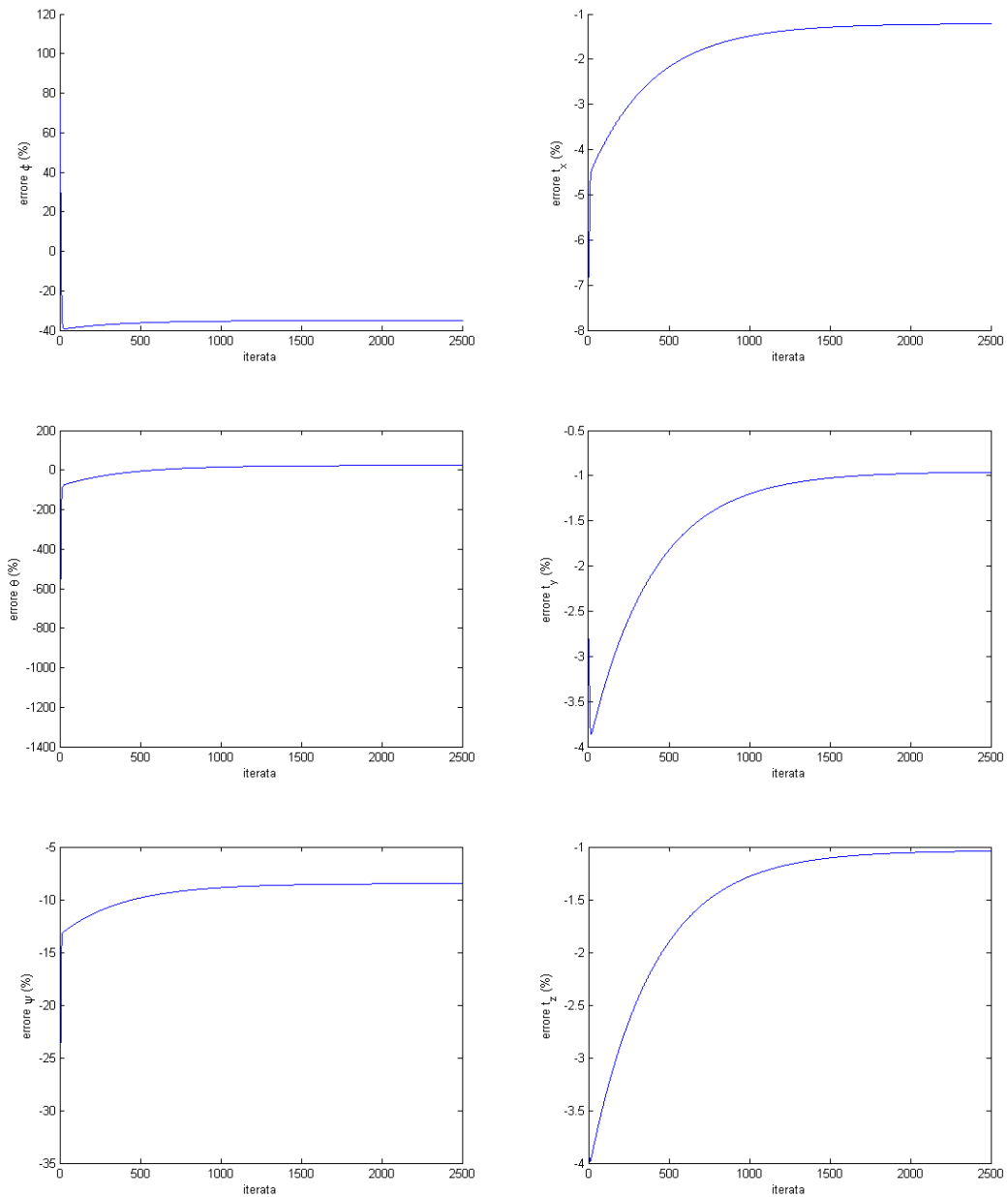


Figura 2.28: Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, $f=100$ mm, $\sigma_{err} = 10^{-4}$ m

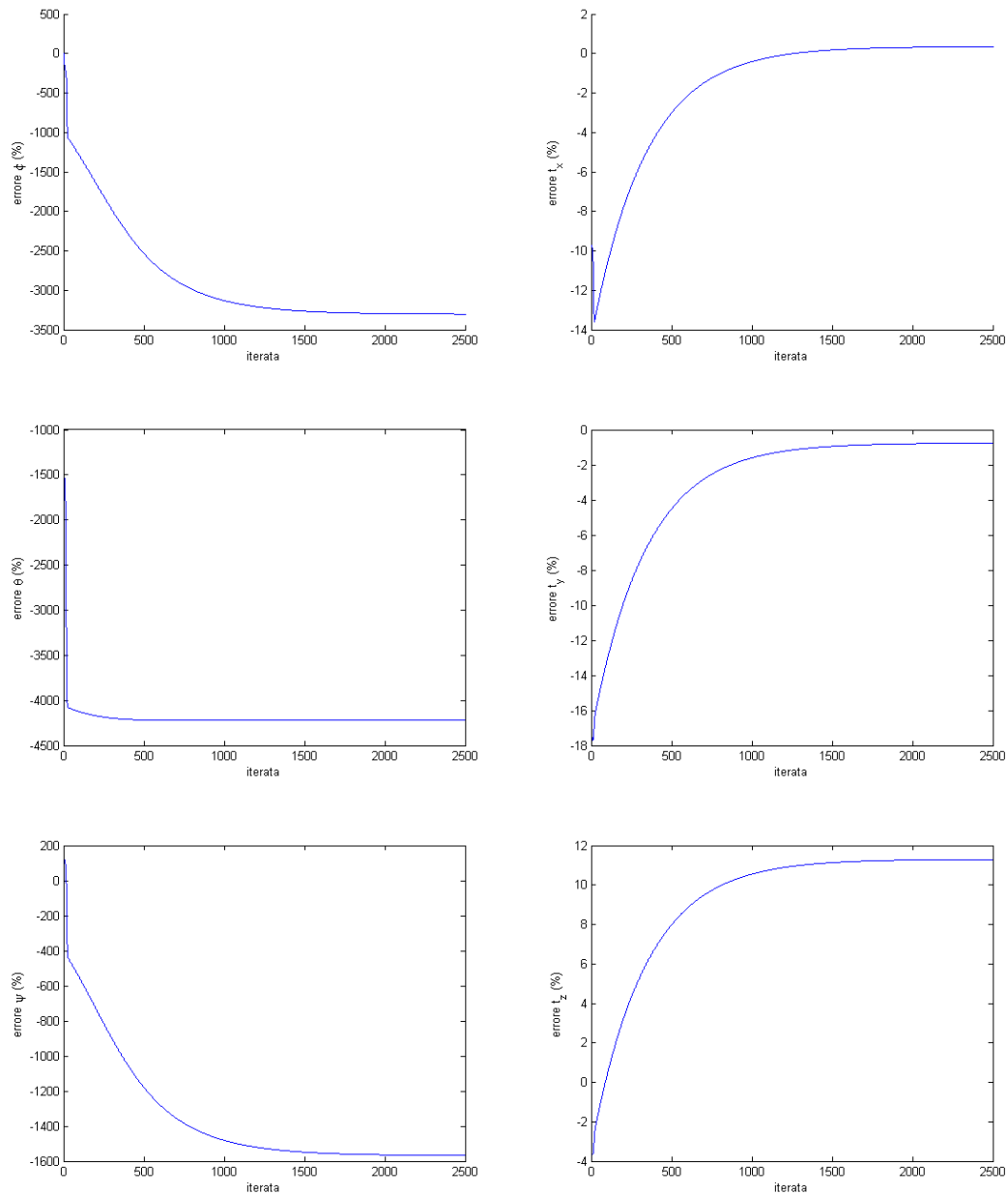


Figura 2.29: Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5, 0, 4, 3]^T$ m, $f=5$ mm, $\sigma_{err} = 10^{-4}$ m

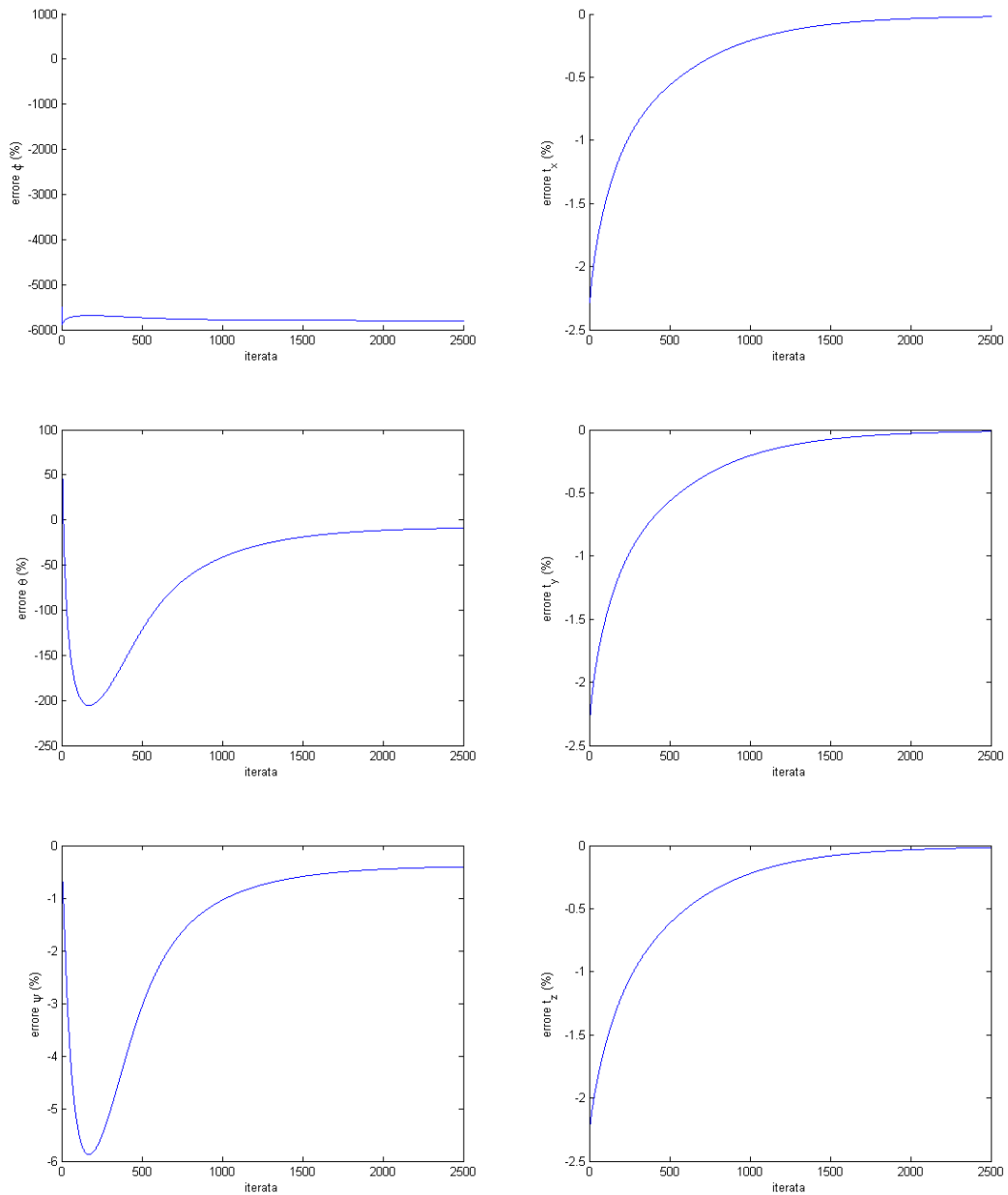


Figura 2.30: Algoritmo di Haralick; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 4 \ 0, 5 \ 3]^T$ m, $f=50$ mm, $\sigma_{err} = 10^{-6}$ m, punti caratteristici nel piano $z=0$ m

2.6 Soluzione iterativa: algoritmo a 2 stadi

L'algoritmo che verrà presentato nel seguente paragrafo è stato sviluppato da Zhang ShiJie, Cao XiBin, Zhang Fan e He Liang nel 2009 [china1]; il lavoro è un miglioramento dell'algoritmo di Haralick.

2.6.1 Equazione di collinearità

Come già illustrato nel modello *pinhole*, per ogni *punto caratteristico* è possibile tracciare una linea che attraversa il piano immagine (e ne definisce le coordinate della proiezione) e passa infine per il centro del sistema di riferimento solidale con la fotocamera, sostanzialmente i punti sono *collineari*; è quindi possibile calcolare i seguenti versori (chiamati dall'autore raggi inversi di proiezione)

$$\mathbf{V}_i = \frac{1}{\sqrt{X_i^2 + Y_i^2 + f^2}} \begin{pmatrix} X_i \\ Y_i \\ f \end{pmatrix} \quad (2.61)$$

avendo indicato con X_i e Y_i le proiezioni dei punti sul piano immagine. E' quindi possibile definire le coordinate dei *punti caratteristici* nel riferimento camera introducendo lo scalare profondità (deep in inglese)

$$\mathbf{s}_i = d_i \mathbf{V}_i \quad (2.62)$$

Come condizione di collinearità, la proiezione ortogonale dei *punti caratteristici* \mathbf{q}_i sul raggio di proiezione inverso deve coincidere con i *punti caratteristici* \mathbf{q}_i stessi; si ricava quindi l'equazione di collinearità

$$\mathbf{q}_i = \mathbf{V}_i^T \mathbf{s}_i \mathbf{V}_i = \mathbf{V}_i^T (R\mathbf{s}_i + \mathbf{t}) \mathbf{V}_i \quad (2.63)$$

2.6.2 Calcolo della distanza ottima

Si definisce la funzione errore, già introdotta precedentemente, alla luce di quanto appena visto come

$$e^2(M, \mathbf{t}, d_i) = \sum_{i=0}^n |d_i \mathbf{V}_i - R\mathbf{s}_i - \mathbf{t}|^2, RR^T = I \quad (2.64)$$

avendo anche evidenziato il vincolo di ortogonalità cui è sottoposta la funzione errore. L'espressione 2.65 può essere ulteriormente riscritta come,

$$e^2(M, \mathbf{t}, d_i) = \sum_{i=0}^n \left| (R\mathbf{s}_i + \mathbf{t}) - \mathbf{V}_i^T (R\mathbf{s}_i + \mathbf{t}) \mathbf{V}_i \right|^2, RR^T = I \quad (2.65)$$

Assumendo nota la matrice di rotazione R allora la funzione errore sarà quadratica in t , il valore ottimo sarà dato dalla soluzione della seguente equazione,

$$\frac{\partial e^2(M, \mathbf{t})}{\partial \mathbf{t}} = 0 \quad (2.66)$$

Dalla 2.66, si ottiene, [**china1**]

$$\mathbf{t}(R) = \left[nI - \sum_{i=0}^n (\mathbf{V}_i \mathbf{V}_i^T) \right]^{-1} \sum_{i=0}^n (\mathbf{V}_i \mathbf{V}_i^T - I) R \mathbf{s}_i \quad (2.67)$$

Infine, è possibile determinare la profondità in funzione della matrice di rotazione, R , e del vettore di traslazione ottimo, $t(R)$, come segue,

$$d_i(R) = \mathbf{V}_i^T (R \mathbf{s}_i + \mathbf{t}(R)) \quad (2.68)$$

2.6.3 Flow-chart delle operazioni da eseguire

In definitiva, l'algoritmo prevede i seguenti step:

1. inizializzare la matrice di rotazione; in applicazioni dinamiche, se disponibile, si può utilizzare la matrice calcolata al passo precedente
2. calcolare i versori dei raggi di proiezione
3. calcolare il vettore di traslazione ottimo con 2.67
4. dalla 2.68 aggiornare le profondità
5. ricostruire le coordinate dei punti caratteristici nel riferimento fotocamera

$$\mathbf{s}_i = d_i \mathbf{V}_i$$

6. aggiornare la matrice di rotazione utilizzando l'algoritmo di Umeyama
7. calcolare l'errore

$$e^2(M, \mathbf{t}, d_i) = \sum_{i=0}^n |d_i \mathbf{V}_i - R \mathbf{s}_i - \mathbf{t}|^2$$

8. se l'errore è maggiore di un errore stabilito e non si è raggiunto il numero massimo di iterazioni tornare al punto 3. In ogni altro caso l'algoritmo termina

2.6.4 Algoritmo di Umeyama

E' un algoritmo di *3D-3D pose estimation* presentato da Shinji Umeyama nel 1989 [umeyama] e costituisce il perfezionamento dei risultati precedentemente ottenuti da altri autori, il cui algoritmo però in caso di dati fortemente corrotti poteva presentare delle riflessioni.

L'algoritmo si basa sulla decomposizione ai valori singolari della matrice delle varianze dei dati e può essere riassunto come segue

1. si calcolano i valori statistici

$$\boldsymbol{\mu}_{p_i} = \frac{1}{N} \sum_{i=1}^N \mathbf{p}_i \quad (2.69a)$$

$$\boldsymbol{\mu}_{s_i} = \frac{1}{N} \sum_{i=1}^N \mathbf{s}_i \quad (2.69b)$$

$$\boldsymbol{\sigma}_{p_i}^2 = \frac{1}{N} \sum_{i=1}^N |\mathbf{p}_i - \boldsymbol{\mu}_{p_i}|^2 \quad (2.69c)$$

$$H = \frac{1}{N} \sum_{i=1}^N (\mathbf{p}_i - \boldsymbol{\mu}_{p_i}) (\mathbf{s}_i - \boldsymbol{\mu}_{s_i})^T \quad (2.69d)$$

2. si calcola la *decomposizione ai valori singolari* della matrice H,

$$(U, W, V) = \text{svd}(H) \quad (2.70)$$

le matrici U e V sono ortogonali di dimensioni 3×3 , mentre W è diagonale e contiene gli autovalori della matrice H.

3. se il *rango*(H)>2 si definisce la matrice S come

$$S = \begin{cases} I & \text{se } \det(H) \geq 0 \\ \text{diag}(1, 1, -1) & \text{se } \det(H) < 0 \end{cases} \quad (2.71)$$

se il *rango*(H)=2 si definisce la matrice S come

$$S = \begin{cases} I & \text{se } \det(U)\det(V) = 1 \\ \text{diag}(1, 1, -1) & \text{se } \det(U)\det(V) = -1 \end{cases} \quad (2.72)$$

avendo indicato con *diag*(·) una matrice quadrata avente sulla diagonale principale i valori indicati fra parentesi, e con *det*(·) il determinante della matrice specificata tra parentesi.

4. si calcolano la matrice di rotazione e il vettore di traslazione,

$$R = USV^T \quad (2.73a)$$

$$c = \frac{1}{\boldsymbol{\sigma}_{p_i}^2} = \text{tr}(WS) \quad (2.73b)$$

$$\mathbf{t} = \boldsymbol{\mu}_{q_i} - cR\boldsymbol{\mu}_{p_i} \quad (2.73c)$$

avendo indicato con $tr(\cdot)$ l'operatore che esegue la somma degli elementi sulla diagonale principale della matrice indicata fra parentesi.

Il parametro c indica il fattore di scala e per le nostre applicazioni risulta sempre unitario.

Per verificare l'efficacia dell'algoritmo è stata realizzata una simulazione in cui vengono generate casualmente sia le rotazioni (comprese fra 0 e 360°) sia il vettore \mathbf{t} (compreso tra 0 e 100 metri) e si calcola l'errore come differenza fra il valore vero e quello stimato. I risultati sono riportati in figura 2.31, si nota come l'errore sia dell'ordine dell'errore macchina.

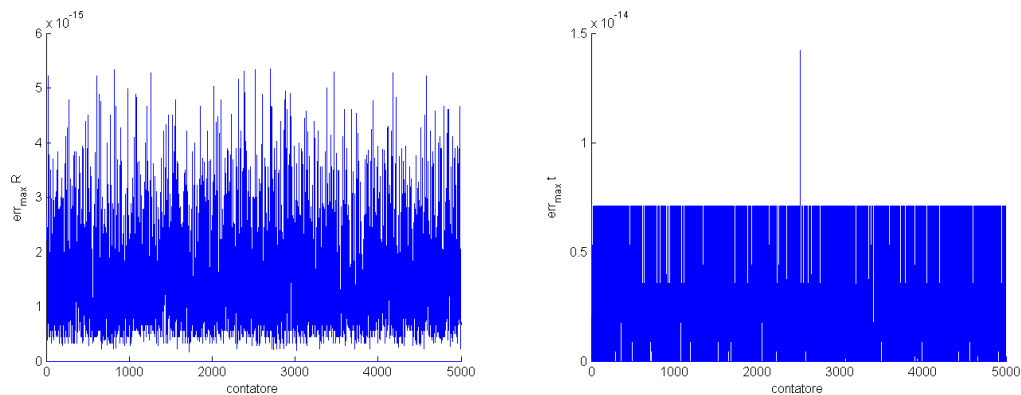


Figura 2.31: Massimo errore nel calcolo di R e \mathbf{t} con l'algoritmo di Umeyama

2.6.5 Simulazione numerica

Le simulazioni sono state effettuate considerando gli stessi valori utilizzati per l'algoritmo di Haralick (Cap. 2.5.3). In tutte le simulazioni la matrice d'assetto iniziale è pari alla matrice *identità*, nei casi reali sarà la matrice d'assetto determinata al passo temporale precedente.

