

## Capitolo 2

# Stabilizzazione con visual servoing

In questo capitolo descriviamo innanzitutto una procedura di localizzazione basata sulla conoscenza delle informazioni visive e delle altezze dei punti del mondo scelti come riferimento. Nel seguito, in riferimento alla legge di controllo descritta in [7], che permette di stabilizzare il veicolo su un'immagine desiderata, passando per la localizzazione in coordinate cartesiane, analizzeremo la possibilità di migliorare la localizzazione, e quindi la stabilizzazione, utilizzando un filtro di Kalman esteso e le informazioni odometriche.

### 2.1 Localizzazione con conoscenza dell'altezza delle feature

Discutiamo il problema della stima di posizione e orientamento di un veicolo autonomo, di tipo unicycle, rispetto ad un sistema di coordinate fisso, utilizzando una singola videocamera. Il metodo descritto di seguito è quello utilizzato da [7]. Consideriamo innanzitutto una terna telecamera  $\langle C \rangle$  solidale al robot con l'origine coincidente con il *pinhole* della videocamera, con l'asse  $Z_c$  diretto lungo l'asse principale della camera e l'asse  $Y_c$  perpendicolare al piano di movimento, a formare una terna sinistrorsa (vedi fig.2.1).

Dall'immagine corrente sono selezionati  $n \geq 2$  punti caratteristici (feature) con coordinate nel frame della telecamera  ${}^C P^i = {}^C [p_1, p_2, p_3]_i^T$ . Assumiamo che il moto del robot sia vincolato sul piano  ${}^C X \times {}^C Z$ : le coordinate  ${}^C p_2^i = h_i$  di ogni feature sono costanti e rappresentano l'altezza della feature rispetto al piano su cui si muove il robot.

La posizione di ogni feature nel piano immagine è descritta dal *perspective*

## 2. Stabilizzazione con visual servoing

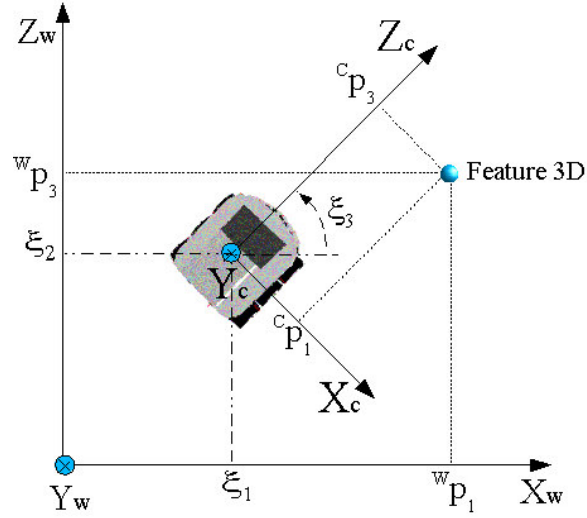


Figura 2.1: Terna fissa  $\langle W \rangle$ , terna solidale al robot  $\langle C \rangle$ , e coordinate del robot  $(\xi_1, \xi_2, \xi_3)$  in terna fissa.

projection mapping (vedi App.A.1)  $\Upsilon : \mathbb{R}^3 \rightarrow \mathbb{R}^2$

$$\Upsilon : {}^C P^i \rightarrow \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} \alpha_x \frac{{}^C p_1^i}{{}^C p_3^i} \\ \alpha_y \frac{h_i}{{}^C p_3^i} \end{bmatrix} \quad (2.1)$$

dove  $(x_i, y_i)$  sono le coordinate delle features sul piano immagine.  $\alpha_x$  e  $\alpha_y$  sono i parametri di calibrazione della telecamera che rappresentano la lunghezza della focale scalati del *pixel dimension scale factor* sull'immagine rispettivamente per l'asse  $x$  e l'asse  $y$  (vedi App.A.1.4).

Assumendo che la coordinata  ${}^C p_2^i = h_i$  sia nota, è possibile invertire la eq.(2.1) ed esprimere la posizione corrente  ${}^C P_i$  tramite la posizione delle feature sul piano immagine, ottenendo:

$$\begin{bmatrix} {}^C p_1^i \\ {}^C p_2^i \\ {}^C p_3^i \end{bmatrix} = \begin{bmatrix} \frac{\alpha_y h_i x_i}{\alpha_x y_i} \\ h_i \\ \frac{h_i \alpha_y}{y_i} \end{bmatrix} \quad (2.2)$$

Si può notare che l'assunzione che l'altezza delle feature sia nota può essere rilasciata facilmente usando osservazioni delle stesse feature da differenti posizioni e un'adeguata procedura di calibrazione, oppure usando una coppia di telecamere (stereovisione).