Variazione assetto e distanza relativa

Questa serie di simulazioni prevede la variazione dell'assetto e della distanza relativa senza considerare i disturbi.

I risultati sono illustrati in figura 2.32, 2.33, 2.34, 2.35, ?? e ??; si osserva che la soluzione converge in meno di venti iterazioni.

Introduzione disturbi

Vengono qui introdotti i rumori di misura ($\sigma = 10^{-4}, 10^{-5}, 10^{-6} m$) sul primo set di rotazioni e traslazioni; i risultati sono illustrati in figura 2.36, 2.37 e 2.38, si

nota come il numero delle iterazioni sia rimasto lo stesso e inoltre non insorgono oscillazioni, proprio come in Haralick; l'errore è più marcato sugli angoli che non sulle distanze e ovviamente diminuisce al diminuire del rumore.

Variazione distanza focale

Come per gli altri algoritmi viene studiata l'influenza della distanza focale sull'errore dei risultati; dal confronto delle figure 2.39 e 2.40 con la figura 2.36 si nota che all'aumentare della distanza focale segue una riduzione dell'errore, di conseguenza si conferma che una diminuzione della distanza focale induce un aumento dell'errore dei risultati.

Variazione distanza punti caratteristici

La distanza dei punti caratteristici non è particolarmente influente sui risultati se non nel caso in cui vengano presi tutti con la terza coordinata nulla, l'effetto, ben visibile in figura 2.41, è una riduzione della precisione e un aumento del numero di iterazioni necessarie.

2.6.6 Conclusioni sull'algoritmo

E' probabilmente l'algoritmo iterativo più preciso fra quelli proposti, è sufficientemente robusto e converge in un numero limitato di iterazioni; la riduzione del rumore può venire migliorata utilizzando un filtro di Kalman, mentre il numero di iterazioni potrebbe decrescere utilizzando come matrice d'assetto di guess quella calcolata al passo precedente (in un applicazione reale), o, alternativamente, utilizzare un algoritmo non iterativo per calcolarne una.

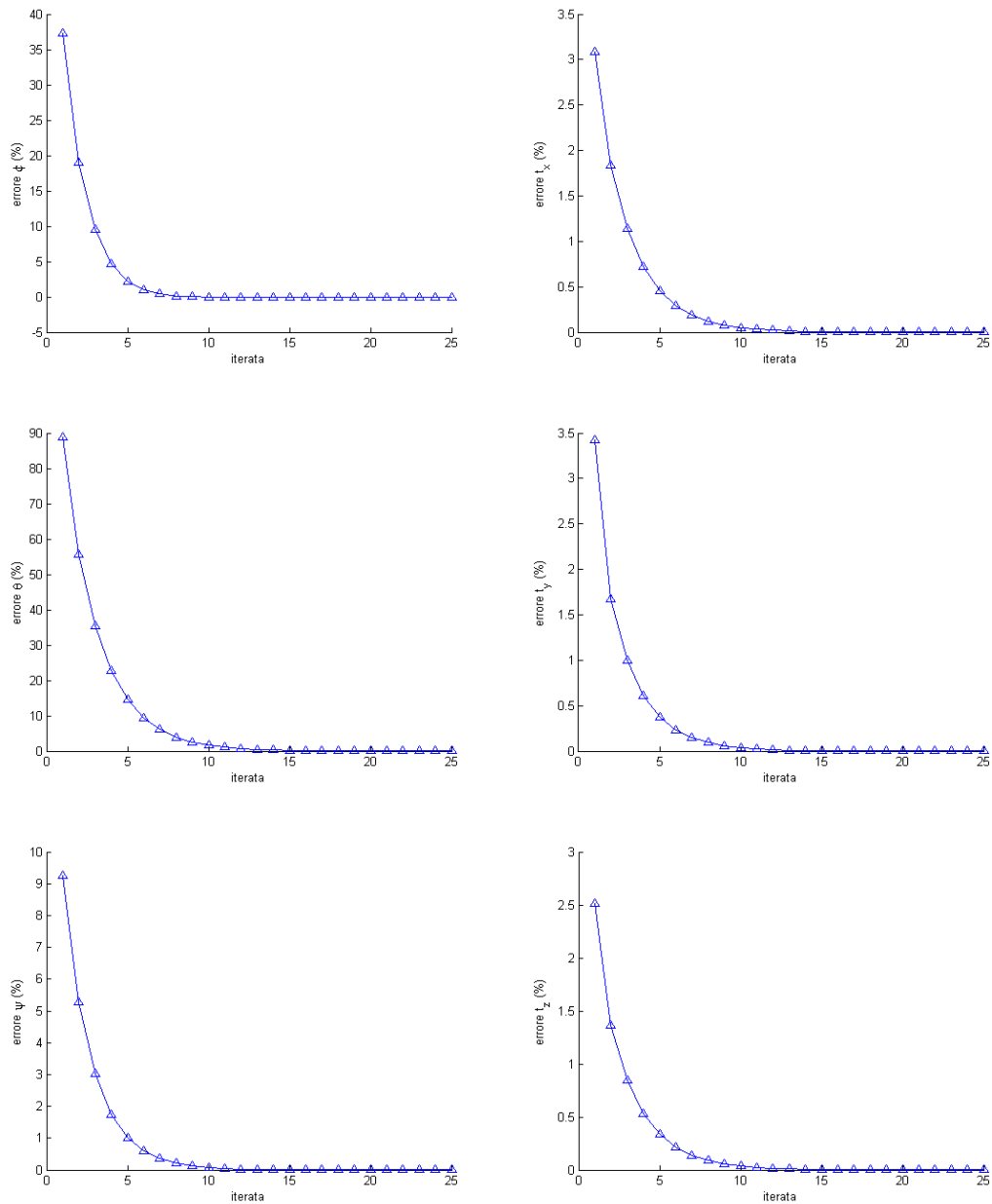


Figura 2.32: Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, $f=50$ mm, $\sigma_{err}=0$ m

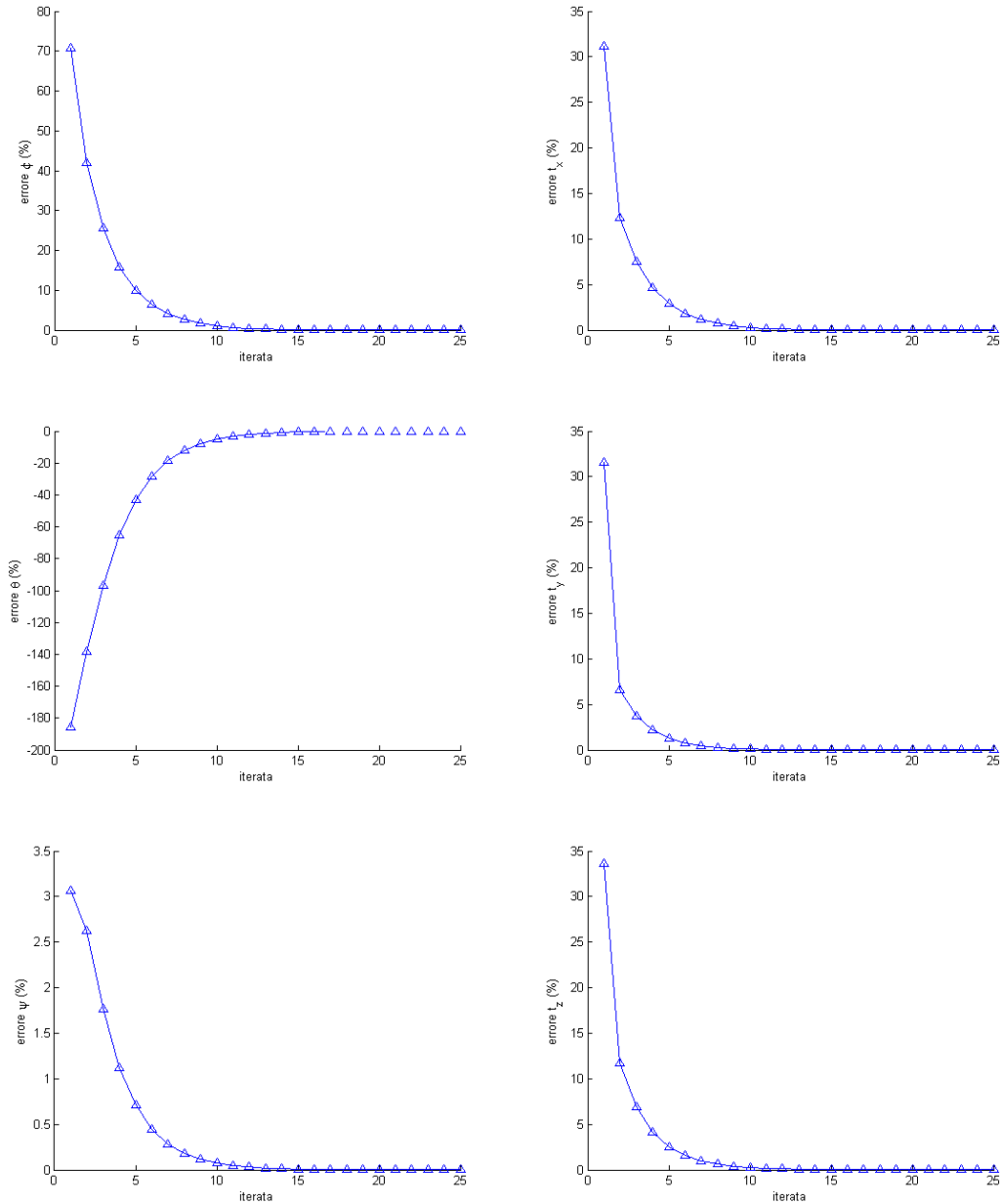


Figura 2.33: Algoritmo 2-stage; valori reali $\phi = -30^\circ$, $\theta = 6^\circ$, $\psi = 35^\circ$, $\mathbf{t} = [0, 5, 0, 4, 3]^T$ m, $f=50$ mm, $\sigma_{err}=0$ m

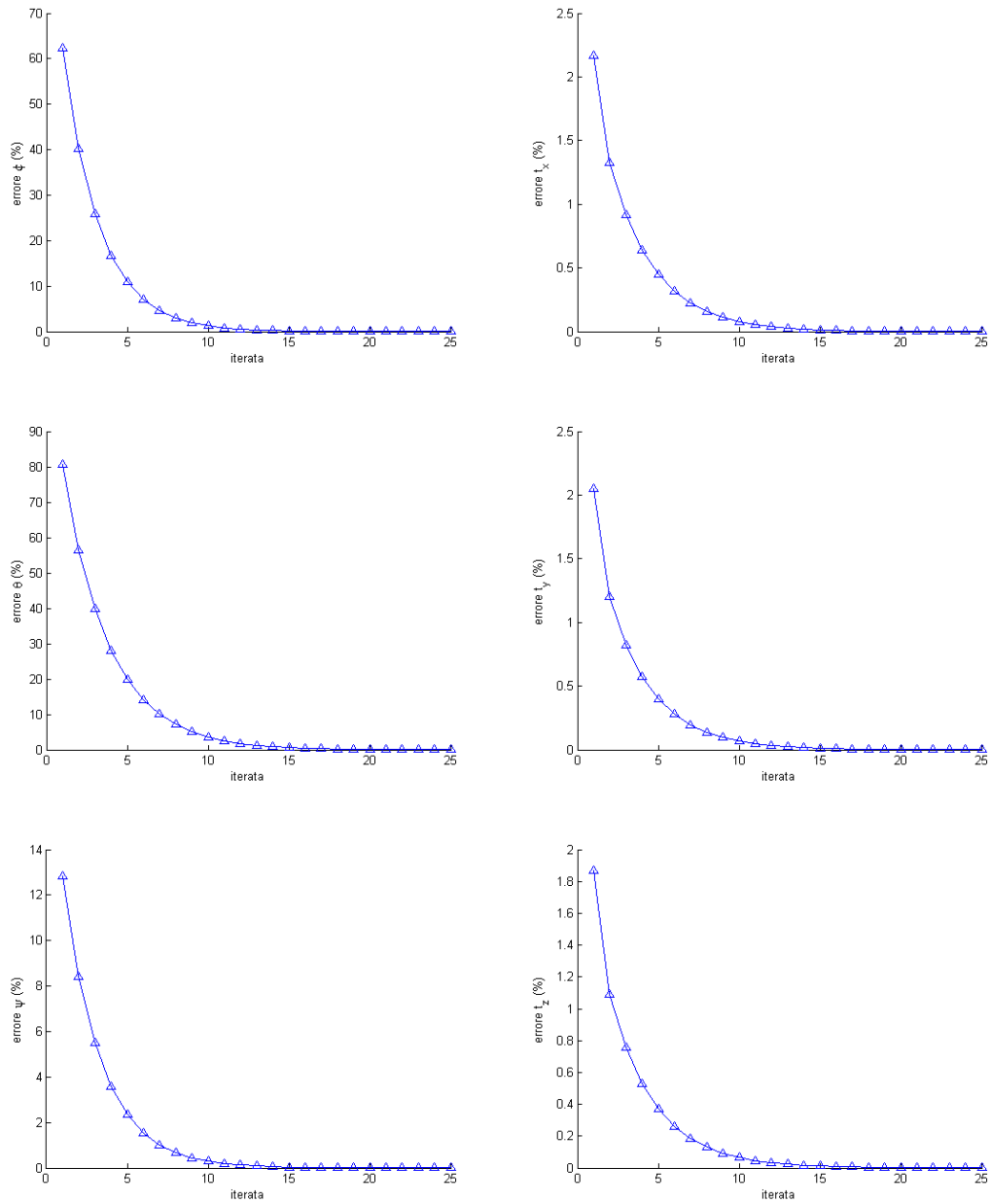


Figura 2.34: Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [1 \ 2 \ 50]^T$ m, $f=50$ mm, $\sigma_{err}=0$ m

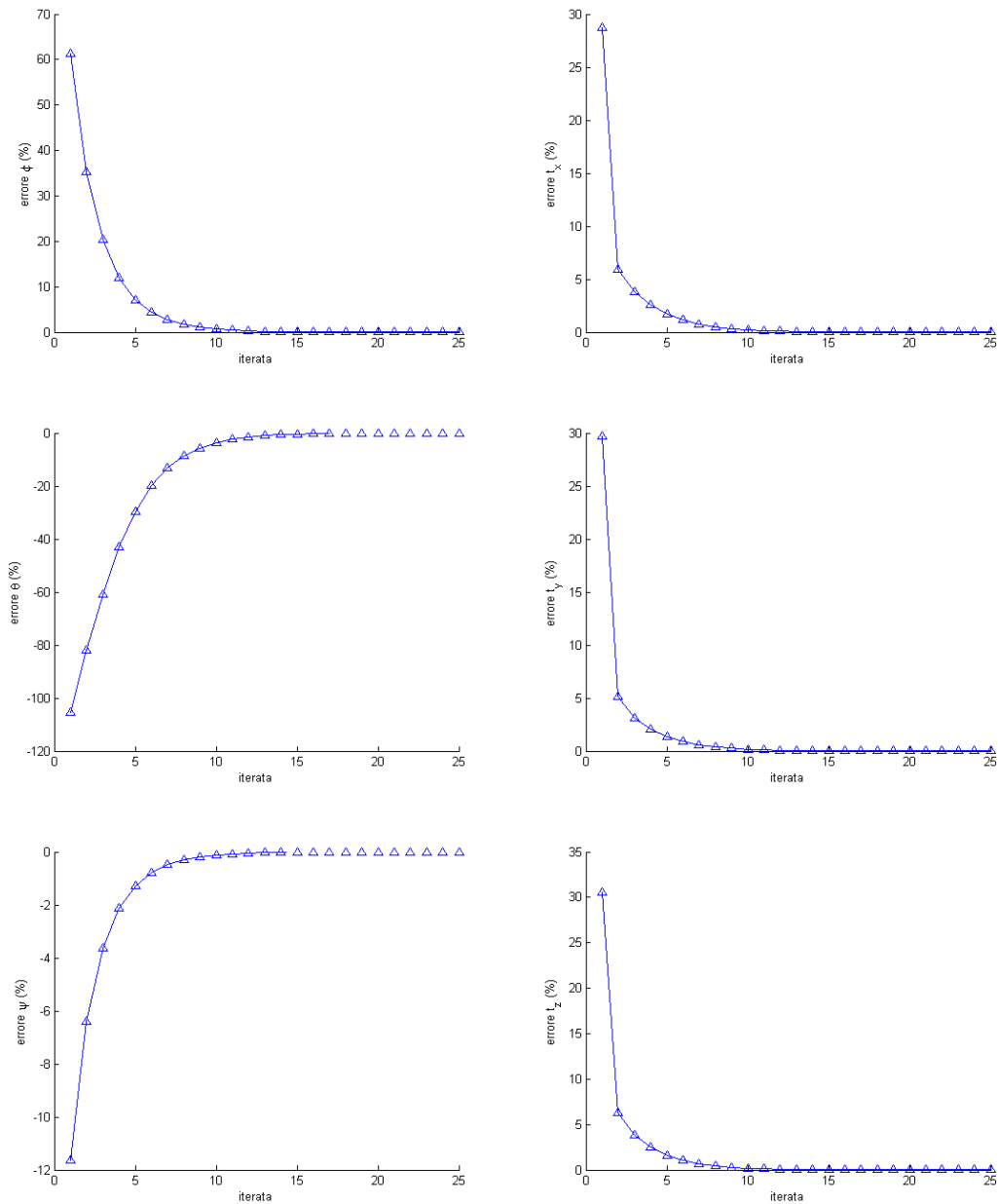


Figura 2.35: Algoritmo 2-stage; valori reali $\phi = 30^\circ$, $\theta = -6^\circ$, $\psi = 35^\circ$, $\mathbf{t} = [1 \ 2 \ 50]^T$ m, $f=50$ mm, $\sigma_{err}=0$ m

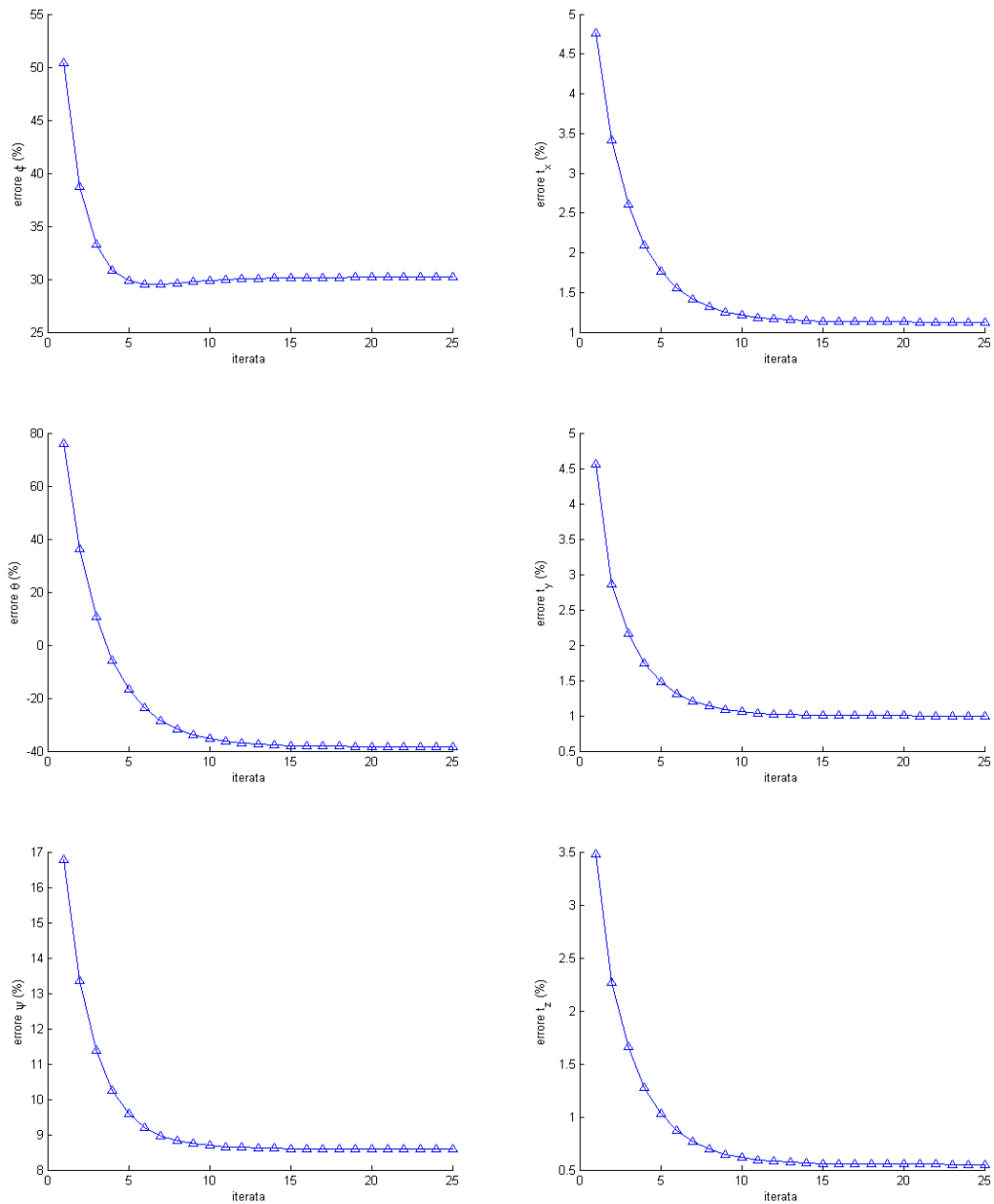


Figura 2.36: Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, $f=50$ mm, $\sigma_{err} = 10^{-4}$ m