## 2.1 Localizzazione con conoscenza dell'altezza delle feature

Consideriamo ora una terna fissa  $\langle W \rangle$  con l'origine coincidente con quella della terna  $\langle C \rangle$  quando il robot è nella configurazione finale desiderata, e con  $X_w = Z_c$  e  $Y_w = Y_c$ .  ${}^W\xi = {}^W[\xi_1, \xi_2, \xi_3]^T \in \mathfrak{R}^2 \times S$  rappresenta la postura del robot. Più precisamente  $(\xi_1, \xi_2)$  sono le coordinate cartesiane del punto centrale dell'interasse dell'uniciclo, e  $\xi_3$  è l'orientazione dell'uniciclo tra l'asse  $Z_c$  e l'asse  $X_w$ , come rappresentato in fig.2.1. Da questa posizione iniziale viene catturata una immagine rappresentante una porzione della scena. Siano  ${}^W P_i = {}^W[p_1, p_2, p_3]_i^T$  le coordinate della  $i$ -esima feature rispetto alla terna fissa  $\langle W \rangle$ . Tutte le features sono ferme in  $\langle W \rangle$ .

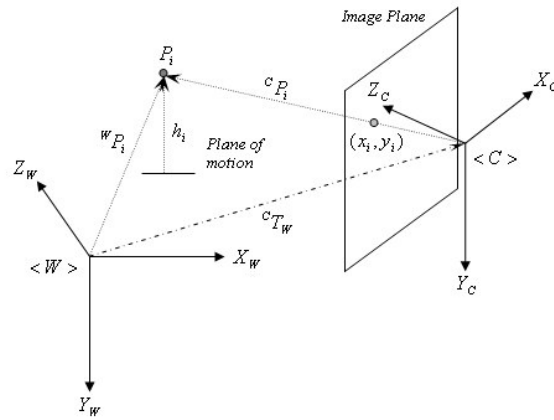


Figura 2.2: Terna fissa  $\langle W \rangle$ , terna solidale al robot  $\langle C \rangle$ , e relative coordinate.

La posizione corrente della feature  ${}^C P_i$  nella terna telecamera è in relazione con  ${}^W P_i$  nella terna fissa mediante una trasformazione rigida (vedi fig.2.2).

$$\begin{bmatrix} {}^C p_1 \\ {}^C p_2 \\ {}^C p_3 \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^W p_1 \\ {}^W p_2 \\ {}^W p_3 \\ 1 \end{bmatrix} \quad (2.3)$$

dove  $R$  rappresenta la matrice di rotazione di  $\xi_3$  tra la terna corrente solidale al robot e la terna di riferimento  $\langle W \rangle$  (è una rotazione intorno all'asse  $Y_c$ , parallelo all'asse costante  $Y_w$ ) e  $T$  la traslazione tra le origini dei sistemi di riferimento:

$$R = \begin{bmatrix} \sin \xi_3 & 0 & -\cos \xi_3 \\ 0 & 1 & 0 \\ \cos \xi_3 & 0 & \sin \xi_3 \end{bmatrix} \quad (2.4)$$

## 2. Stabilizzazione con visual servoing

---

$$T = \begin{bmatrix} \xi_2 \cos \xi_3 - \xi_1 \sin \xi_3 \\ 0 \\ -\xi_1 \cos \xi_3 - \xi_2 \sin \xi_3 \end{bmatrix} \quad (2.5)$$

Si noti che la rotazione deve subire uno sfasamento di ulteriori  $\frac{\pi}{2}$  perchè nelle condizioni finali le terne sono sfalsate.

Espandendo la eq.(2.3) e rimuovendo la seconda equazione che fornisce il valore (costante) dell'altezza della feature, si ottiene:

$$\begin{bmatrix} C p_1^i \\ C p_3^i \end{bmatrix} = \begin{bmatrix} W p_3^i & W p_1^i & 1 & 0 \\ -W p_1^i & W p_3^i & 0 & 1 \end{bmatrix} b \quad (2.6)$$

dove

$$b = \begin{bmatrix} -\cos \xi_3 \\ \sin \xi_3 \\ \xi_2 \cos \xi_3 - \xi_1 \sin \xi_3 \\ -\xi_1 \cos \xi_3 - \xi_2 \sin \xi_3 \end{bmatrix} \quad (2.7)$$

L'equazione eq.(2.6) fornisce 2 equazioni scalari non lineari in 3 incognite ( $\xi_1, \xi_2, \xi_3$ ) per ogni feature osservata sull'immagine.

A rigore, il precedente sistema con solo 2 features può fornire una soluzione per la localizzazione del veicolo, a condizione che alcune condizioni geometriche siano soddisfatte (la retta che passa per i punti corrispondenti alle features in esame non deve essere perpendicolare al piano di moto del veicolo). Ovviamente si acquisisce robustezza usando un maggior numero di features. Prendendo in considerazione  $n \geq 4$  features, le incognite correnti di posizione e orientamento  ${}^W\xi$  dell'uniciclo possono essere valutate risolvendo (nel senso dei minimi quadrati) il sistema lineare:

$$\begin{bmatrix} C p_1^1 \\ C p_3^1 \\ \vdots \\ C p_1^n \\ C p_3^n \end{bmatrix} = \begin{bmatrix} W p_3^1 & W p_1^1 & 1 & 0 \\ -W p_1^1 & W p_3^1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ W p_3^n & W p_1^n & 1 & 0 \\ -W p_1^n & W p_3^n & 0 & 1 \end{bmatrix} b \quad (2.8)$$

Il problema ammette soluzione per ogni posizione del veicolo in  $\mathbb{R}^2 \times S$ , purchè le  $n \geq 4$  features non appartengano al piano perpendicolare al piano di moto. Una volta calcolato  $b$ , i valori  $(\xi_1, \xi_2, \xi_3)$  possono essere facilmente calcolati invertendo la eq.(2.7).

## 2.2 Considerazioni sulla localizzazione

L'algoritmo di controllo [7] è di tipo PBVS (vedi par.1.2), per cui la sua precisione dipende fortemente dalla calibrazione e dalla modellizzazione della

videocamera (ad esempio, sono insidiosi gli effetti di distorsione ai bordi dell'immagine [3]). Inoltre è necessario un affidabile algoritmo di tracking.

Dall'analisi delle eq.(2.1) e eq.(2.2) si deduce che un errore minimo sull'identificazione delle coordinate di un punto sul piano immagine comporta un errore assoluto di corrispondenza del punto in 3D tanto più grande quanto più è distante il punto dalla videocamera. Nella pratica, ciò è insidioso per il controllore [7], perchè potrebbe indurre picchi indesiderati nelle velocità imposte al robot.

Per migliorare la localizzazione integreremo le informazioni visive con le misurazioni odometriche. Per far questo utilizzeremo il filtro di Kalman esteso.

E' difficile modellare con cura i processi rumorosi che si presentano nell'estrazione delle feature dall'immagine, così come è difficile modellare i processi aleatori che disturbano le misure degli encoder (uno su tutti lo slittamento delle ruote). Assumeremo per semplicità che i processi rumorosi siano compatibili con le condizioni di applicabilità dell'EKF (vedi app.B).

### 2.2.1 Quali ingressi al filtro

Facciamo una breve discussione sull'impostazione del problema del miglioramento della localizzazione. Abbiamo preso in considerazione due diversi schemi di simulazione.

#### Localizzazione esterna al filtro

Nel primo schema la localizzazione viene effettuata da una procedura esterna al filtro. Il sistema da considerare è il seguente:

$$\begin{cases} \xi_1[k+1] = \xi_1[k] + u_1[k] \cos(\xi_3[k]) + \omega_1[k] \\ \xi_2[k+1] = \xi_2[k] + u_1[k] \sin(\xi_3[k]) + \omega_2[k] \\ \xi_3[k+1] = \xi_3[k] + u_2[k] + \omega_3[k] \end{cases} \quad (2.9)$$

dove  $u_1[k]$  e  $u_2[k]$  sono gli ingressi, cioè rispettivamente lo spostamento lineare [millimetri] e lo spostamento angolare [radianti] del robot in un periodo di campionamento.  $\omega \in N(0, Q)$  è un processo aleatorio gaussiano bianco (rumore di stato) avente matrice di covarianza  $Q$ . Le misure sono:

$$\begin{aligned} z_1[k] &= \xi_1[k] + v_1[k] \\ z_2[k] &= \xi_2[k] + v_2[k] \\ z_3[k] &= \xi_3[k] + v_3[k] \end{aligned} \quad (2.10)$$

dove  $v \in N(0, R)$  è un processo aleatorio gaussiano bianco (rumore di misura) avente matrice di covarianza  $R$ . Dalle misure sul piano immagine e

## 2. Stabilizzazione con visual servoing

---

dalla conoscenza parziale della geometria delle feature si ricavava la postura del robot secondo il metodo descritto in precedenza. La localizzazione così ottenuta viene data in ingresso al filtro come misura. I dati provenienti dagli encoder vengono utilizzati dall'EKF per la predizione.