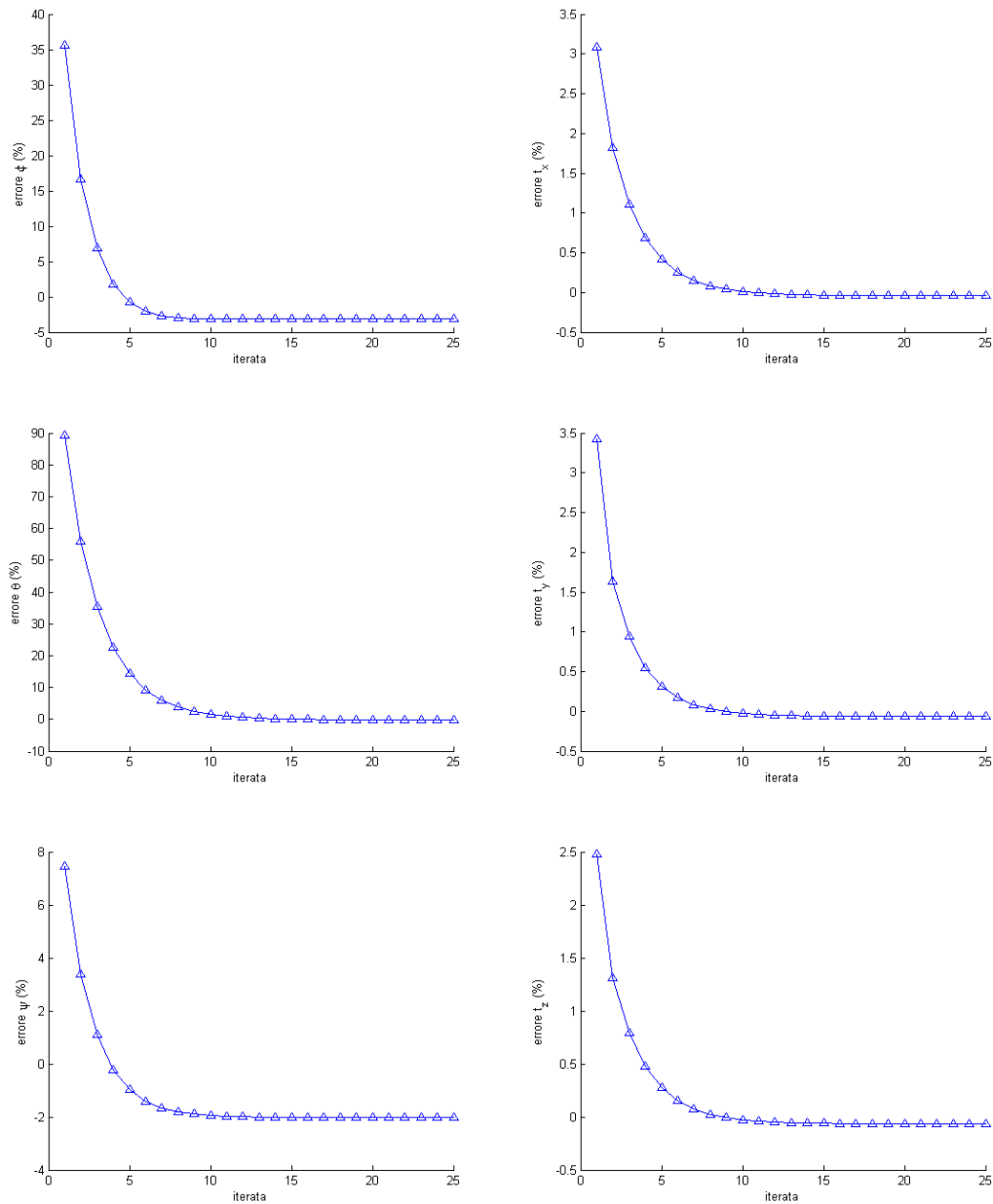


Figura 2.37: Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5, 0, 4, 3]^T$ m, $f=50$ mm, $\sigma_{err} = 10^{-5}$ m

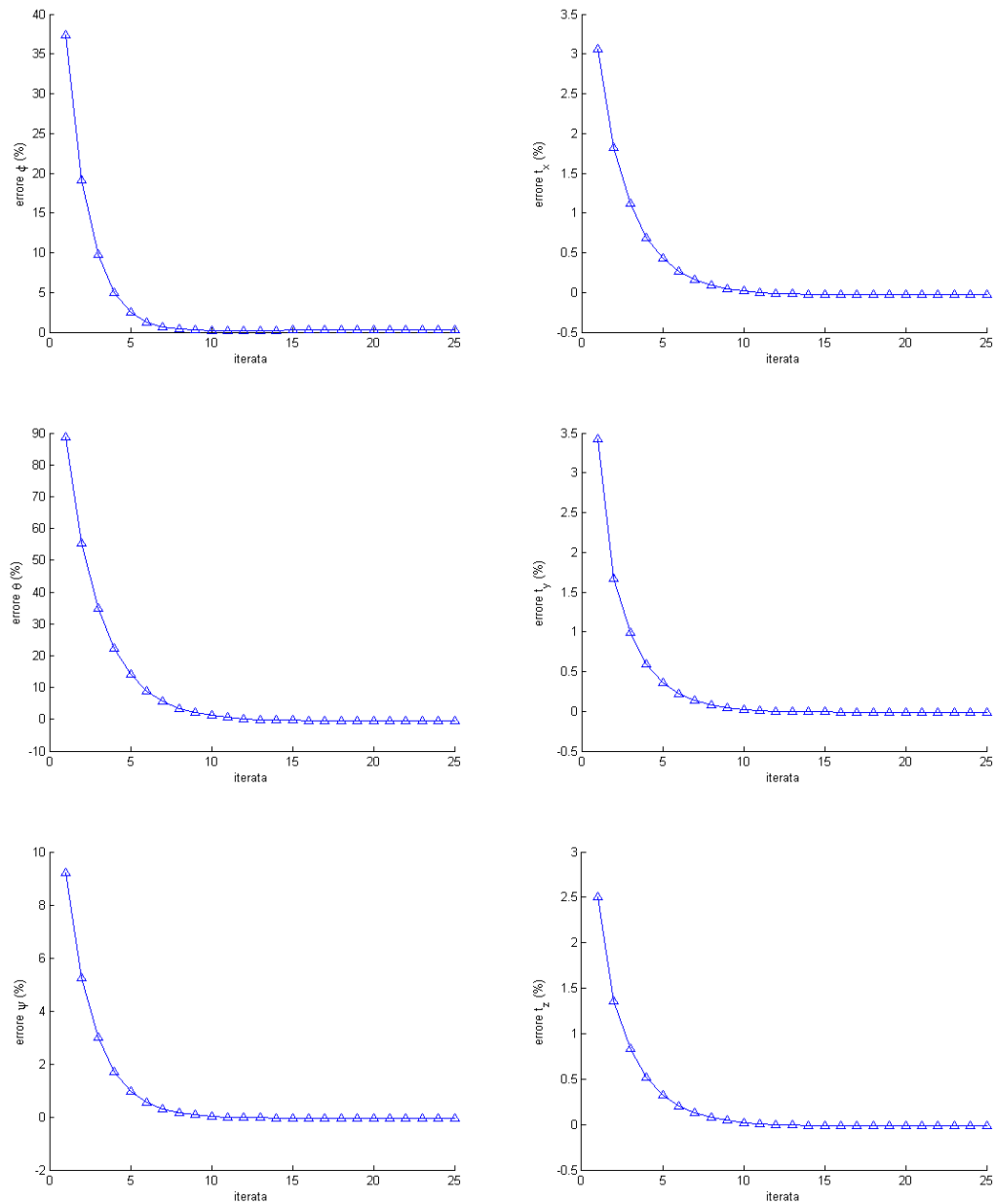


Figura 2.38: Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, $f=50$ mm, $\sigma_{err} = 10^{-6}$ m

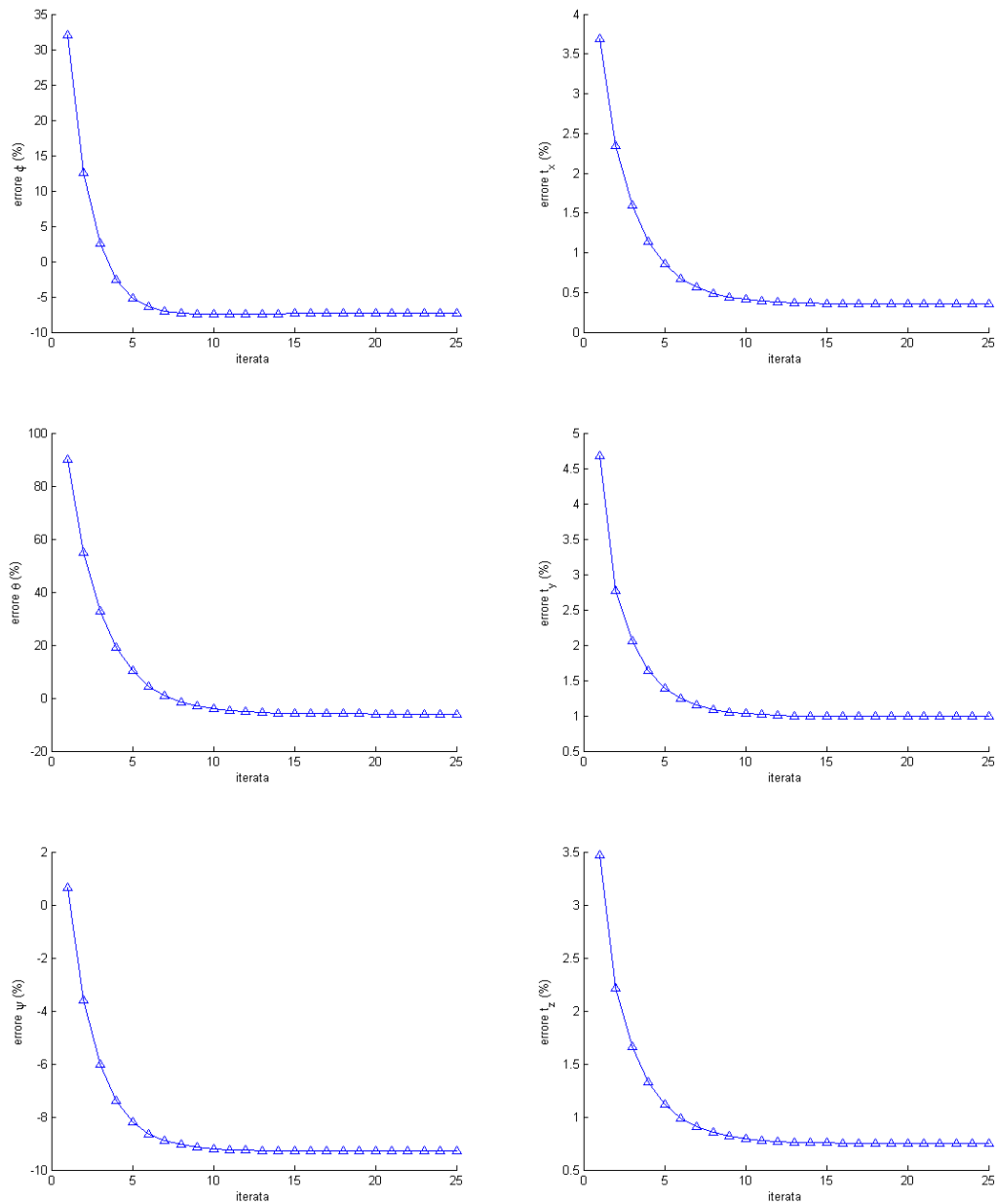


Figura 2.39: Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5, 0, 4, 3]^T$ m, $f=100$ mm, $\sigma_{err} = 10^{-4}$ m

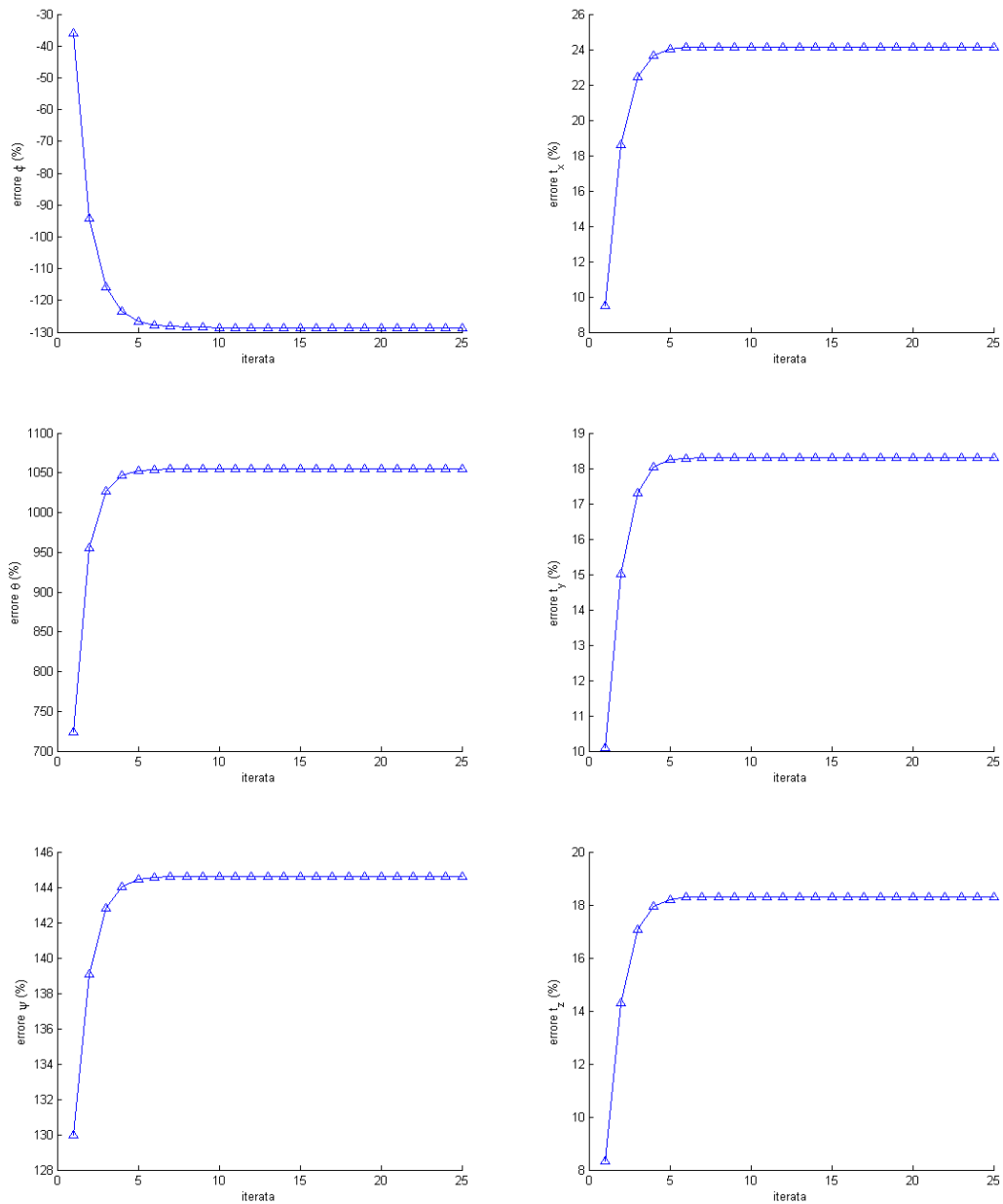


Figura 2.40: Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, $f=5$ mm, $\sigma_{err} = 10^{-4}$ m

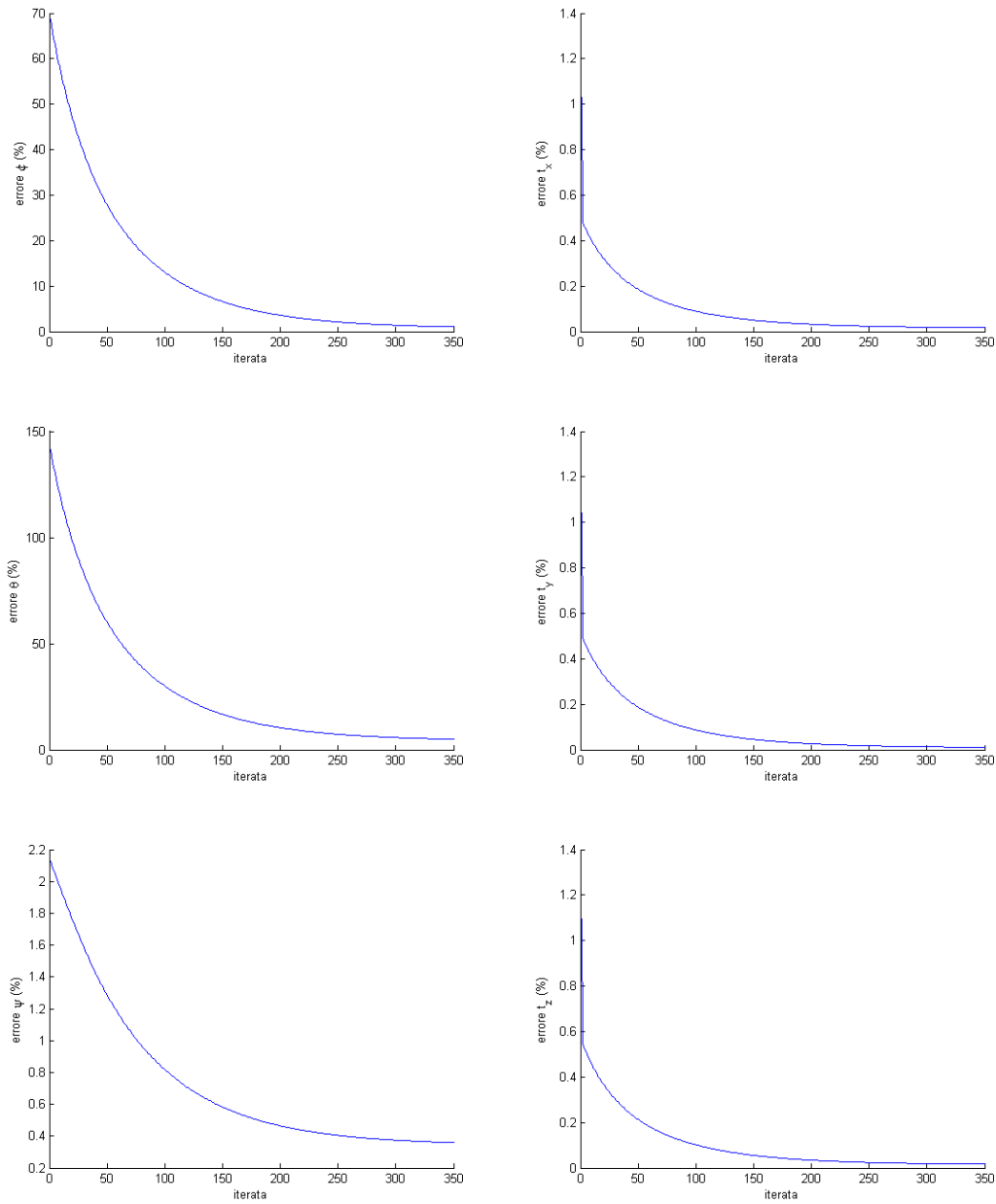


Figura 2.41: Algoritmo 2-stage; valori reali $\phi = 3^\circ$, $\theta = -2^\circ$, $\psi = 6^\circ$, $\mathbf{t} = [0, 5 \ 0, 4 \ 3]^T$ m, $f=50$ mm, $\sigma_{err} = 10^{-6}$ m, punti caratteristici nel piano $z=0$ m

t_z [m]	EPnP		EPnP+GN (4 iter.)		2-stage (20 iter.)		EPnP+2-stage (4 iter.)	
	$e_M R$ [°]	$e_M \mathbf{t}$ [m]	$e_M R$ [°]	$e_M \mathbf{t}$ [m]	$e_M R$ ($\cdot 10^{-3}$) [°]	$e_M \mathbf{t}$ [m]	$e_M R$ [°]	$e_M \mathbf{t}$ [m]
2-10	0,0412	0,1229	0,1246	0,2189	2,2437	0,2189	9,4313	0,8381
11-20	0,0748	0,1403	0,2485	0,1217	4,2676	0,8677	10,1429	1,6358
21-30	0,2216	0,1547	0,5481	0,1217	6,8569	2,1531	10,8596	3,0365
31-40	0,2551	0,2521	0,6473	0,1218	9,0898	3,6943	12,9581	4,7866
41-50	0,1765	0,2976	0,7480	0,1219	13,3064	6,2734	15,2052	7,3386

Tabella 2.1: Confronto algoritmo 2 – stadi e *EPnP* per la determinazione della posa; t_z variabile fra 2 e 50 m, $t_x = 0,5$ e $t_y = 0,4$ m ; sono riportati gli errori massimi sugli angoli e su \mathbf{t}

2.7 Confronto fra l' algoritmo a 2 stadi e quello di Lepetit

I risultati numerici presentati nei paragrafi precedenti sono utili a capire il carattere generale di un algoritmo, ma non sono sufficienti a decretarne la precisione assoluta e la robustezza; a tal fine sarebbe necessario intraprendere una campagna numerica e sperimentale di test atti a caratterizzare la qualità della misura. Nel ambito del presente lavoro, che si propone di effettuare uno *studio preliminare* e non una caratterizzazione/dimensionamento dello strumento di misura, è comunque necessario progettare un test numerico minimo per effettuare un confronto e avvalorare altresì la correttezza delle conclusioni tratte nei precedenti paragrafi.

A tal fine si è rilevato l' *errore massimo* in termini di assetto, quindi angoli, e distanza relativa per ogni possibile combinazione di angoli compresa fra -12° e 12° con passo di 1° , per una distanza variabile tra 2 e 50 metri con passo di 1 metro; la varianza dei disturbi posta pari a $\sigma = 10^{-5}$ m; i risultati in tabella 2.1.

Si sottolinea che i risultati non hanno valore statistico ma devono essere analizzati criticamente valutando l'ordine di grandezza, comunque corretto, e soprattutto effettuando un confronto; va inoltre precisato che per una varianza dei disturbi pari a 10^{-4} m l' algoritmo a 2 stadi ha fornito risultati non accettabili.

Per quanto riguarda l' algoritmo *EPnP* notiamo che l' errore massimo è estremamente contenuto e soprattutto è poco influenzato dalla distanza relativa dei due oggetti; se effettuiamo indi l'ottimizzazione con *Gauss – Newton* i risultati migliorano di tre ordini di grandezza per quanto riguarda gli angoli mentre rimangono costanti sul vettore \mathbf{t} , il tutto in meno di cinque iterate; analizzando più

t_z [m]	EPnP+GN (1° iter.)		EPnP+GN (2° iter.)		EPnP+GN (3° iter.)		EPnP+GN (4° iter.)	
	e_MR ($\cdot 10^{-3}$) [°]	$e_M\mathbf{t}$ [m]	e_MR ($\cdot 10^{-3}$) [°]	$e_M\mathbf{t}$ [m]	e_MR ($\cdot 10^{-3}$) [°]	$e_M\mathbf{t}$ [m]	e_MR ($\cdot 10^{-3}$) [°]	$e_M\mathbf{t}$ [m]
2-10	0,4417	0,1217	0,1246	0,1217	0,1246	0,1217	0,1246	0,1217
11-20	0,7243	0,1217	0,2485	0,1217	0,2485	0,1217	0,2485	0,1217
21-30	8,4231	0,1217	0,5481	0,1217	0,5481	0,1217	0,5481	0,1217
31-40	71,343	0,1218	0,6473	0,1218	0,6473	0,1218	0,6473	0,1218
41-50	6,0565	0,1219	0,7480	0,1219	0,7480	0,1219	0,7480	0,1219

Tabella 2.2: Soluzioni ottenute con l'algoritmo $EPnP$ in funzione del numero di iterazioni del metodo di *Gauss – Newton*

in dettaglio la convergenza delle iterazioni, visibile in tabella 2.2, si nota come solo 2 iterazioni siano sufficienti per ottenere la massima riduzione dell'errore.

E' invece deludente la prestazione dell'algoritmo iterativo che presenta un errore non trascurabile e, soprattutto, un degrado delle prestazioni all'aumentare della distanza; l'algoritmo è stato inizializzato con una matrice di rotazione che descrive un'attitudine pari a un grado in meno su ogni asse, in pratica è come se il satellite avesse velocità angolare pari a $1^\circ/s$ su ogni asse e l'integrazione temporale fosse di un secondo.

E' stato infine testato il caso in cui si utilizza l'algoritmo $EPnP$ per il calcolo della matrice d'assetto da utilizzare come *guess* per l'algoritmo a 2 stadi; il numero delle iterazioni è stato fissato a quattro per poter effettuare un confronto con *Gauss – Newton* anche se, è doveroso specificarlo, aumentandolo il risultato non cambiava in maniera sostanziale.

CAPITOLO 3

MODELLO DEL SATELLITE E DELL'AMBIENTE

In molti settori, ma soprattutto in quello spaziale, effettuare delle simulazioni e test sperimentali è spesso difficile/impossibile o semplicemente troppo costoso; per ovviare a tale problema si ricorre all'utilizzo dei calcolatori e di modelli matematico-fisici atti a rappresentare la realtà e l'oggetto dello studio con un grado di precisione consono allo scopo. Quindi in ambito spaziale si ricorre a delle equazioni che descrivono il moto orbitale, e tutte le azioni che le influenzano (ambiente), e ad altre che modellano la dinamica propria del corpo e la sua influenza sugli altri modelli (satellite).

Una volta ricreata la realtà di cui non disponiamo, potremo utilizzarla per testare i nostri programmi, quindi algoritmi di *stima della posizione e dell'assetto relativo*, *controllo* e *filtro di Kalman*.

3.1 Definizione dei sistemi di riferimento

Diversi sistemi di riferimento vengono normalmente impiegati nei progetti ingegneristici, ognuno dei quali risulta essere la scelta più comoda per la risoluzione del determinato problema.

3.1.1 Riferimento inerziale, I

E' il sistema di riferimento principale rispetto al quale vengono risolte le equazioni della meccanica; per applicazioni terrestri viene scelto con l'origine coincidente con il centro di massa della terra, l'asse \mathbf{x} (versore $\hat{\mathbf{x}}$) diretto verso il punto d'ariete γ , \mathbf{z} (versore $\hat{\mathbf{z}}$) diretto verso il polo nord celeste e l'asse \mathbf{y} (versore $\hat{\mathbf{y}}$) di conseguenza a completare la terna destrorsa; in questa particolare forma prende il nome *earth centered inertial (ECI)*. A rigore il punto d'Ariete presenta

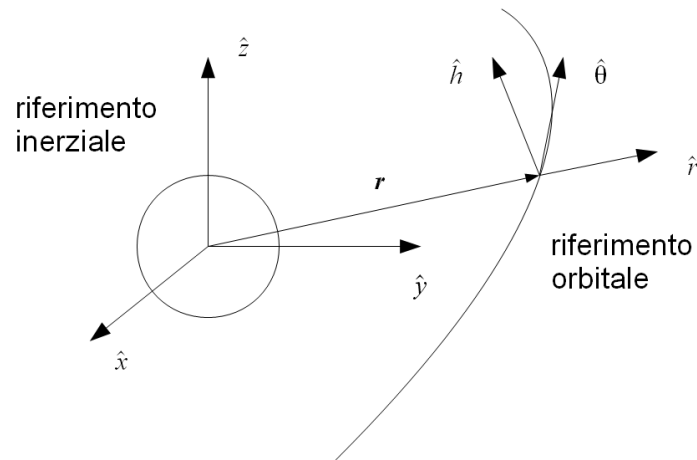


Figura 3.1: sistemi di riferimento per la meccanica orbitale

un moto non trascurabile sul lungo periodo ma per applicazioni ingegneristiche l'approssimazione a inerziale è perfettamente lecita.

3.1.2 Riferimento orbitale, O

Il *riferimento orbitale* è posizionato nel centro di massa del satellite ed è disposto in modo che l'asse \mathbf{x} coincida con la congiungente Terra-Satellite mentre l'asse \mathbf{z} con quello del momento della quantità di moto; il terzo asse completa la terna destrorsa. I versori di questo riferimento vengono indicati con $\hat{\mathbf{r}}$, $\hat{\boldsymbol{\theta}}$, $\hat{\mathbf{h}}$ e valgono

$$\hat{\mathbf{r}} = \frac{\mathbf{r}}{|\mathbf{r}|} \quad (3.1)$$

$$\hat{\mathbf{h}} = \frac{\mathbf{r} \wedge \mathbf{v}}{|\mathbf{r} \wedge \mathbf{v}|} \quad (3.2)$$

$$\hat{\boldsymbol{\theta}} = \hat{\mathbf{r}} \wedge \hat{\mathbf{h}} \quad (3.3)$$

avendo indicato con \mathbf{r} e \mathbf{v} rispettivamente il vettore posizione e il vettore velocità espressi nel sistema di riferimento orbitale.; la matrice di rotazione per passare dal riferimento *inerziale* a quello *orbitale* vale

$$S_I^O = \begin{bmatrix} \hat{\mathbf{r}}^T \\ \hat{\boldsymbol{\theta}}^T \\ \hat{\mathbf{h}}^T \end{bmatrix} \quad (3.4)$$

Alternativamente, se si hanno a disposizione gli elementi orbitali, si calcola come

$$S_I^O = \begin{bmatrix} c(u)c(\Omega) - s(u)c(i)s(\Omega) & c(u)s(\Omega) + s(u)c(i)c(\Omega) & s(u)s(i) \\ -s(u)c(\Omega) - c(u)c(i)s(\Omega) & -s(u)s(\Omega) + c(u)c(i)c(\Omega) & c(u)s(i) \\ s(i)s(\Omega) & -s(i)c(\Omega) & c(i) \end{bmatrix} \quad (3.5)$$

avendo indicato con $s()$ e $c()$ le funzioni seno e coseno e posto $u = \theta + \omega$.

Quando si risolvono problemi di *docking* o di *formation flying* è necessario calcolare posizione, velocità e accelerazione relativi dei due satelliti espressi nel riferimento non inerziale orbitale che, per comodità, viene chiamato *riferimento di Hill* (H) oppure *LVLH* (*Local Vertical/Local Horizontal*).

3.1.3 Riferimento assi corpo, B

Anche questo riferimento è posto nel centro di massa del satellite ma, contrariamente a quello orbitale, è solidale al satellite stesso, tant'è che viene anche chiamato *riferimento solidale*, sebbene più comunemente indicato come *assi corpo* (*body*). La direzione degli assi viene scelta coincidente con quella degli assi principali d'inerzia poiché comporta una semplificazione del tensore d'inerzia che diviene diagonale.

La matrice di rotazione che trasforma le coordinate di un vettore dal sistema di riferimento corpo a quello inerziale assume una diversa forma in relazione al tipo di parametrizzazione scelta (angoli di Eulero, vettore di Gibbs, quaternioni). Se si utilizzano i quaternioni per parametrizzare l'assetto assoluto del satellite, tale matrice ha la seguente forma, [berni]; nel presente lavoro si è deciso di utilizzare i quaternioni che, sebbene privi di significato fisico, sono esenti da singolarità; la matrice che ruota un vettore dal riferimento inerziale a quello assi corpo è,

$$S_I^B = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \quad (3.6)$$

dove

$$\begin{aligned} q_1 &= e_1 \sin(\varphi/2) \\ q_2 &= e_2 \sin(\varphi/2) \\ q_3 &= e_3 \sin(\varphi/2) \\ q_4 &= \cos(\varphi/2) \end{aligned} \quad (3.7)$$

esprimono la normalizzazione rispetto alla rotazione della parametrizzazione con asse e angolo di Eulero.

3.2 Meccanica orbitale

Le orbite dei satelliti studiati sono posizionate nella bassa atmosfera, quindi i disturbi rappresentativi sono la resistenza aerodinamica e le forze legate all'asfe-

ricità della terra che, per uno studio preliminare come il presente, caratterizzano correttamente gli ordini di grandezza del problema. Tale approccio è in linea con quanto viene presentato in letteratura.[aggiungere ref. Dimauro, Cinese]

3.2.1 Equazione di Keplero

L'equazione di Keplero ci permette di calcolare la traiettoria di un oggetto in orbita intorno ad un pianeta istante per istante. Nel caso non si considerino i disturbi orbitali il problema è risolvibile per via analitica; diversamente occorre effettuare un'integrazione numerica diretta oppure calcolare le variazioni temporali dei parametri kepleriani con le equazioni di Gauss-Lagrange, integrare per avere la storia dei parametri stessi e quindi calcolare, se necessario, i vettori \mathbf{r} e \mathbf{v} .

Si è scelta l'integrazione diretta dell'equazione di Keplero

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{f}_P + \mathbf{f}_C \quad (3.8)$$

avendo indicato con μ la *costante planetaria* della terra, \mathbf{f}_P l'accelerazione di *perturbazione* e con \mathbf{f}_C l'accelerazione di *controllo*.

3.2.2 Determinazione dei parametri Kepleriani

L'integrazione analitica dell'equazione di Keplero, essendo essa un'equazione differenziale vettoriale del secondo ordine, necessita di sei costanti d'integrazione che hanno un significato fisico ben definito e sono chiamate parametri Kepleriani. E' utile talvolta poter passare da una rappresentazione cartesiana, \mathbf{r} e \mathbf{v} , ai para-

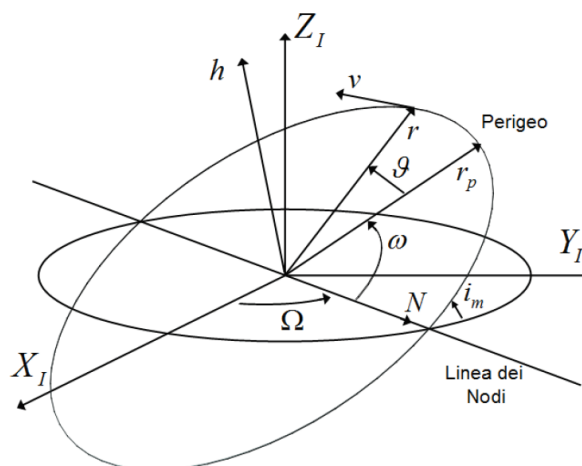


Figura 3.2: Visualizzazione dei 6 parametri Kepleriani

metri kepleriani (per un orbita perturbata non sono costanti); i passi da seguire sono i seguenti

- *semiasse maggiore*

$$a = \frac{\mu}{\frac{2\mu}{\sqrt{\mathbf{r} \cdot \mathbf{r}}} - \mathbf{v} \cdot \mathbf{v}} \quad (3.9)$$

- *eccentricità*

$$e = \sqrt{1 - \frac{(\mathbf{r} \wedge \mathbf{v}) \cdot (\mathbf{r} \wedge \mathbf{v})}{a\mu}} \quad (3.10)$$

- *inclinazione*

$$i = \arccos\left(\frac{h_z}{\mathbf{h}}\right) \quad (3.11)$$

avendo indicato con \mathbf{h} il momento delle quantità di moto e con h_z la sua componente lungo $\hat{\mathbf{z}}$

- *ascensione retta del nodo ascendente (RAAN)*

$$\Omega = \begin{cases} \arccos(\hat{\mathbf{x}} \cdot \hat{\mathbf{n}}) & \text{se } \hat{\mathbf{n}} \cdot \hat{\mathbf{y}} > 0 \\ 2\pi - \arccos(\hat{\mathbf{x}} \cdot \hat{\mathbf{n}}) & \text{se } \hat{\mathbf{n}} \cdot \hat{\mathbf{y}} < 0 \end{cases} \quad (3.12)$$

con n versore della *linea dei nodi* e calcolabile come segue

$$\hat{\mathbf{n}} = \frac{\hat{\mathbf{z}} \wedge \hat{\mathbf{h}}}{|\hat{\mathbf{z}} \wedge \hat{\mathbf{h}}|} \quad (3.13)$$

- *argomento del perigeo*

$$\omega = \begin{cases} \arccos(\hat{\mathbf{n}} \cdot \hat{\mathbf{e}}) & \text{se } \mathbf{e} \cdot \hat{\mathbf{z}} > 0 \\ 2\pi - \arccos(\hat{\mathbf{n}} \cdot \hat{\mathbf{e}}) & \text{se } \mathbf{e} \cdot \hat{\mathbf{z}} < 0 \end{cases} \quad (3.14)$$

con e vettore *eccentricità* e calcolabile come segue

$$\mathbf{e} = \frac{1}{\mu} \left[\left(v^2 - \frac{\mu}{r} \right) \mathbf{r} - (\mathbf{r} \cdot \mathbf{v}) \mathbf{v} \right] \quad (3.15)$$

- *anomalia vera*

$$\nu = \begin{cases} \arccos(\hat{\mathbf{e}} \cdot \hat{\mathbf{r}}) & \text{se } \mathbf{r} \cdot \mathbf{v} > 0 \\ 2\pi - \arccos(\hat{\mathbf{e}} \cdot \hat{\mathbf{r}}) & \text{se } \mathbf{r} \cdot \mathbf{v} < 0 \end{cases} \quad (3.16)$$

3.2.3 Disturbi orbitali: resistenza atmosferica

La resistenza dovuta all'interazione fra satellite e atmosfera non è trascurabile per orbite al di sotto dei 600 km; creare un modello dettagliato è molto difficile a causa della complessità intrinseca del problema, ma è possibile crearne uno verosimile che tenga conto dell'orientazione del satellite, della quota e delle superfici caratteristiche.

La forza aerodinamica risultante viene scomposta lungo la velocità relativa satellite-atmosfera e nella direzione perpendicolare assumendo i rispettivi nomi di *resistenza* e *portanza*; analizzando gli ordini di grandezza dei termini in gioco si nota che la velocità del satellite è molto maggiore rispetto a quella dell'atmosfera (causata dalla rotazione terrestre), quindi la seconda viene trascurata; inoltre, trattando con un corpo che è sostanzialmente tozzo, la *portanza* risulta trascurabile. In definitiva si ottiene

$$\mathbf{F}_a = -\frac{1}{2}\rho A v^2 C_D \hat{\mathbf{v}} \quad (3.17)$$

avendo indicato con A la superficie aerodinamica, con v il modulo della velocità del satellite, con C_D il coefficiente di resistenza (*drag* in inglese) e con ρ la densità dell'aria calcolata secondo un modello esponenziale del tipo

$$\rho(\mathbf{r}) = \rho_0 e^{-\left(\frac{\mathbf{r}-\mathbf{r}_0}{\lambda}\right)} \quad (3.18)$$

con λ opportuno fattore di scala. Nel riferimento orbitale possiamo scrivere l'accelerazione come

$$f_a^O = -\frac{1}{2} \frac{\rho C_D A}{m_s} \left[(|\mathbf{v}| \mathbf{v} \cdot \hat{\mathbf{r}}) \hat{\mathbf{r}} + (|\mathbf{v}| \mathbf{v} \cdot \hat{\boldsymbol{\theta}}) \hat{\boldsymbol{\theta}} + 0 \hat{\mathbf{h}} \right] \quad (3.19)$$

avendo indicato con m_s la massa del satellite; si noti che la componente nella terza direzione è nulla.