Abbiamo verificato in simulazione che questo schema di filtraggio non è in grado di migliorare in modo apprezzabile la localizzazione: ciò perchè l'EKF può utilizzare solo gli encoder per migliorare la stima della localizzazione, che arriva dall'esterno già rumorosa. Dato che l'odometria è, in genere, più rumorosa della lettura delle feature sul piano immagine, il filtro non può apportare sostanziali miglioramenti alla stima.

### Localizzazione interna al filtro

Nel secondo schema la localizzazione viene realizzata direttamente dal filtro. Per la predizione si utilizza sempre l'odometria, invece le misure sono, in questo caso, le coordinate delle feature sul piano immagine. Questa impostazione, che descriveremo dettagliatamente nel seguito, è in grado di migliorare sensibilmente la localizzazione.

## 2.3 Sistema robot-videocamera

Facciamo riferimento alla fig.2.1. Il sistema che prendiamo in considerazione (per  $n$  feature) è:

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ w \dot{p}_1^1 \\ w \dot{p}_2^1 \\ w \dot{p}_3^1 \\ \dots \\ w \dot{p}_1^n \\ w \dot{p}_2^n \\ w \dot{p}_3^n \end{bmatrix} = \begin{bmatrix} \cos \xi_3 \\ \sin \xi_3 \\ 0 \\ 0 \\ 0 \\ \dots \\ 0 \\ 0 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \dots \\ 0 \\ 0 \\ 0 \end{bmatrix} u_2 + \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \\ 0 \\ 0 \\ 0 \\ \dots \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.11)$$

Gli ingressi del sistema sono la velocità lineare  $u_1$  e la velocità angolare  $u_2$  impresse al robot.  $\omega \in N(0, Q)$  è un processo aleatorio gaussiano a media nulla avente matrice di covarianza  $Q$ . Abbiamo incertezza sulla localizzazio-

## 2.4 Implementazione del filtro

ne del robot, mentre supponiamo di avere la migliore stima possibile delle coordinate delle feature<sup>1</sup>.

Le misure a nostra disposizione sono le letture delle coordinate delle feature sul piano immagine. Le coordinate sul piano immagine sono legate alle coordinate in terna solidale al robot secondo la relazione eq.(2.1). La relazione che lega le coordinate in terna fissa a quelle in terna solidale al robot è (vedi par.2.1):

$$\begin{bmatrix} C p_1^i \\ C p_2^i \\ C p_3^i \\ 1 \end{bmatrix} = \begin{bmatrix} \sin \xi_3 & 0 & -\cos \xi_3 & \xi_2 \cos \xi_3 - \xi_1 \sin \xi_3 \\ 0 & 1 & 0 & 0 \\ \cos \xi_3 & 0 & \sin \xi_3 & -\xi_1 \cos \xi_3 - \xi_2 \sin \xi_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} W p_1^i \\ W p_2^i \\ W p_3^i \\ 1 \end{bmatrix} \quad (2.12)$$

cioè:

$$\begin{aligned} C p_1^i &= W p_1^i \sin \xi_3 - W p_3^i \cos \xi_3 - \xi_1 \sin \xi_3 + \xi_2 \cos \xi_3 \\ C p_2^i &= W p_2^i \\ C p_3^i &= W p_1^i \cos \xi_3 + W p_3^i \sin \xi_3 - \xi_1 \cos \xi_3 - \xi_2 \sin \xi_3 \end{aligned} \quad \text{con } i = 1, \dots, n \quad (2.13)$$

Possiamo quindi riscrivere la relazione prospettica eq.(2.1) utilizzando le eq.(2.13) come:

$$\begin{cases} z_1^i = h_{2i-1}(x) = \alpha_x \frac{C p_1^i}{C p_3^i} = \alpha_x \frac{W p_1^i \sin \xi_3 - W p_3^i \cos \xi_3 - \xi_1 \sin \xi_3 + \xi_2 \cos \xi_3}{W p_1^i \cos \xi_3 + W p_3^i \sin \xi_3 - \xi_1 \cos \xi_3 - \xi_2 \sin \xi_3} + v_1(t) \\ z_2^i = h_{2i}(x) = \alpha_y \frac{C p_2^i}{C p_3^i} = \alpha_y \frac{W p_2^i}{W p_1^i \cos \xi_3 + W p_3^i \sin \xi_3 - \xi_1 \cos \xi_3 - \xi_2 \sin \xi_3} + v_2(t) \end{cases} \quad (2.14)$$

con  $i = 1, \dots, n$ .  $v \in N(0, R)$  è un processo aleatorio gaussiano a media nulla avente matrice di covarianza  $R$ , rappresentante l'incertezza nella lettura sul piano immagine.