3.2.4 Disturbi orbitali: gradiente di gravità

A causa della non perfetta sfericità della terra (in generale di ogni pianeta) e della densità non uniforme si verifica la nascita di forze di disturbo sul satellite [**mengali**] di cui bisogna tener conto per continuare ad utilizzare il modello Kepleriano. Queste azioni dipendono dalla posizione spaziale del satellite e sono determinabili in forma di sommatoria in cui ogni termine viene pesato da un coefficiente sperimentale; senza addentrarsi negli sviluppi matematici e fermandosi al primo termine (indicato con J_2) si ottiene nel riferimento cartesiano

$$\mathbf{f}_{J_2}^I = -\frac{3}{2} J_2 \mu \frac{R_{terra}^2}{r^5} \begin{pmatrix} \left(1 - 5\left(\frac{z}{r}\right)^2\right) x \\ \left(1 - 5\left(\frac{z}{r}\right)^2\right) y \\ \left(3 - 5\left(\frac{z}{r}\right)^2\right) z \end{pmatrix} \quad (3.20)$$

avendo indicato con x , y e z le componenti di \mathbf{r} nel riferimento ECI ; analogamente nel riferimento orbitale ma in funzione dei parametri orbitali si ha

$$\mathbf{f}_{J_2}^O = -\frac{3\mu J_2 R_{terra}^2}{2r^4} \begin{pmatrix} 1 - 3\sin(\iota)^2 \sin(u)^2 \\ \sin(\iota)^2 \sin(2u) \\ \sin(2\iota) \sin(u) \end{pmatrix} \quad (3.21)$$

ovviamente nelle direzioni $\hat{\mathbf{r}}$, $\hat{\boldsymbol{\theta}}$ e $\hat{\mathbf{h}}$.

3.2.5 Equazioni di Hill

Nei problemi di *docking* e *formation flying* è spesso conveniente analizzare il moto orbitale relativo fra due o più satelliti dal punto di vista del riferimento di *Hill* e non del riferimento *inerziale*; nel nostro caso abbiamo a che fare con due satelliti, il primo chiamato *target* che è in orbita intorno alla terra e il secondo, definito come *chaser*, che deve avvicinarsi per effettuare il *docking*; il sistema di coordinate scelto è quello orbitale del *target*, che infatti viene anche chiamato *riferimento di Hill*; è un sistema non fisso in quanto soggetto ad accelerazioni e di ciò occorre tener conto nella formulazione delle equazioni.

Partendo dalla seguente relazione spaziale

$$\mathbf{r}_C^H = \mathbf{r}_T^H + \boldsymbol{\rho} \quad (3.22)$$

in cui si indica con ρ la distanza relativa fra i due satelliti in coordinate orbitali

$$\boldsymbol{\rho} = x\hat{\mathbf{r}} + y\hat{\boldsymbol{\theta}} + z\hat{\mathbf{h}} \quad (3.23)$$

Derivando due volte la 3.23 due volte ¹ si ottiene l'espressione dell'accelerazione

$$\ddot{\boldsymbol{\rho}} = -2\boldsymbol{\omega}_{H/I}^H \wedge \dot{\boldsymbol{\rho}} - \boldsymbol{\omega}_{H/I}^H \wedge (\boldsymbol{\omega}_{H/I}^H \wedge \boldsymbol{\rho}) - \dot{\boldsymbol{\omega}}_{H/I}^H \wedge \boldsymbol{\rho} + \Delta\mathbf{f}_P + \mathbf{f}_C \quad (3.24)$$

avendo indicato con $\boldsymbol{\omega}_{H/I}^H$ la velocità angolare della terna di Hill rispetto a quella inerziale espressa nel riferimento di Hill, con $\dot{\boldsymbol{\omega}}_{H/I}^H$ l'accelerazione angolare della terna di Hill rispetto a quella inerziale espressa nel riferimento di Hill, \mathbf{f}_C le accelerazioni di controllo espresse nel riferimento orbitale, e con $\Delta\mathbf{f}_P$ la differenza di accelerazioni di perturbazione fra *chaser* (pedice C) e *target* (pedice T), cioè

$$\Delta\mathbf{f}_P = (\mathbf{f}_{J_2,c}^H + \mathbf{f}_{aero,c}^H) - (\mathbf{f}_{J_2,t}^H + \mathbf{f}_{aero,t}^H) \quad (3.25)$$

¹si ricorda che i versori $\hat{\mathbf{r}}$, $\hat{\boldsymbol{\theta}}$ e $\hat{\mathbf{h}}$ sono variabili nel tempo; la loro derivata si calcola con la regola di Poisson: dati due sistemi di riferimento R_1 e R_2 con velocità angolare relativa $\boldsymbol{\omega}_{R_1/R_2}$, e dato un vettore \mathbf{r} , la derivata di \mathbf{r} in R_1 viene legata alla derivata di \mathbf{r} in R_2 mediante la relazione seguente

$$\dot{\mathbf{r}}^{R_1} = \dot{\mathbf{r}}^{R_2} + \boldsymbol{\omega}_{R_1/R_2} \wedge \mathbf{r}^{R_2}$$

E'utile esplicitare i termini dell'equazione 3.24 nel riferimento di Hill, a tal fine si ricordano le seguenti relazioni

$$\mathbf{r}_t = r_t \hat{\mathbf{r}}_t \quad (3.26a)$$

$$\mathbf{r}_c = (r_t + x) \hat{\mathbf{r}}_c + y \hat{\boldsymbol{\theta}}_c + z \hat{\mathbf{h}}_c \quad (3.26b)$$

e quindi, tenendo conto che la velocità angolare intorno all'asse $\hat{\boldsymbol{\theta}}$ è nulla perché il piano orbitale deve sempre passare per il centro d'attrazione, si ottengono

$$\begin{cases} \ddot{x} = \left(\omega_h^2 - \frac{\mu}{r_c^3}\right)x + \dot{\omega}_h - \omega_r \omega_h z + 2\omega_h \dot{y} + \mu \left(\frac{1}{r_t^2} - \frac{r_t}{r_c^3}\right) + \Delta f_x^P + f_x^C \\ \ddot{y} = -\dot{\omega}_h x + \left(\omega_r^2 + \omega_h^2 - \frac{\mu}{r_c^3}\right)y + \dot{\omega}_r z - 2\omega_h \dot{x} + 2\omega_r \dot{z} + \Delta f_y^P + f_y^C \\ \ddot{z} = -\omega_r \omega_h x - \dot{\omega}_r y + \left(\omega_r^2 - \frac{\mu}{r_c^3}\right)z - 2\omega_r \dot{y} + \Delta f_z^P + f_z^C \end{cases} \quad (3.27)$$

3.2.6 Determinazione dell'accelerazione angolare orbitale

Le equazioni di Hill, per essere risolte, richiedono la conoscenza del termine $\dot{\boldsymbol{\omega}}_{H/I}^H$; esso è ricavabile in due modi: derivazione numerica diretta della velocità orbitale oppure, conoscendo i modelli dei disturbi, calcolo per via analitica; il primo ha il vantaggio che può essere applicato indipendentemente dal tipo, dalla formulazione e dal numero di disturbi inseriti, presenta però il problema dell'inizializzazione: la derivata al passo corrente viene calcolata a partire da n valori precedenti, dove n coincide con l'ordine dello schema, e quindi occorre far iniziare l'integrazione delle equazioni di Hill n passi dopo l'inizio della simulazione; il secondo è invece utilizzabile fin dal primo passo ma, per essere esteso ad altri disturbi, deve essere modificato di conseguenza; l'uno può essere utilizzato per il test dell'altro e viceversa.

In presenza di disturbi orbitali, grazie alle equazioni di Lagrange-Gauss, è possibile ricavare la velocità angolare orbitale come segue

$$\boldsymbol{\omega}_{H/I}^H = \frac{r_t f_h}{h} \hat{\mathbf{r}}_t + 0 \hat{\boldsymbol{\theta}} + \frac{h}{r_t^2} \hat{\mathbf{h}} \quad (3.28)$$

avendo indicato con r il modulo del raggio (si ricorda che nel riferimento orbitale vale $\mathbf{r}_t = r_t \hat{\mathbf{r}}_t$), f_h l'accelerazione di disturbo lungo $\hat{\mathbf{h}}$ e h il modulo del momento delle quantità di moto; ricordando che vale la seguente

$$\dot{h} = r_t f_\theta \quad (3.29)$$

derivando analiticamente la formula 3.28 si ricava

$$\dot{\boldsymbol{\omega}}_{H/I}^H = \left(\frac{\dot{r}_t f_h}{h} + \frac{r_t \dot{f}_h}{h} - \frac{r_t^2 f_h f_\theta}{h^2}\right) \hat{\mathbf{r}}_t + 0 \hat{\boldsymbol{\theta}} + \left(\frac{t_\theta}{r_t} - 2 \frac{h \dot{r}_t}{r_t^3}\right) \hat{\mathbf{h}} \quad (3.30)$$

Per completare il quadro è necessario determinare ancora \dot{h} , quindi, partendo dalla componente lungo $\hat{\mathbf{h}}$ dell'equazione 3.21

$$f_h = -\frac{3\mu J_2 R_{terra}^2}{r_t^4} \sin(2i) \sin(u) \quad (3.31)$$

e derivando analiticamente si ottiene

$$f_h = \frac{-3\mu J_2 R_{terra}^2}{r_t^4} \left[-2 \sin(u) \sin(2i) \frac{\dot{r}_t}{r_t} + \sin(u) (\cos(i)^2 - \sin(i)^2) i + \frac{1}{2} \sin(2i) \cos(u) \dot{u} \right] \quad (3.32)$$

L'equazione 3.31 è costituita solo dalla componente lungo $\hat{\mathbf{h}}$ delle accelerazioni causate dall'asfericità del pianeta, infatti, la resistenza aerodinamica non presenta componenti in tale direzione; qualora il modello o il tipo di disturbi cambiasse in maniera tale da modificare l'equazione 3.31, occorrerebbe ricavare nuovamente l'equazione 3.32.

E' infine possibile calcolare le variazioni dei parametri Kepleriani d'interesse a partire dal seguente sistema [airchina]

$$\begin{cases} \omega_r = \dot{\Omega} \sin(i) \sin(u) + i \cos(u) & = \frac{r_t}{h} f_h \\ \omega_\theta = \dot{\Omega} \sin(i) \cos(u) - i \sin(u) & = 0 \\ \omega_h = \dot{\Omega} \cos(i) + \dot{u} & = \frac{h}{r_t^2} \end{cases} \quad (3.33)$$

si ottengono

$$i = \frac{r_t}{h} f_h \cos(u) \quad (3.34a)$$

$$\dot{\Omega} = \frac{r_t \sin(u)}{h \sin(i)} f_h \quad (3.34b)$$

$$\dot{u} = \frac{h}{r_t^2} - \frac{r_t}{h} \sin(u) \cot(i) \quad (3.34c)$$

Le equazioni 3.30, 3.32 e 4.1 permettono la determinazione di $\dot{\boldsymbol{\omega}}_{H/I}^H$ in forma chiusa.

3.3 Dinamica dell'assetto

3.3.1 Equazioni di Eulero

La dinamica angolare del corpo rigido è descritta dalle equazioni di Eulero

$$\begin{cases} I_x \dot{\omega}_x + (I_z - I_y) \omega_z \omega_y & = M_x^P + M_x^C \\ I_y \dot{\omega}_y + (I_x - I_z) \omega_x \omega_z & = M_y^P + M_y^C \\ I_z \dot{\omega}_z + (I_y - I_x) \omega_y \omega_x & = M_z^P + M_z^C \end{cases} \quad (3.35)$$

in cui I_x , I_y e I_z sono le componenti del tensore d'inerzia scritto in *assi baricentrici e principali d'inerzia*, ossia

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (3.36)$$

mentre con M^P e M^C rispettivamente le coppie di perturbazione e di controllo applicate al satellite.

Le equazioni vengono integrate numericamente in assi corpo a partire dalle seguenti condizioni iniziali

$$\boldsymbol{\omega}(t_0) = \boldsymbol{\omega}_0 \quad (3.37)$$

3.3.2 Parametrizzazione dell'assetto con i quaternioni

Come già anticipato nel paragrafo 3.1.3, la parametrizzazione dell'assetto viene effettuata utilizzando i quaternioni; è quindi necessario avere a disposizione una legge matematica che ci descriva come variano nel tempo in relazione alla posizione angolare assunta dal corpo nello spazio. Attraverso le 3.35 siamo in grado di ricavare le velocità angolari che poi utilizziamo per l'assetto attraverso la seguente

$$\dot{\mathbf{q}} = \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}) \mathbf{q}(t) \quad (3.38)$$

dove

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} = \begin{bmatrix} -[\boldsymbol{\omega}^\wedge] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & \mathbf{0} \end{bmatrix} \quad (3.39)$$

Anche in questo caso l'equazione viene integrata a partire dalle seguenti condizioni iniziali

$$\mathbf{q}(t_0) = \mathbf{q}_0 \quad (3.40)$$

3.3.3 Disturbi sull'assetto: resistenza atmosferica

Le coppie di disturbo indotte dall'atmosfera vengono determinate suddividendo il satellite in elementi di area, su ognuno è applicata una forza di disturbo infinitesima. Indicando con \mathbf{b} il vettore distanza tra il centro di massa del satellite e il punto in cui è applicata la forza, la coppia aerodinamica vale:

$$\mathbf{M}_{aero} = \int_B \mathbf{b} \wedge d\mathbf{f}_{aero} = \frac{1}{2} \rho C_D v^2 \int_B (\hat{\mathbf{N}} \cdot \hat{\mathbf{v}}) (\hat{\mathbf{v}} \wedge \mathbf{b}) dA \quad (3.41)$$

avendo indicato con $\hat{\mathbf{N}}$ il versore della normale all'elemento di area considerato. La 3.41 può essere semplificata riducendo l'integrale ad una sommatoria.

$$\mathbf{M}_{aero} = \sum \mathbf{b}_i \wedge \mathbf{F}_{aero} \quad (3.42)$$

3.3.4 Disturbi sull'assetto: gradiente di gravità

Essendo il satellite un corpo tridimensionale, non tutti i suoi punti si trovano alla stessa distanza dal centro di massa del pianeta e quindi si generano delle differenze di forze che globalmente generano una coppia perturbativa.

Facendo le seguenti ipotesi:

- si considera un unico attrattore con distribuzione di massa sferica
- il satellite è rigido e le sue dimensioni sono piccole rispetto al raggio orbitale

si ricava il modello

$$\mathbf{M}_{gg} = \frac{3\mu}{r^3} \begin{pmatrix} (I_z - I_y) c_2 c_3 \\ (I_x - I_z) c_1 c_3 \\ (I_y - I_x) c_1 c_2 \end{pmatrix} \quad (3.43)$$

avendo indicato con c_1 , c_2 e c_3 i coseni direttori della radiale in assi principali d'inerzia.

CAPITOLO 4

CONTROLLO OTTIMO NON-LINEARE E FILTRO DI KALMAN ESTESO (EKF)

A causa delle non-linearità che caratterizzano le equazioni dell'assetto e della meccanica orbitale, è necessario utilizzare un sistema di controllo e un filtro di kalman sviluppati appositamente per tener conto delle non-linearità. In particolare, per il filtro di Kalman, si è scelto un approccio classico che prevede la linearizzazione del sistema e quindi l'applicazione delle medesime equazioni del filtro lineare discreto; invece, per quanto riguarda il controllo, si è optato per un nuovo approccio che non linearizza le equazioni ma, scrivendole in una forma pseudo-lineare, riesce comunque a risolvere il problema del controllo ottimo.

4.1 Evoluzione del controllo ottimo di sistemi non-lineari

Il problema del controllo ottimo di sistemi non lineari viene affrontato con diverse tecniche, alcune delle quali proposte negli ultimi vent'anni a testimonianza della complessità del problema e, soprattutto, della necessità di ottenere una soluzione computazionalmente vantaggiosa sfruttabile per applicazioni real-time; l'approccio classico consiste nella linearizzazione delle equazioni in un punto di equilibrio

$$\dot{\mathbf{x}} = A|_{\bar{\mathbf{x}}}\mathbf{x}(t) + B\mathbf{u}(t)$$

e quindi dell'applicazione delle tecniche valide per sistemi lineari quali *LQR*, *LQG*, *H₂* La linearizzazione implica però una perdita di informazioni non trascurabile laddove le non-linearità sono più marcate.

Nel 1962 Pearson propose la strategia di controllo SDRE, nel 1975 venne espansa da Wernli&Cook ma solo a metà degli anni novanta nuovi impulsi alla tecnica

vennero indipendentemente da Mracek&Cloutier e Friedland. L'algoritmo proposto si basa sulla fattorizzazione delle non-linearità nel prodotto del vettore di stato per una matrice che dipende a sua volta dallo stato stesso; quindi il problema non lineare è trasformato in un problema lineare (trasformazione non unica) contenente una matrice con coefficienti dipendenti dallo stato (*state dependent coefficient*, SDC)

$$\dot{\mathbf{x}} = A(\mathbf{x})\mathbf{x}(t) + B\mathbf{u}(t)$$

Risolvendo quindi un'equazione algebrica di Riccati (ARE), composta anche da matrici con termini dipendenti dallo stato (in inglese *state-dependent Riccati equation* (SDRE)), si ottiene la soluzione del controllo sub-ottimo.

La tecnica che invece viene presentata nel prossimo paragrafo è di recente concezione e differisce da quanto sopra esposto perché non risolve direttamente la SDRE ma, sfruttando comunque la scrittura pseudo-lineare del sistema, propone una soluzione approssimata dell'equazione di Hamilton-Jacobi-Bellman (HJB); questa viene ricavata come sommatoria di vari termini e risulta computazionalmente vantaggiosa rispetto a risolvere ad ogni passo la SDRE.

4.1.1 Controllo ottimo non lineare: il metodo θ - D

Il sistema non lineare tempo-invariante è descritto dall'equazione generale

$$\dot{\mathbf{x}} = f(\mathbf{x}) + B\mathbf{u} \quad (4.1)$$

il problema del controllo ottimo si propone di ricavare l'azione di controllo \mathbf{u} tale da minimizzare il funzionale

$$J = \frac{1}{2} \int_0^\infty [\mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u}] dt \quad (4.2)$$

dove Q è una matrice semi-definita positiva, R è una matrice definita positiva, B è una matrice costante e deve valere $f(0) = 0$; si assume inoltre che $f(\mathbf{x})$ sia continua e differenziabile.

La soluzione al problema si trova risolvendo l'equazione alle derivate parziali di Hamilton-Jacobi-Bellman (HJB)

$$\frac{\partial V^T}{\partial \mathbf{x}} f(\mathbf{x}) - \frac{1}{2} \frac{\partial V^T}{\partial \mathbf{x}} B R^{-1} B^T \frac{\partial V}{\partial \mathbf{x}} + \frac{1}{2} \mathbf{x}^T Q \mathbf{x} = 0 \quad (4.3)$$

dove $V(\mathbf{x})$ è la funzione di costo ottima

$$V(\mathbf{x}) = \min_{\mathbf{u}} \frac{1}{2} \int_0^\infty [\mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u}] dt \quad (4.4)$$

La condizione necessaria porta alla soluzione

$$\mathbf{u} = -R^{-1} B^T \frac{\partial V}{\partial \mathbf{x}} \quad (4.5)$$

che è però in generale molto difficile da ottenere.

Per ovviare a tale problema è stato formulato [missile] [controllo] il metodo $\theta - D$ il cui punto di partenza è la riformulazione dell'equazione 4.1 nella forma pseudo-lineare

$$\dot{\mathbf{x}} = F(\mathbf{x})\mathbf{x} + B\mathbf{u} = \left[A_0 + \theta \frac{A(\mathbf{x})}{\theta} \right] \mathbf{x} + \left[B_0 + \theta \frac{B(\mathbf{x})}{\theta} \right] \mathbf{u} \quad (4.6)$$

dove A_0 e B_0 sono matrici costanti tali che (A_0, B_0) sia stabilizzabile, applicativamente vengono scelte coincidenti con le matrici A e B valutate in zero $A_0(\mathbf{x} = \mathbf{0})$ e $B_0(\mathbf{x} = \mathbf{0})$, la coppia $[(A_0 + A(\mathbf{x})), (B_0 + B(\mathbf{x}))]$ deve essere controllabile ad ogni istante mentre la variabile θ è solo di convenienza e in generale viene scelta unitaria.