## 2.4 Implementazione del filtro

A rigore, perchè la localizzazione sia possibile, sono necessarie  $n = 2$  feature. Con  $n = 2$  o  $n = 3$  possono sussistere problemi di osservabilità dipendenti dalla posizione relativa tra il veicolo e le feature. Con  $n = 4$  non ci sono problemi purchè le feature non siano complanari. Specializziamo, quindi, lo

---

<sup>1</sup>Nella terna fissa di riferimento, con il veicolo immobile, è possibile effettuare un elevato numero di osservazioni della stessa scena. Ciò permette, utilizzando tecniche statistiche, di abbassare il rumore di stima dovuto a errate letture dal piano immagine. Ciò non è possibile con il robot in movimento.

## 2. Stabilizzazione con visual servoing

---

studio per 4 feature. Il sistema discretizzato è il seguente:

$$\left\{ \begin{array}{l} S_1[k+1] = S_1[k] + u_1[k] \cos(S_3[k]) + \omega_1[k] \\ S_2[k+1] = S_2[k] + u_1[k] \sin(S_3[k]) + \omega_2[k] \\ S_3[k+1] = S_3[k] + u_2[k] + \omega_3[k] \\ S_4[k+1] = S_4[k] \\ S_5[k+1] = S_5[k] \\ S_6[k+1] = S_6[k] \\ S_7[k+1] = S_7[k] \\ S_8[k+1] = S_8[k] \\ S_9[k+1] = S_9[k] \\ S_{10}[k+1] = S_{10}[k] \\ S_{11}[k+1] = S_{11}[k] \\ S_{12}[k+1] = S_{12}[k] \\ S_{13}[k+1] = S_{13}[k] \\ S_{14}[k+1] = S_{14}[k] \\ S_{15}[k+1] = S_{15}[k] \end{array} \right. \quad (2.15)$$

dove  $u_1[k]$  e  $u_2[k]$  sono gli ingressi e rappresentano rispettivamente lo spostamento lineare e angolare del robot nell'intervallo di campionamento.

**Oss. 2.1** *Il modello della dinamica dell'uniciclo utilizzato per la predizione nell'EKF è tanto più impreciso quanto più è veloce il robot rispetto alla frequenza di campionamento del sistema. Nel problema esaminato un limite per il tempo di campionamento è dato dal frame-rate della telecamera che dipende soprattutto dalla luminosità dall'ambiente. Nelle situazioni di interesse il frame-rate varia da 6 – 7fps fino a 30 – 32fps.*

Per quanto riguarda le misure, indicando con:

$$\begin{aligned} \beta_1 &= {}^W p_1^1 \sin \xi_3 - {}^W p_3^1 \cos \xi_3 - \xi_1 \sin \xi_3 + \xi_2 \cos \xi_3 \\ \beta_2 &= {}^W p_1^1 \cos \xi_3 + {}^W p_3^1 \sin \xi_3 - \xi_1 \cos \xi_3 - \xi_2 \sin \xi_3 \\ \beta_3 &= {}^W p_1^2 \sin \xi_3 - {}^W p_3^2 \cos \xi_3 - \xi_1 \sin \xi_3 + \xi_2 \cos \xi_3 \\ \beta_4 &= {}^W p_1^2 \cos \xi_3 + {}^W p_3^2 \sin \xi_3 - \xi_1 \cos \xi_3 - \xi_2 \sin \xi_3 \\ \beta_5 &= {}^W p_1^3 \sin \xi_3 - {}^W p_3^3 \cos \xi_3 - \xi_1 \sin \xi_3 + \xi_2 \cos \xi_3 \\ \beta_6 &= {}^W p_1^3 \cos \xi_3 + {}^W p_3^3 \sin \xi_3 - \xi_1 \cos \xi_3 - \xi_2 \sin \xi_3 \\ \beta_7 &= {}^W p_1^4 \sin \xi_3 - {}^W p_3^4 \cos \xi_3 - \xi_1 \sin \xi_3 + \xi_2 \cos \xi_3 \\ \beta_8 &= {}^W p_1^4 \cos \xi_3 + {}^W p_3^4 \sin \xi_3 - \xi_1 \cos \xi_3 - \xi_2 \sin \xi_3 \end{aligned}$$