Si aggiunge quindi una perturbazione in serie di potenze al funzionale 4.2

$$J_P = \frac{1}{2} \int_0^\infty \left[\mathbf{x}^T \left(Q_0 + \theta \frac{Q(x)}{\theta} + \sum_{i=1}^\infty D_i \theta^i \right) \mathbf{x} + \mathbf{u}^T R \mathbf{u} \right] dt \quad (4.7)$$

e quindi l'equazione 4.3 si modifica di conseguenza

$$\frac{\partial V^T}{\partial \mathbf{x}} \left[A_0 + \theta \frac{A(\mathbf{x})}{\theta} \right] \mathbf{x} - \frac{1}{2} \frac{\partial V^T}{\partial \mathbf{x}} B R^{-1} B^T \frac{\partial V}{\partial \mathbf{x}} + \frac{1}{2} \mathbf{x}^T \left(Q_0 + \theta \frac{Q(x)}{\theta} + \sum_{i=1}^\infty D_i \theta^i \right) \mathbf{x} = 0 \quad (4.8)$$

Se a questo punto si assume un'espansione in serie di potenze anche per la derivata della funzione di costo ottima

$$\frac{\partial V(\mathbf{x})}{\partial \mathbf{x}} = \sum_{i=0}^\infty T_i(\mathbf{x}, \theta) \theta^i \mathbf{x} \quad (4.9)$$

dove $T_i(\mathbf{x}, \theta)$ è una matrice simmetrica; sostituendo quest'ultima nell'equazione 4.5 si ottiene la funzione di controllo ottima

$$\mathbf{u} = -R^{-1} B^T \sum_{i=0}^\infty T_i(\mathbf{x}, \theta) \theta^i \mathbf{x} \quad (4.10)$$

Per ottenere la matrice T si sostituisce la 4.9 nell'equazione 4.8 e, ponendo a zero i coefficienti della serie di potenze in θ , si ottiene un sistema; la prima è un'equazione algebrica di Riccati

$$T_0 A_0 + A_0^T T_0 - T_0 B_0 R^{-1} B_0^T T_0 + Q_0 = 0 \quad (4.11)$$

e viene risolta solo una volta per determinare T_0 ; ponendo

$$\tilde{A}_0 = A_0 - B_0 R^{-1} B_0^T T_0$$

le successive n equazioni divengono

$$\begin{aligned}
T_n \tilde{A}_0 + \tilde{A}_0^T T_n = \varepsilon_i(t) & \left[-\frac{T_{i-1} A(\mathbf{x})}{\theta} - \frac{A(\mathbf{x})^T T_{i-1}}{\theta} + \sum_{j=1}^{i-1} T_j B_0 R^{-1} B_0^T T_{i-j} + \right. \\
& \left. + \sum_{j=0}^{i-2} T_j \frac{B(\mathbf{x})}{\theta} R^{-1} \frac{B(\mathbf{x})^T}{\theta} T_{i-2-j} + \sum_{j=0}^{i-1} T_j \left(B_0 R^{-1} \frac{B(\mathbf{x})^T}{\theta} + \frac{B(\mathbf{x})}{\theta} R^{-1} B_0^T \right) T_{i-1-j} \right]
\end{aligned} \tag{4.12}$$

avendo posto

$$\varepsilon_i(t) = 1 - k_i e^{l_i t} \tag{4.13}$$

dove t è il tempo e k_i e l_i sono stati considerati costanti e pari rispettivamente a 0,999 e 0,01; questo termine è stato introdotto per evitare instabilità o comunque valori troppo elevati della soluzione. Operativamente sono stati considerati solo i primi due termini nella sommatoria della soluzione espressa in equazione 4.10.

4.1.2 Definizione del problema

Nelle fasi finali di una manovra di *docking* lo scenario che si prospetta è costituito da due satelliti in moto relativo che si devono avvicinare mantenendo un certo assetto relativo al fine di entrare in contatto correttamente; gli obiettivi da perseguire sono quindi due: uno di meccanica orbitale ed esprimibile in termini di distanza e velocità relativa in assi corpo, l'altro di assetto, sempre relativo, ed esprimibile in funzione della parametrizzazione scelta, nel nostro caso i quaternioni. Il *vettore di stato* assume la forma

$$\mathbf{X} = [\boldsymbol{\rho} \ \dot{\boldsymbol{\rho}} \ \boldsymbol{\omega}_{B_c}^{B_c} \ \mathbf{q}_r]^T \tag{4.14}$$

dove $\boldsymbol{\rho}$ e $\dot{\boldsymbol{\rho}}$ sono rispettivamente posizione e velocità relativa espresse nel riferimento di Hill, $\boldsymbol{\omega}_{B_c}^{B_c}$ è la velocità angolare del chaser rispetto al riferimento inerziale espressa nel riferimento assi corpo chaser e \mathbf{q}_r sono i quaternioni che parametrizzano l'assetto relativo dei due satelliti. La parte di meccanica orbitale si ottiene riarrangiando le equazioni di Hill nella forma pseudo-lineare, operazione che può però essere effettuata solo se l'ipotesi $f(0) = 0$ è verificata e in questo caso non lo è; per ovviare al problema si aumenta il vettore di stato del sistema introducendo lo stato s_a tale che **[missile] [controllo] [giuseppe]**

$$\dot{s}_a = -\lambda_a s_a \tag{4.15}$$

con $\lambda = 0,01$; e si moltiplicano e dividono i termini che non rispettano l'ipotesi (detti *bias*) per questo nuovo stato.

Inoltre, per migliorare l'efficacia dell'inseguimento, si implementa il controllo come un servomeccanismo e quindi lo stato è aumentato dall'integrale della posizione relativa $\boldsymbol{\rho}_I$; il vettore di stato definitivo è il seguente

$$\mathbf{X} = [\boldsymbol{\rho} \ \dot{\boldsymbol{\rho}} \ s_a \ \boldsymbol{\rho}_I \ \boldsymbol{\omega}_{B_c}^{B_c} \ \mathbf{q}_r]^T \tag{4.16}$$

Essendo il controllo basato sulla controreazione dello stato è necessario introdurre il vettore di stato di *riferimento*, cioè quello che vogliamo che il nostro stato raggiunga e che utilizziamo per il calcolo dell'errore

$$\mathbf{X}_{rif} = \left[\boldsymbol{\rho}_{rif}^H \dot{\boldsymbol{\rho}}_{rif}^H s_a \boldsymbol{\rho}_{I,rif}^H \boldsymbol{\omega}_{B_t/I}^{B_c} \mathbf{q}_{r,rif} \right]^T \quad (4.17)$$

Il sistema di riferimento in cui ogni stato è espresso dipende ovviamente dalle equazioni in cui compare, così i termini di meccanica orbitale relativa derivando dalle equazioni di Hill sono appunto espressi nel medesimo riferimento; ciò ci costringe a dover effettuare un passaggio ulteriore, infatti i nostri obiettivi sono espressi in assi corpo (la fotocamera e i punti caratteristici sono anch'essi solidali con i satelliti) e quindi

$$\boldsymbol{\rho}_{rif}^H = S_I^{B_t} S_h^I [0 \ 0 \ d]^T \quad (4.18)$$

essendo d la distanza cui vogliamo i due satelliti. Per la velocità, sfruttando le relazioni di Poisson già introdotte nel precedente capitolo, si ottiene

$$\dot{\boldsymbol{\rho}}_{rif}^H = S_I^H S_{B_t}^I \left[\boldsymbol{\omega}_{B_t/I}^{B_t} \wedge \right] [0 \ 0 \ d]^T - \left[\boldsymbol{\omega}_{H/I}^H \wedge \right] \boldsymbol{\rho}_{rif}^H \quad (4.19)$$

Lo scenario di simulazione implementato prevede che il satellite *target* sia libero di muoversi, cioè privo di controllo, mentre tutto l'onere di lavoro spetta al sistema di controllo del *chaser* che in pratica deve inseguire il satellite *target*; essendo le misure effettuate con una fotocamera si vuole che le due facce dei satelliti adibite ai meccanismi di docking, e su cui è ipotizzabile siano montati fotocamera e punti caratteristici, si guardino il più possibile; questa specifica è traducibile nell'imposizione che i due sistemi di riferimento assi corpo abbiano lo stesso assetto, da cui si deduce che i quaternioni relativi devono valere

$$\mathbf{q}_r = [0 \ 0 \ 0 \ 1]^T \quad (4.20)$$

Restano infine da definire le matrici peso Q , associata allo stato, ed R , associata all'azione di controllo, utilizzate nella formulazione del controllo ottimo e introdotte in equazione 4.2; il loro valore non può essere definito analiticamente ma costituisce un grado di libertà per il progettista che le può utilizzare per cercare di ottenere le prestazioni che desidera.

Il valore che assumono non ha un significato assoluto ma ha senso come peso relativo, cioè conta il rapporto fra Q ed R per ottenere variazioni nella regolazione. Quindi se la matrice Q è relativamente grande rispetto a R si otterranno delle azioni di controllo elevate e quindi un'azione rapida con variazioni dello stato contenute; se il peso relativo è invertito l'azione di controllo sarà limitata.

Esiste in ogni caso una regola per calcolare dei valori sensati, utilizzabili in ogni caso come punto di partenza per la ricerca delle prestazioni desiderate, e prevede di prendere le matrici peso diagonali con ogni termine calcolato come segue

$$Q_{ii} = \frac{1}{\text{valore max di } x_i^2} \quad (4.21)$$

$$R_{jj} = \frac{1}{\text{valore max di } u_j^2} \quad (4.22)$$

intendendo con valore massimo il valore che si vuole non venga ecceduto durante la fase di controllo.

Le matrici utilizzate durante le simulazioni sono state prese diagonali e costanti.

4.1.3 Simulazione numerica

Per verificare il sistema di controllo è stato creato il seguente scenario; il satellite *target* occupa un'orbita circolare con inclinazione di 45° e semiasse maggiore 6778,136 km; le condizioni iniziali del *chaser* sono formulate in termini relativi e prevedono velocità relativa nel riferimento non inerziale di Hill nulla e distanza relativa $[-2 \ -30 \ -3]$ m. L'assetto iniziale è descritto da una rotazione con angoli di Eulero YZX rispetto alla terna orbitale di riferimento, per entrambi corrispondente a $[0 \ 0 \ -90^\circ]$; la velocità di rotazione iniziale è nulla per entrambi. La massa totale di ciascun satellite è di 1000 kg ed è stata considerata costante per il tempo della simulazione; la matrice d'inerzia è pari a

$$I_t = I_c = \begin{bmatrix} 3073 & 0 & 0 \\ 0 & 646 & 0 \\ 0 & 0 & 3073 \end{bmatrix} \quad [kgm^2]$$

I pesi utilizzati nel controllo valgono

$$\begin{aligned} Q &= \text{diag}([100, 150, 100, 600, 5 \cdot 10^3, 1100, 0, 100, 1 \cdot 10^{-1}, 50, 1, 1, 1, 1, 1, 1, 1]) \\ R &= \text{diag}([5 \cdot 10^5, 5 \cdot 10^5, 8 \cdot 10^6, 10^{-5}, 10^{-5}, 10^{-5}]) \end{aligned}$$

Lo scenario prevede che il *chaser*, inizialmente occupante la posizione $[-2 \ 3 \ -50]$ m in assi corpo *target*, raggiunga la posizione $[0 \ 0 \ -10]$ m in assi corpo *target*; questa simulazione ricrea la situazione di avvicinamento finale in una ipotetica manovra di docking in cui la fotocamera, con vista lungo l'asse z positivo del riferimento corpo del *chaser*, deve puntare verso il *target*.

Considerazioni sui risultati

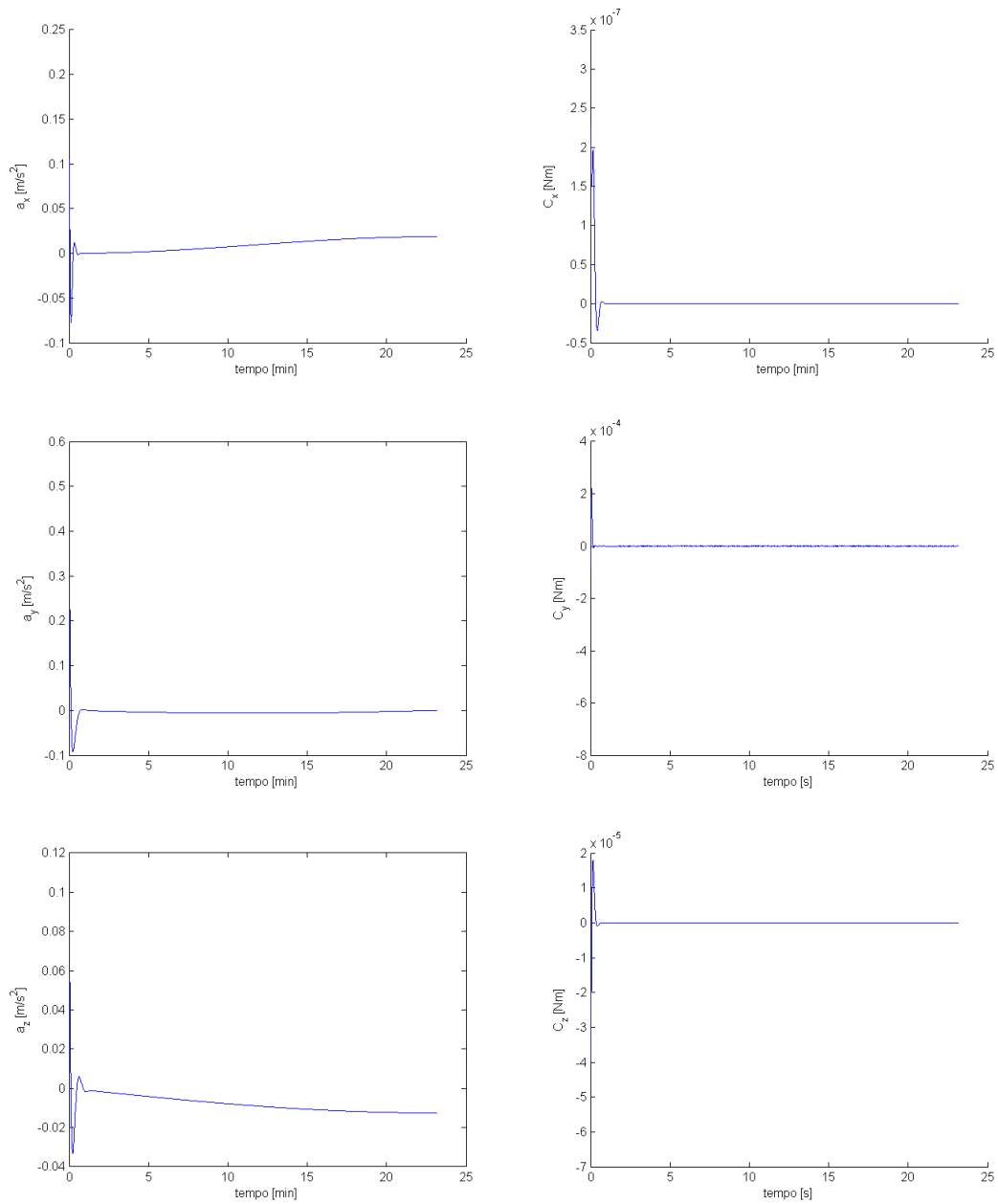


Figura 4.1: Accelerazioni nel riferimento orbitale e coppie di controllo

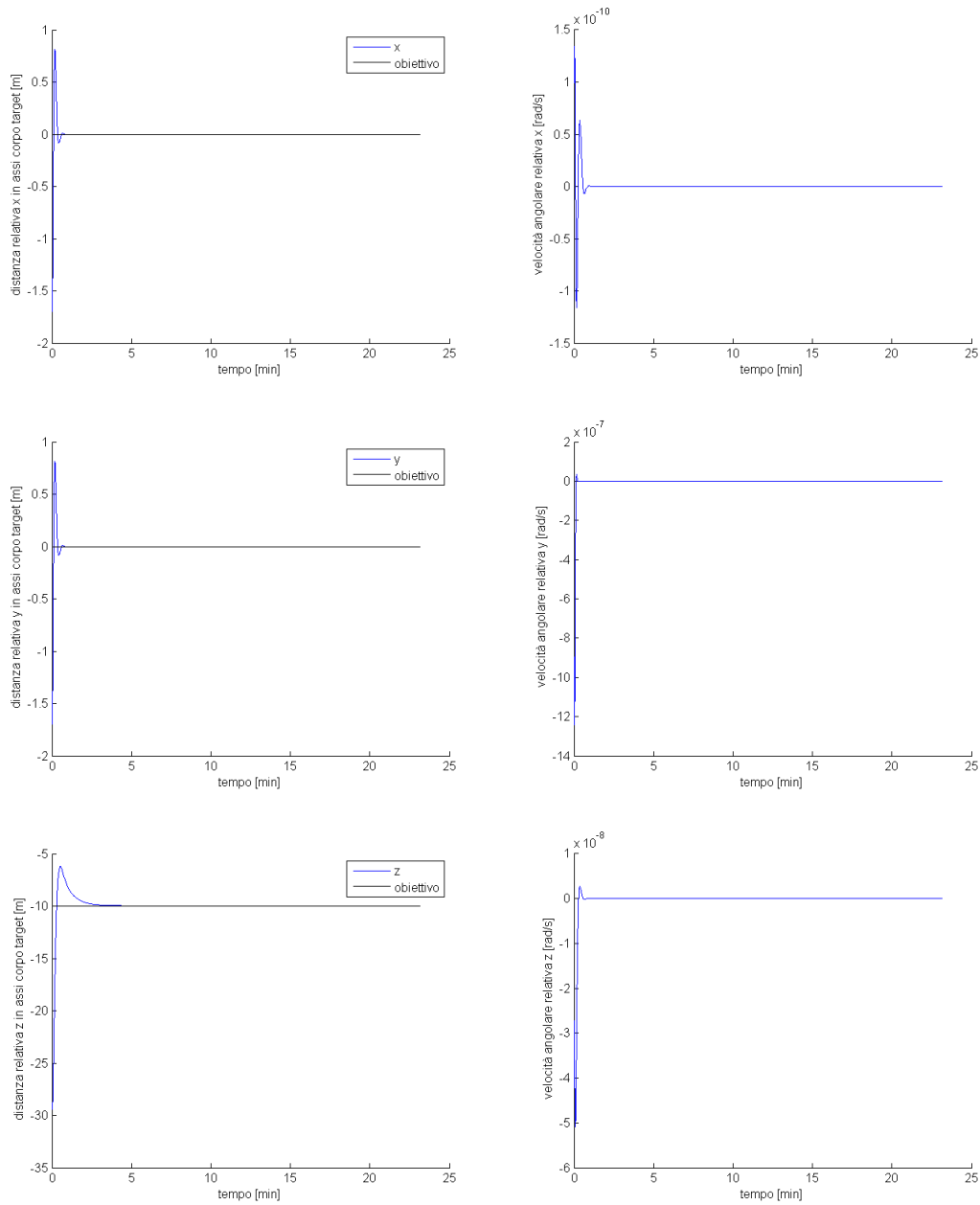


Figura 4.2: Posizione nel riferimento corpo target e velocità angolare relativa

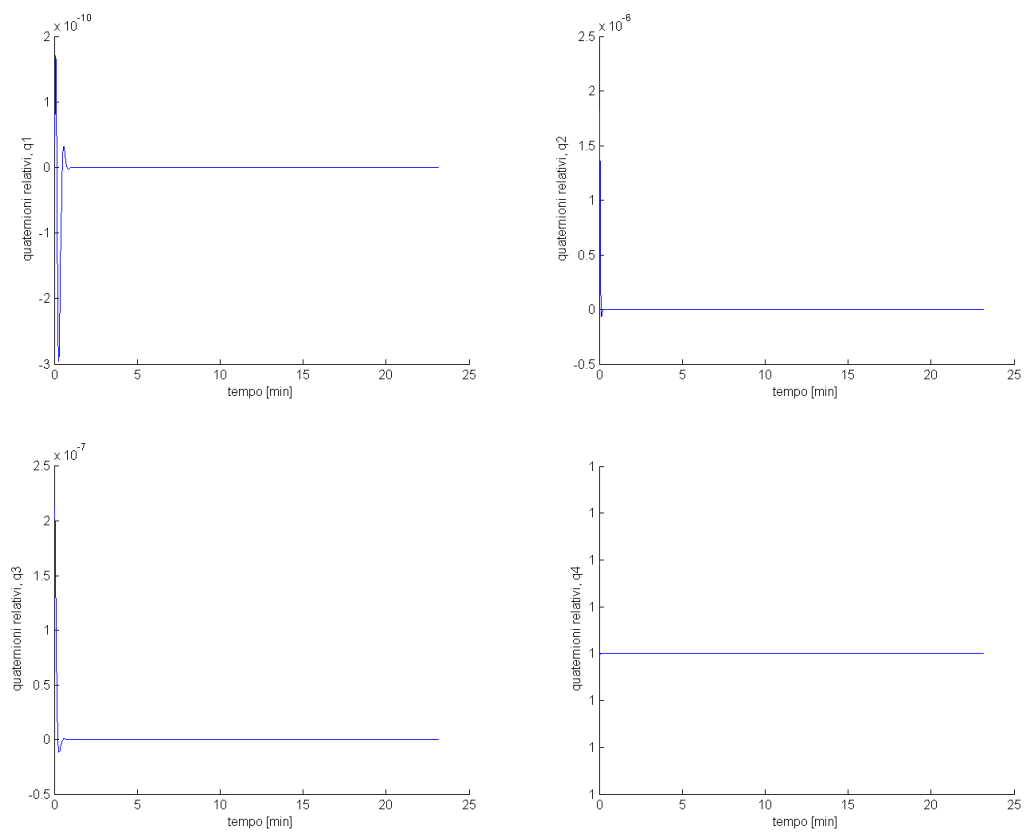


Figura 4.3: Quaternioni relativi

4.2 Filtro di Kalman non-lineare discreto

In ogni misura si ottiene un misurato che differisce dal misurando in proporzione alla qualità della misura, che dipende, fra gli altri, dall'entità del rumore presente; il filtro di Kalman, conoscendo la natura del rumore ed avendo un modello sufficientemente accurato del sistema, si propone di correggere per quanto possibile questo problema e di fornire quindi la miglior *stima* del misurando.

Il filtro si basa sulla definizione di un errore ottenuto come differenza tra le misure al passo temporale attuale (\tilde{z}_k) e la stima delle misure predetta, mediante il modello dinamico, dal passo temporale precedente (\hat{z}^+); viene quindi moltiplicato per un guadagno (*guadagno di Kalman*, K_k) e sommato allo stato¹ predetto (\hat{x}^+): questa costituisce la miglior stima del vettore di stato (che dipende implicitamente dalle misure). Quanto detto è visibile nella seguente equazione

$$\hat{x}^+ = \hat{x}^- + K_k [\tilde{z}_k - \hat{z}^+] \quad (4.23)$$

avendo indicato il passo temporale attuale con k e con gli apici $-$ e $+$ rispettivamente i valori predetti e quelli aggiornati.

Il processo è iterativo e viene ripetuto ad ogni passo, come visibile in figura

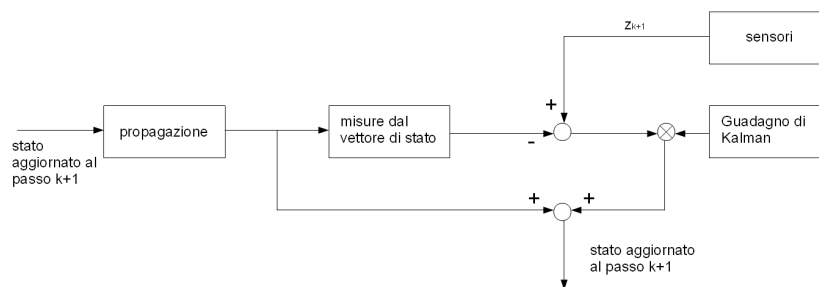


Figura 4.4: schema del filtro di Kalman

Il problema che ci troviamo ad analizzare presenta la caratteristica di non essere lineare, ciò impedisce di formulare un modello agli stati e conseguentemente applicare il filtro di Kalman; esso è in generale descrivibile dalla generica equazione non-lineare stocastica

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, w_k) \quad (4.24)$$

analogamente per le misure

$$z_{k+1} = h(\mathbf{x}_{k+1}, v_{k+1}) \quad (4.25)$$

¹si ricorda che lo stato non è univocamente definito ma viene scelto dal progettista in base alle esigenze

con w_k e v_k variabili casuali che rappresentano rispettivamente il *rumore di processo* e *di misura*, considerato *Gaussiano, bianco* e a *media nulla*.

Per ottenere un sistema agli stati si ricorre alla linearizzazione delle equazioni intorno ad un punto di equilibrio, nel nostro caso scegliamo lo stato predetto; questa scelta non è unica e soprattutto sensata se lo stato predetto non differisce troppo da quello vero; in quest'ottica scriviamo le predizioni dello stato e delle misure al passo temporale successivo senza considerare i disturbi

$$\hat{\mathbf{x}}_{k+1}^- = f\left(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, 0\right) \quad (4.26)$$

$$\hat{\mathbf{z}}_{k+1}^- = h\left(\hat{\mathbf{x}}_{k+1}^-, 0\right) \quad (4.27)$$

Nell'intorno di questi valori linearizziamo le equazioni

$$\mathbf{x}_{k+1} \approx \hat{\mathbf{x}}_{k+1}^- + F\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^+\right) + Ww_k \quad (4.28)$$

$$\mathbf{z}_{k+1} \approx \hat{\mathbf{z}}_{k+1}^- + H\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^+\right) + Vv_k \quad (4.29)$$

avendo indicato con F , W , H e V le matrici Jacobiane composte dalle derivate parziali

$$F_{i,j} = \frac{\partial f_i}{\partial \mathbf{x}_j}\left(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, 0\right) \quad (4.30)$$

$$W_{i,j} = \frac{\partial f_i}{\partial w_j}\left(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, 0\right) \quad (4.31)$$

$$H_{i,j} = \frac{\partial h_i}{\partial \mathbf{x}_j}\left(\hat{\mathbf{x}}_{k+1}^-, 0\right) \quad (4.32)$$

$$V_{i,j} = \frac{\partial h_i}{\partial v_j}\left(\hat{\mathbf{x}}_{k+1}^-, 0\right) \quad (4.33)$$

I disturbi non sono stati introdotti perché la loro azione è poco influente se il passo temporale è piccolo; il loro effetto viene recuperato pesando l'errore che dipende dalle misure (vedi equazione 4.24).

Definendo i seguenti errori

$$\hat{\mathbf{e}}_{x_{k+1}}^- = \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^- \quad (4.34)$$

$$\hat{\mathbf{e}}_{z_{k+1}}^- = \mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1}^- \quad (4.35)$$

è possibile riscrivere le equazioni 4.28 e 4.29 come segue

$$\hat{\mathbf{e}}_{x_{k+1}}^- = F\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^+\right) + Ww_k \quad (4.36)$$

$$H\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^+\right) + Vv_k = H\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^+\right) + Vv_k \quad (4.37)$$

Le equazioni fin qui ricavate ci permettono di sfruttare gli stessi risultati ottenuti per il filtro di Kalman lineare discreto; omettendo le dimostrazioni (vedi Cassidris) si riportano le formule del guadagno di Kalman e delle covarianze dell'errore

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (4.38)$$

$$P_k^+ = [I - K_k H_k] P_k^- \quad (4.39)$$

$$P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + W_k Q_k W_k^T \quad (4.40)$$

dove si è indicato con P e R rispettivamente le matrici delle covarianze degli errori dello stato e delle misure

4.2.1 La matrice di sensitività

Come introdotto nel precedente paragrafo la matrice di sensitività per l'EKF è il risultato di un processo di linearizzazione dell'equazione differenziale stocastica che modella le misure; l'obiettivo è di ricavare appunto una matrice che esprima puntualmente il legame fra il vettore di stato e le misure effettuate senza considerare i disturbi.

Introducendo (u_i, v_i) le osservazioni dei punti caratteristici sul piano immagine, (X_i, Y_i, Z_i) le posizioni note dei punti caratteristici nel riferimento di Hill e (x, y, z) le componenti del vettore che unisce il sistema di riferimento del target a quello del chaser espresso nel riferimento di Hill

[*metteredisegnino*]

si può scrivere

$$\mathbf{b}_i^H = S_H^{B_c} \mathbf{r}_i^H \quad (4.41)$$

dove

$$\mathbf{b}_i^{B_c} = \frac{1}{\sqrt{f^2 + u_i^2 + v_i^2}} \begin{pmatrix} u_i \\ v_i \\ f \end{pmatrix} \quad (4.42)$$

$$\mathbf{r}_i^H = \frac{1}{\sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2}} \begin{pmatrix} X_i - x \\ Y_i - y \\ Z_i - z \end{pmatrix} \quad (4.43)$$

Per il filtro di Kalman esteso è necessario linearizzare la relazione 4.41 rispetto ai quaternioni e rispetto alla distanza relativa, nel primo caso si ottiene

$$\frac{\partial \mathbf{b}_i}{\partial \mathbf{q}} = [2 S_H^{B_c} \mathbf{r}_i \wedge] \quad (4.44)$$

mentre nel secondo

$$\frac{\partial \mathbf{b}_i}{\partial \rho} = S_H^{B_c} \frac{\partial \mathbf{r}_i}{\partial \rho} \quad (4.45)$$

dove

$$\frac{\partial \mathbf{r}_i}{\partial \boldsymbol{\rho}} = \frac{1}{s_i} \begin{bmatrix} -[(Y_i - y)^2 + (Z_i - z)^2] & (X_i - x)(Y_i - y) & (X_i - x)(Z_i - z) \\ (X_i - x)(Y_i - y) & -[(X_i - x)^2 + (Z_i - z)^2] & (Y_i - y)(Z_i - z) \\ (X_i - x)(Z_i - z) & (Y_i - y)(Z_i - z) & -[(X_i - x)^2 + (Y_i - y)^2] \end{bmatrix} \quad (4.46)$$

con

$$s_i = [(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2]^{3/2}$$

Per non appesantire troppo la notazione si è ommesso di indicare i termini predetti, che sono

$$\boldsymbol{\rho} = \hat{\boldsymbol{\rho}}^- = [\hat{x}^- \hat{y}^- \hat{z}^-]^T \quad (4.47)$$

$$S_H^{Bc} = S_{B_t}^{Bc}(\hat{\boldsymbol{q}}^-)S_H^{Bt} \quad (4.48)$$

e influenzano le relazioni 4.43, 4.44, 4.45 e 4.46.

Per quanto riguarda la matrice delle covarianze degli errori di misura Shuster ha dimostrato che è possibile calcolarla come segue

$$R_k = \text{diag} [\sigma_1^2 \sigma_2^2 \dots \sigma_N^2] \odot I_{3 \times 3} \quad (4.49)$$

con N numero di punti caratteristici e \odot prodotto tensoriale di Kronecker.

4.2.2 Modello dei giroscopi

Entrambi i satellite montano tre giroscopi che misurano la velocità angolare rispetto ad un riferimento inerziale; queste informazioni vengono utilizzate dal filtro e quindi ci serve un modello dei giroscopi da poter manipolare. Una classica formulazione [**shuster**] è ottenuta sommando al valore vero² altri due termini: il primo è un errore casuale mentre il secondo (*drift*) è un termine crescente causato dalla natura integrale del giroscopio; in formule

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{\beta} + \boldsymbol{\eta}_v \quad (4.50)$$

$$\dot{\boldsymbol{\beta}} = \boldsymbol{\eta}_u \quad (4.51)$$

avendo indicato con $\tilde{\boldsymbol{\omega}}$ la velocità angolare misurata; $\boldsymbol{\eta}_v$ e $\boldsymbol{\eta}_u$ sono rumori Gaussiani bianchi indipendenti a media nulla con

$$E \{ \boldsymbol{\eta}_v(t) \boldsymbol{\eta}_v^T(\tau) \} = I_{3 \times 3} \sigma_v^2 \delta(t - \tau) \quad (4.52)$$

$$E \{ \boldsymbol{\eta}_u(t) \boldsymbol{\eta}_u^T(\tau) \} = I_{3 \times 3} \sigma_u^2 \delta(t - \tau) \quad (4.53)$$

Nel caso analizzato sono stati considerati giroscopi montati sia sul *target* che sul *chaser*.

²il termine *valore vero* ha senso nell'ambito di una simulazione numerica mentre in un applicazione reale si parla di *miglior stima del valore*

4.2.3 Soluzione analitica dell'equazione dell'assetto relativo a tempo-discreto

Nel capitolo 3 è stata introdotta l'equazione per calcolare l'evoluzione dei quaternioni nel tempo; quella stessa relazione è espandibile per l'assetto relativo [crassidris]

$$\dot{\mathbf{q}} = \frac{1}{2} \Theta(\boldsymbol{\omega}_t, \boldsymbol{\omega}_c) \mathbf{q} \quad (4.54)$$

la cui soluzione discreta vale

$$\mathbf{q}_{k+1} = \bar{\Omega}(\boldsymbol{\omega}_{c,k}) \bar{\Gamma}(\boldsymbol{\omega}_{t,k}) \mathbf{q}_k \quad (4.55)$$

con

$$\bar{\Omega}(\boldsymbol{\omega}_{c,k}) = \begin{bmatrix} \cos\left(\frac{1}{2} |\boldsymbol{\omega}_{c,k}| \Delta t\right) I_{3 \times 3} - [\psi_k \wedge] & \psi_k \\ \psi_k^T & \cos\left(\frac{1}{2} |\boldsymbol{\omega}_{c,k}| \Delta t\right) \end{bmatrix} \quad (4.56)$$

$$\bar{\Gamma}(\boldsymbol{\omega}_{t,k}) = \begin{bmatrix} \cos\left(\frac{1}{2} |\boldsymbol{\omega}_{t,k}| \Delta t\right) I_{3 \times 3} - [\zeta_k \wedge] & \zeta_k \\ -\zeta_k^T & \cos\left(\frac{1}{2} |\boldsymbol{\omega}_{t,k}| \Delta t\right) \end{bmatrix} \quad (4.57)$$

dove

$$\psi_k = \frac{\sin\left(\frac{1}{2} |\boldsymbol{\omega}_{c,k}| \Delta t\right) \boldsymbol{\omega}_{c,k}}{|\boldsymbol{\omega}_{c,k}|} \quad (4.58)$$

$$\zeta_k = \frac{\sin\left(\frac{1}{2} |\boldsymbol{\omega}_{t,k}| \Delta t\right) \boldsymbol{\omega}_{t,k}}{|\boldsymbol{\omega}_{t,k}|} \quad (4.59)$$

e con Δt si è indicato l'intervallo temporale di campionamento. Queste equazioni verranno utilizzate per la fase di propagazione dell'assetto.

4.2.4 EKF per l'assetto

In questo paragrafo viene ricavata la formulazione del EKF per la determinazione dell'assetto relativo da parte del *chaser*; a tal fine si suppone che le misure effettuate dai giroscopi del target e la matrice S_H^{Bt} siano disponibili al chaser in ogni istante utile.

L'utilizzo dei quaternioni per la parametrizzazione dell'assetto causa dei problemi nell'applicazione delle equazioni del filtro; essi sono principalmente causati dal fatto che i quattro quaternioni non sono indipendenti ma legati dall'equazione di normalizzazione. Questo vincolo rende singolare la matrice della covarianza relativa allo stato d'assetto. Per aggirare il problema [shuster] si considerano le quattro componenti come indipendenti e si introduce un nuovo concetto di errore

sullo stato: non è più la differenza aritmetica fra i quaternioni veri e quelli stimati ma il quaternioni che descrive la rotazione per passare dai quaternioni stimati a quelli veri; in quest'ottica descriveranno sicuramente una piccola rotazione e la componente scalare sarà prossima all'unità, la parte vettoriale conterrà quindi tutte le informazioni utili e verrà indicata con $\delta\alpha$.

Definendo $\delta\omega_c = \omega - \hat{\omega}_c$ e $\delta\omega_t = \omega_t - \hat{\omega}_t$ e ricordando la relazione per il calcolo della velocità angolare relativa

$$\omega_{ct} = \omega_c - S_H^{B_c}(\mathbf{q})\omega_t \quad (4.60)$$

si può scrivere

$$\omega_{ct} = \hat{\omega}_c - S_H^{B_c}(\mathbf{q})\hat{\omega}_t + \delta\omega_c - S_H^{B_c}(\mathbf{q})\delta\omega_t \quad (4.61)$$

Introducendo quindi i $\delta\alpha$, gli errori dei giroscopi e non considerando i termini di secondo ordine si ottiene [**crassidris**]

$$\delta\dot{\alpha} = -[\hat{\omega}_c \wedge] \delta\alpha - \Delta\beta_c + S_H^{B_c}(\hat{\mathbf{q}})\Delta\beta_t + S_H^{B_c}(\hat{\mathbf{q}})\eta_{tv} - \eta_{cv} \quad (4.62)$$

Definendo i seguenti vettori di stato

$$\Delta\mathbf{x} = [\delta\alpha^T \ \Delta\beta_t^T \ \Delta\beta_c^T]^T \quad (4.63)$$

$$\mathbf{w} = [\eta_{tv}^T \ \eta_{cv}^T \ \eta_{tu}^T \ \eta_{cu}^T]^T \quad (4.64)$$

la dinamica dell'errore sullo stato è in forma compatta

$$\Delta\dot{\mathbf{x}} = F\Delta\mathbf{x} + G\mathbf{w} \quad (4.65)$$

dove le matrici sono così definite

$$F = \begin{bmatrix} -[\hat{\omega}_c \wedge] & S_H^{B_c}(\hat{\mathbf{q}}) & -I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \quad (4.66)$$

$$G = \begin{bmatrix} S_H^{B_c}(\hat{\mathbf{q}}) & -I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \quad (4.67)$$

$$Q = \begin{bmatrix} \sigma_{tv}^2 I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \sigma_{cv}^2 I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{tu}^2 I_{3 \times 3} & I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{cu}^2 I_{3 \times 3} \end{bmatrix} \quad (4.68)$$

A questo punto è possibile fare uno schema che riassume i passi necessari per l'implementazione dell'algoritmo

- *Inizializzazione*: i valori dello stato e la matrice delle covarianze devono essere inizializzati

$$\hat{\mathbf{q}}(t_0) = \hat{\mathbf{q}}_0, \hat{\boldsymbol{\beta}}_t = \hat{\boldsymbol{\beta}}_{t_0}, \hat{\boldsymbol{\beta}}_c = \hat{\boldsymbol{\beta}}_{c_0}, \mathbf{P}(t_0) = \mathbf{P}_0 \quad (4.69)$$

- *Guadagno*: calcolo del guadagno di Kalman

$$K_k = P_k^- H_k^T(\mathbf{q}_k^-) \left[H_k(\mathbf{q}_k^-) P_k^- H_k^T(\mathbf{q}_k^-) + R_k \right]^{-1} \quad (4.70)$$

$$H_k(\mathbf{q}_k^-) = \begin{bmatrix} \left[S_H^{B_c}(\mathbf{q}_k^-) \mathbf{r}_1 \wedge \right] & 0_{3 \times 3} & 0_{3 \times 3} \\ \vdots & \vdots & \vdots \\ \left[S_H^{B_c}(\mathbf{q}_k^-) \mathbf{r}_N \wedge \right] & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}_{t_k} \quad (4.71)$$

- *Aggiornamento*: calcolo della miglior stima

$$P_k^+ = \left[I - K_k H_k(\mathbf{q}_k^-) \right] P_k^- \quad (4.72)$$

$$\Delta \hat{\mathbf{x}}_k^+ = K_k \begin{bmatrix} \mathbf{z}_{1,k} - S_H^{B_c}(\hat{\mathbf{q}}^-) \mathbf{r}_1 \\ \vdots \\ \mathbf{z}_{N,k} - S_H^{B_c}(\hat{\mathbf{q}}^-) \mathbf{r}_N \end{bmatrix}_{t_k} \quad (4.73)$$

$$\Delta \hat{\mathbf{q}}_k^+ = \frac{1}{2} \Xi(\hat{\mathbf{q}}_k^-) \delta \hat{\boldsymbol{\alpha}}_k^+ \quad (4.74)$$

$$\hat{\boldsymbol{\beta}}_{t,k}^+ = \hat{\boldsymbol{\beta}}_{t,k}^- + \Delta \hat{\boldsymbol{\beta}}_{t,k}^+ \quad (4.75)$$

$$\hat{\boldsymbol{\beta}}_{c,k}^+ = \hat{\boldsymbol{\beta}}_{c,k}^- + \Delta \hat{\boldsymbol{\beta}}_{c,k}^+ \quad (4.76)$$

Indicando con $\tilde{\boldsymbol{\omega}}$ la velocità angolare misurata, la si depura dell'errore di *bias*

$$\hat{\boldsymbol{\omega}}_{t,k}^+ = \tilde{\boldsymbol{\omega}}_{t,k} - \hat{\boldsymbol{\beta}}_{t,k}^+ \quad (4.77)$$

$$\hat{\boldsymbol{\omega}}_{c,k}^+ = \tilde{\boldsymbol{\omega}}_{c,k} - \hat{\boldsymbol{\beta}}_{c,k}^+ \quad (4.78)$$

Normalizzazione dei quaternioni

$$\Delta \hat{\mathbf{q}}_k^+ = \begin{bmatrix} \hat{q}_{k,1}^+ & \hat{q}_{k,2}^+ & \hat{q}_{k,3}^+ & \sqrt{1 - (\hat{q}_{k,1}^+)^2 - (\hat{q}_{k,2}^+)^2 - (\hat{q}_{k,3}^+)^2} \end{bmatrix} \quad (4.79)$$

$$\hat{\mathbf{q}}_k^+ = \hat{\mathbf{q}}_k^- + \Delta \hat{\mathbf{q}}_k^+ \quad (4.80)$$

$$\hat{\mathbf{q}}_k^+ = \frac{\hat{\mathbf{q}}_k^+}{|\hat{\mathbf{q}}_k^+|} \quad (4.81)$$

- *Propagazione*: calcolo della stima dei valori al passo successivo

$$\hat{\mathbf{q}}_{k+1}^- = \bar{\Omega} \left(\hat{\omega}_{c,k}^+ \right) \bar{\Gamma} \left(\hat{\omega}_{t,k}^+ \right) \hat{\mathbf{q}}_k^+ \quad (4.82)$$

$$P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + Q_k \quad (4.83)$$

dove con Φ si è indicata la matrice di transizione che è possibile calcolare seguendo il procedimento di Van Loaf [**crassidris**], valido per sistemi a parametri costanti e con passo temporale d'integrazione fisso; si compone la seguente matrice

$$\mathcal{A} = \begin{bmatrix} -F & GQG^T \\ 0 & F^T \end{bmatrix} \Delta t \quad (4.84)$$

utilizzata per calcolare la seguente

$$\mathcal{B} = e^{\mathcal{A}} = \begin{bmatrix} \mathcal{B}_{11} & \Phi^{-1}Q \\ 0 & \Phi^T \end{bmatrix} \quad (4.85)$$

da cui si evince che

$$\begin{aligned} \Phi &= \mathcal{B}_{22}^T \\ Q &= \Phi \mathcal{B}_{12} \end{aligned}$$

Se l'intervallo temporale d'integrazione è sufficientemente piccolo vale la relazione

$$Q_k = \Delta t G Q G^T \quad (4.86)$$

4.2.5 EKF per la determinazione dell'orbita e dell'assetto

L'estensione del filtro anche alla parte di meccanica orbitale prevede, oltre all'aggiunta delle variabili d'interesse al vettore di stato, anche l'espansione delle matrici conformemente al modello; in particolare la F deriva dalla linearizzazione delle equazioni di Hill.

Le equazioni che descrivono il moto relativo sono già state ricavate nel capitolo precedente e vengono qui riportate tralasciando però il termine $\omega_r = 0$ in concordanza con l'equazione 4.26; esso infatti nasce per la presenza dei disturbi che non vengono considerati nella fase di predizione; quindi

$$\begin{cases} \ddot{x} &= \left(\omega_h^2 - \frac{\mu}{r_c^3} \right) x + \dot{\omega}_h y + 2\omega_h \dot{y} + \mu \left(\frac{1}{r_t^2} - \frac{r_t}{r_c^3} \right) \\ \ddot{y} &= -\dot{\omega}_h x + \left(\omega_h^2 - \frac{\mu}{r_c^3} \right) y - 2\omega_h \dot{x} \\ \ddot{z} &= -\frac{\mu}{r_c^3} z \end{cases} \quad (4.87)$$

Il raggio del *chaser* viene definito come somma vettoriale del raggio del target e del vettore distanza relativa $\boldsymbol{\rho}$ espressi nel riferimento di *Hill*, in componenti

$$\mathbf{r}_c^H = \mathbf{r}_t^H + \boldsymbol{\rho}^H = (r_t + x)\hat{\mathbf{r}} + y\hat{\boldsymbol{\theta}} + z\hat{\mathbf{h}}$$

avendo indicato con (x, y, z) le componenti del vettore $\boldsymbol{\rho}$ nel riferimento di Hill. Notando che la distanza fra i satelliti è molto minore dei raggi orbitali si può effettuare l'approssimazione

$$\mathbf{r}_c^H = \sqrt{(x + r_t)^2 + y^2 + z^2} \approx \sqrt{r_t^2 + 2xr_t} = r_t \sqrt{1 + 2\frac{x}{r_t}}$$

ricordando che in un approssimazione al primo ordine vale

$$(1 + \delta)^n \approx 1 + n\delta \quad (\delta \ll 1)$$

possiamo scrivere

$$\frac{1}{r_c^3} = \frac{1}{r_t^3} \left(1 - 3\frac{x}{r_t}\right) \quad (4.88)$$

Con l'ausilio della 4.88 riscriviamo il sistema 4.87 come segue

$$\begin{cases} \ddot{x} = \frac{\mu}{r_t^3} \left(3\frac{x^2}{r_t} + 2x\right) + \omega^2 x + \dot{\omega}y + 2\omega\dot{y} \\ \ddot{y} = \frac{\mu}{r_t^3} \left(3\frac{x}{r_t} - 1\right) y - \omega x + \omega^2 y - 2\omega\dot{x} \\ \ddot{z} = \frac{\mu}{r_t^3} \left(3\frac{x}{r_t} - 1\right) z \end{cases} \quad (4.89)$$

Il sistema viene linearizzato e scritto agli stati formando una matrice di dimensioni 6×6 ; per dettagli vedere l'appendice. Il vettore di stato relativo alla sola parte orbitale è così composto

$$\mathbf{X}_{orb} = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^T \quad (4.90)$$

mentre quello totale relativo all'errore sullo stato

$$\mathbf{X} = [\delta\boldsymbol{\alpha}^T \ \Delta\boldsymbol{\beta}_t^T \ \Delta\boldsymbol{\beta}_c^T \ \Delta x \ \Delta y \ \Delta z \ \Delta\dot{x} \ \Delta\dot{y} \ \Delta\dot{z}]^T \quad (4.91)$$

Le matrici introdotte nel paragrafo precedente si modificano come segue

$$F = \begin{bmatrix} -[\hat{\boldsymbol{\omega}}_c \wedge] & S_H^{B_c}(\hat{\mathbf{q}}) & -I_{3 \times 3} & 0_{3 \times 10} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 6} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 6} \\ 0_{6 \times 3} & 0_{6 \times 3} & 0_{6 \times 3} & \left. \frac{\partial \mathbf{f}(\mathbf{X})}{\partial \mathbf{X}} \right|_{\hat{\mathbf{x}}} \end{bmatrix} \quad (4.92)$$

$$G = \begin{bmatrix} S_H^{B_c}(\hat{\mathbf{q}}) & -I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \quad (4.93)$$

$$Q = \begin{bmatrix} \sigma_{tv}^2 I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 1} \\ 0_{3 \times 3} & \sigma_{cv}^2 I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 1} \\ 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{tu}^2 I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 1} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{cu}^2 I_{3 \times 3} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 1} \\ 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & w_x & 0 & 0 \\ 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0 & w_y & 0 \\ 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0 & 0 & w_z \end{bmatrix} \quad (4.94)$$

I passi da seguire sono gli stessi formulati per l' EKF dell'assetto ma adattati alle nuove dimensioni del problema, quindi i valori iniziali sono

$$\hat{\mathbf{q}}(t_0) = \hat{\mathbf{q}}_0, \quad \hat{\boldsymbol{\beta}}_t = \hat{\boldsymbol{\beta}}_{t_0}, \quad \hat{\boldsymbol{\beta}}_c = \hat{\boldsymbol{\beta}}_{c0}, \quad \boldsymbol{\rho}(t_0) = \boldsymbol{\rho}_0, \quad \dot{\boldsymbol{\rho}}(t_0) = \dot{\boldsymbol{\rho}}_0 \quad (4.95)$$

Nel calcolo del guadagno la formula rimane la stessa ma la matrice di sensitività, in questo caso, contiene anche i termini relativi alle misure della distanza relativa

$$H_k(\mathbf{q})_k^- = \begin{bmatrix} [S_H^{B_c}(\mathbf{q})_k^- \mathbf{r}_1 \wedge] & 0_{3 \times 3} & 0_{3 \times 3} & \frac{\partial \hat{\mathbf{b}}_1^-}{\partial \hat{\boldsymbol{\rho}}^-} & 0_{3 \times 7} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ [S_H^{B_c}(\mathbf{q})_k^- \mathbf{r}_N \wedge] & 0_{3 \times 3} & 0_{3 \times 3} & \frac{\partial \hat{\mathbf{b}}_N^-}{\partial \hat{\boldsymbol{\rho}}^-} & 0_{3 \times 7} \end{bmatrix}_{t_k} \quad (4.96)$$

La fase di *aggiornamento* segue la procedura classica e non necessita di particolari accorgimenti; invece nella fase di *propagazione* non abbiamo a disposizione un'equazione dedicata come con i quaternioni che teneva conto implicitamente anche delle forzanti attraverso le misure delle velocità angolari dei giroscopi, dobbiamo quindi sfruttare il modello agli stati; a tal fine è necessario introdurre la matrice B che moltiplica le forze di controllo

$$B = \begin{bmatrix} 0_{12 \times 12} \\ I_{3 \times 3} \end{bmatrix} \quad (4.97)$$

la cui forma nel caso discretizzato vale

$$\Upsilon = \left[I \Delta t + \frac{1}{2} F \Delta t^2 \right] B \quad (4.98)$$

La stima al passo successivo si calcola con l'equazione alle differenze

$$\hat{\mathbf{X}}_{k+1}^- = \Phi \hat{\mathbf{X}}_k^+ + \Upsilon \mathbf{u}_k \quad (4.99)$$

avendo indicato con \mathbf{u} il vettore delle azioni di controllo; di quest'ultimo vettore si considera unicamente la parte relativa alla meccanica orbitale.

4.2.6 Simulazioni numeriche

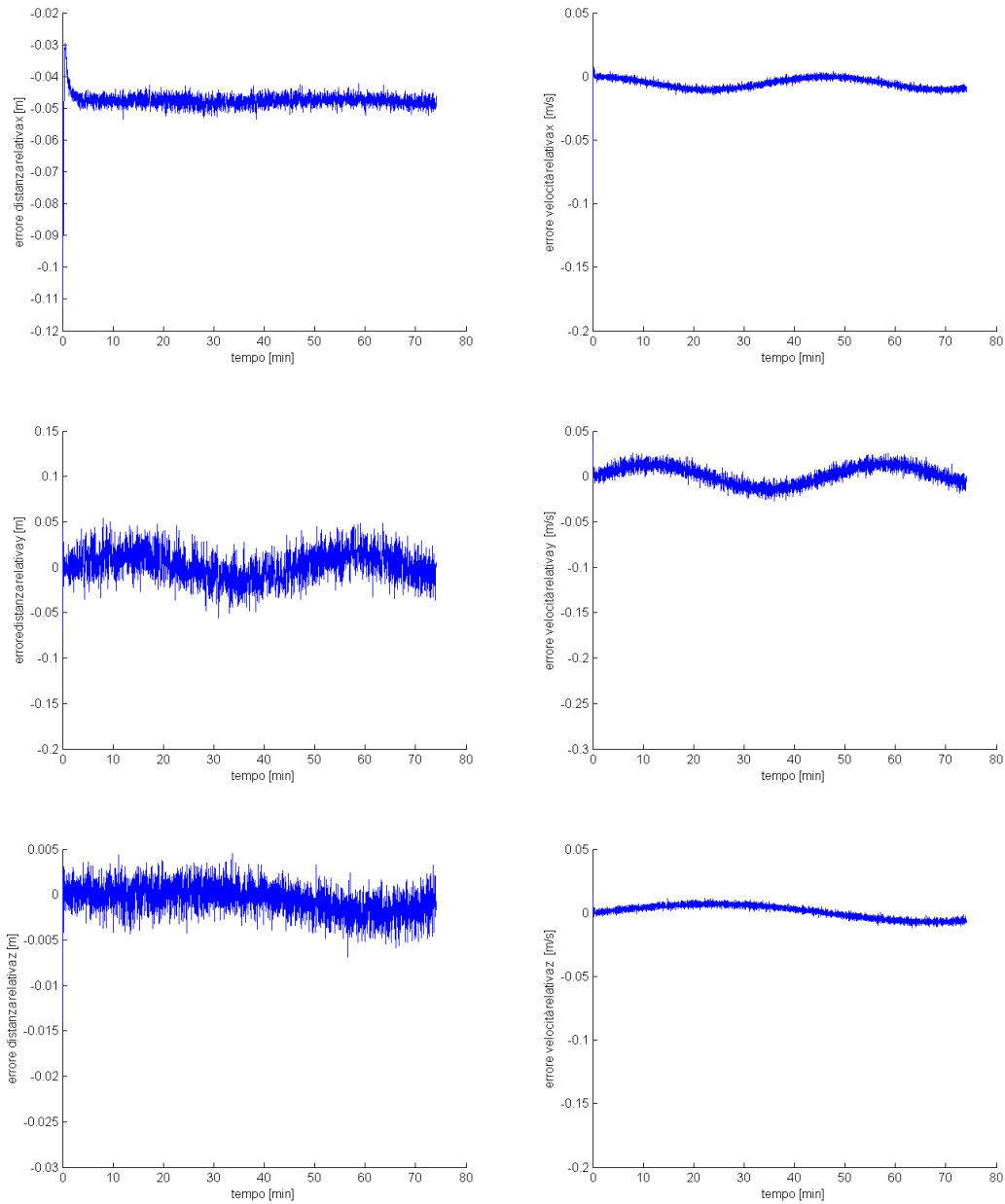


Figura 4.5: Errore nella stima della distanza e velocità relativa con il filtro di Kalman

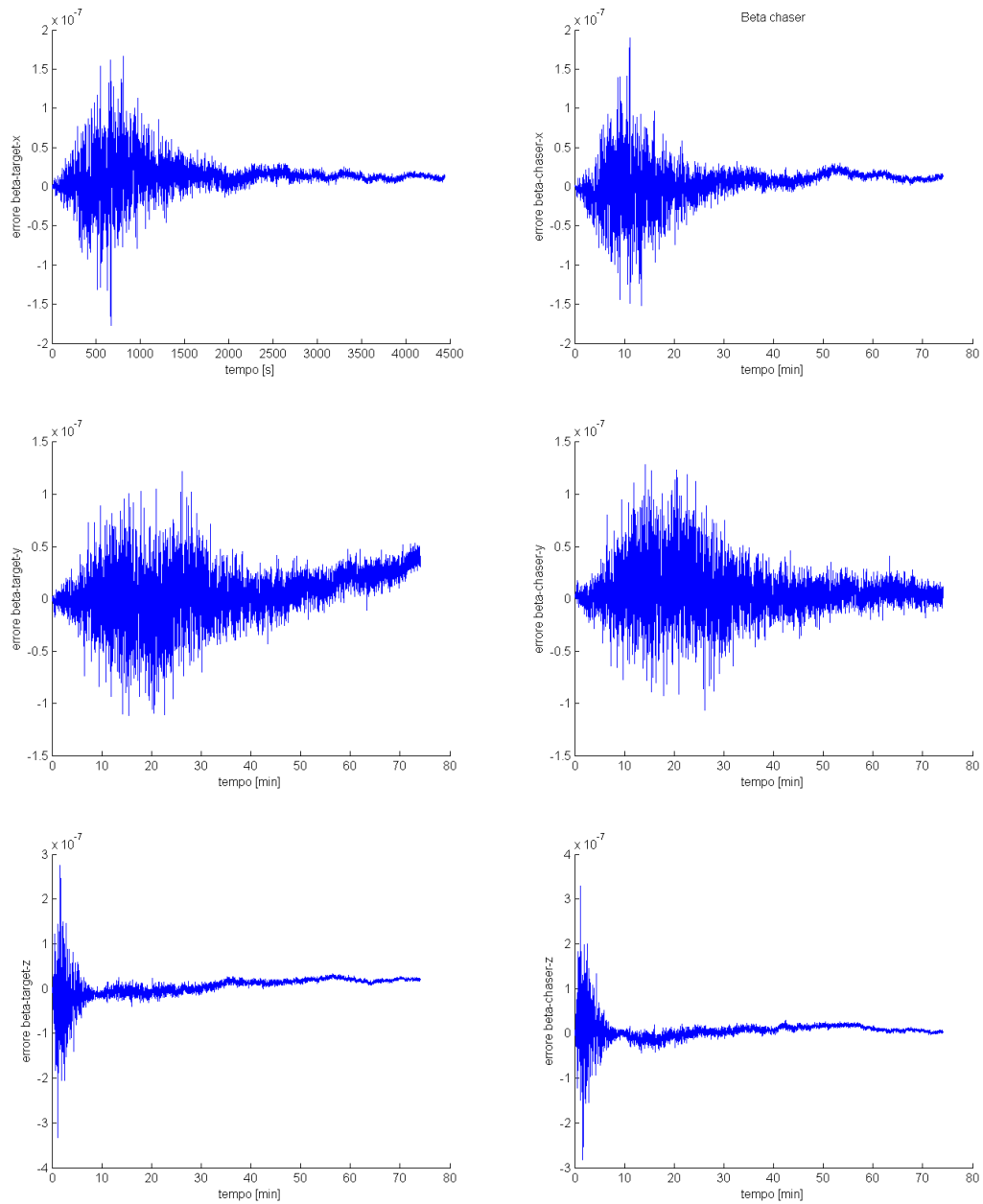


Figura 4.6: Errore nella stima dei termini di drift dei giroscopi con il filtro di Kalman

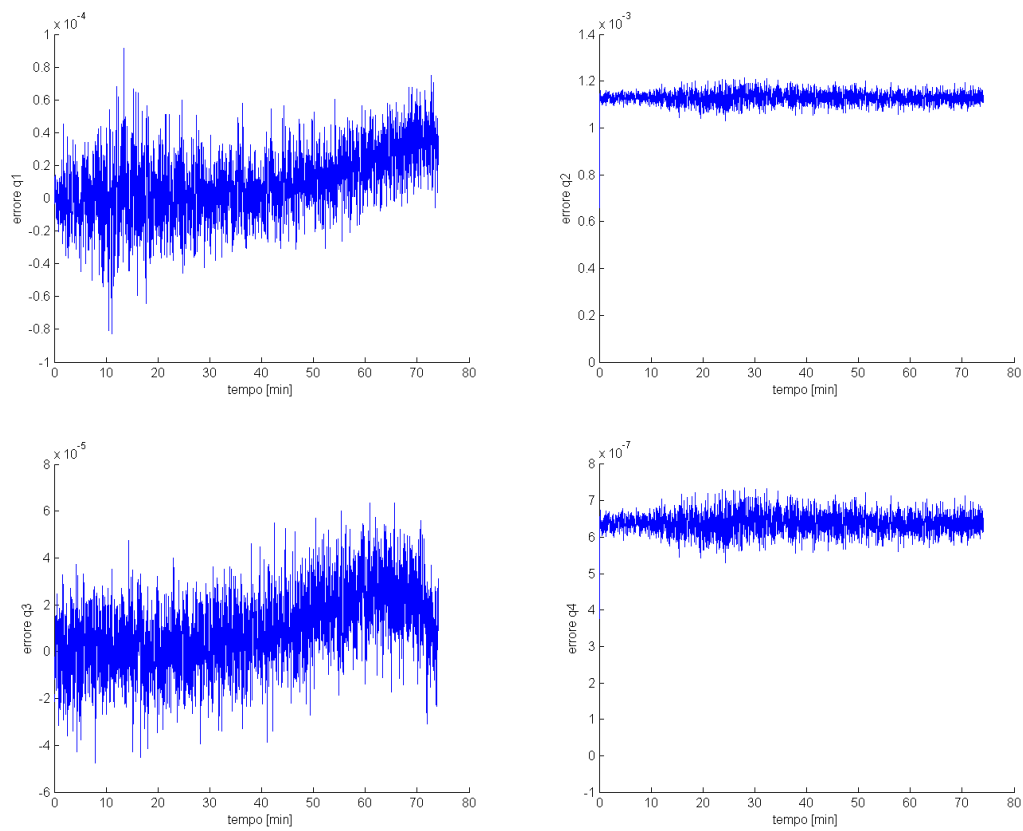


Figura 4.7: Errore nella stima dei quaternioni relativi con il filtro di Kalman

- 5.1 Controllo integrato con filtro di Kalman
- 5.2 Controllo integrato con filtro di Kalman e algoritmo EPnP
- 5.3 Controllo integrato con filtro di Kalman e algoritmo 2-stage

CAPITOLO 6 _____

_____ CONCLUSIONI E SVILUPPI FUTURI

6.1 Conclusioni

6.2 Sviluppi futuri

APPENDICE A _____
MATICI PSEUDO-LINEARI PER IL CONTROLLO

APPENDICE B

MATRICI LINEARIZZATE PER IL FILTRO DI KALMAN

Rinominando il vettore di stato orbitale con le x da 1 a 10 si ottengono i valori della matrice F

$$\begin{aligned}F_{1,4} &= 1 \\F_{2,5} &= 1 \\F_{3,6} &= 1 \\F_{4,1} &= \frac{\mu}{r_t^3}(6x/r_t + 2) + \omega^2 \\F_{4,2} &= \dot{\omega} \\F_{4,5} &= 2\omega \\F_{5,1} &= 3\frac{\mu}{r_t^4}y - \omega \\F_{5,2} &= -\frac{\mu}{r_t^3} + 3\frac{\mu}{r_t^4}x + \omega^2 \\F_{5,4} &= -2\omega \\F_{6,1} &= 3\frac{\mu}{r_t^4}z \\F_{6,3} &= -\frac{\mu}{r_t^3} + 3\frac{\mu}{r_t^4}x\end{aligned}$$

bibliografia