## 2.4 Implementazione del filtro

si ha:

$$\left\{ \begin{array}{l} z_1^1[k] = \alpha_x \frac{\beta_1}{\beta_2} \\ z_2^1[k] = \alpha_y \frac{w p_2^1[k]}{\beta_2} \\ z_1^2[k] = \alpha_x \frac{\beta_3}{\beta_4} \\ z_2^2[k] = \alpha_y \frac{w p_2^2[k]}{\beta_4} \\ z_1^3[k] = \alpha_x \frac{\beta_5}{\beta_6} \\ z_2^3[k] = \alpha_y \frac{w p_2^3[k]}{\beta_6} \\ z_1^4[k] = \alpha_x \frac{\beta_5}{\beta_6} \\ z_2^4[k] = \alpha_y \frac{w p_2^4[k]}{\beta_6} \end{array} \right. \quad (2.16)$$

Osserviamo che  $\beta_1$  e  $\beta_2$  sono rispettivamente la coordinata  ${}^C p_1^1$  e la coordinata  ${}^C p_3^1$  espresse in terna fissa, e così via.

Gli Jacobiani da utilizzare nel filtro sono:

$$A = \frac{\partial f}{\partial S} = \left( \begin{array}{ccc|cccc} 1 & 0 & -\sin(x_3[k])u_1[k] & & & & \\ 0 & 1 & \cos(x_3[k])u_1[k] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 1 & & & & \\ \hline \mathbf{0} & & & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & & & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & & & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \hline \mathbf{0} & & & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{array} \right)$$

$$W = \frac{\partial f}{\partial \omega} = \left( \begin{array}{cccccc} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \right)$$

$$H = \frac{\partial h}{\partial S} = \left( \begin{array}{ccc|c} \alpha_x \frac{\beta_1 \cos \xi_3 - \beta_2 \sin \xi_3}{\beta_2^2} & \alpha_x \frac{\beta_1 \sin \xi_3 + \beta_2 \cos \xi_3}{\beta_2^2} & \alpha_x \frac{\beta_1^2 + \beta_2^2}{\beta_2^2} & -H(1, 1) \\ \alpha_y \frac{w p_2^1 \cos \xi_3}{\beta_2^2} & \alpha_y \frac{w p_2^1 \sin \xi_3}{\beta_2^2} & \alpha_y \frac{w p_2^1 \beta_1}{\beta_2^2} & -H(2, 1) \\ \alpha_x \frac{\beta_3 \cos \xi_3 - \beta_4 \sin \xi_3}{\beta_4^2} & \alpha_x \frac{\beta_3 \sin \xi_3 + \beta_4 \cos \xi_3}{\beta_4^2} & \alpha_x \frac{\beta_3^2 + \beta_4^2}{\beta_4^2} & \mathbf{0} \\ \alpha_y \frac{w p_2^2 \cos \xi_3}{\beta_4^2} & \alpha_y \frac{w p_2^2 \sin \xi_3}{\beta_4^2} & \alpha_y \frac{w p_2^2 \beta_3}{\beta_4^2} & \mathbf{0} \\ \alpha_x \frac{\beta_5 \cos \xi_3 - \beta_6 \sin \xi_3}{\beta_6^2} & \alpha_x \frac{\beta_5 \sin \xi_3 + \beta_6 \cos \xi_3}{\beta_6^2} & \alpha_x \frac{\beta_5^2 + \beta_6^2}{\beta_6^2} & \mathbf{0} \\ \alpha_y \frac{w p_2^3 \cos \xi_3}{\beta_6^2} & \alpha_y \frac{w p_2^3 \sin \xi_3}{\beta_6^2} & \alpha_y \frac{w p_2^3 \beta_5}{\beta_6^2} & \mathbf{0} \\ \alpha_x \frac{\beta_7 \cos \xi_3 - \beta_8 \sin \xi_3}{\beta_8^2} & \alpha_x \frac{\beta_7 \sin \xi_3 + \beta_8 \cos \xi_3}{\beta_8^2} & \alpha_x \frac{\beta_7^2 + \beta_8^2}{\beta_8^2} & \mathbf{0} \\ \alpha_y \frac{w p_2^4 \cos \xi_3}{\beta_8^2} & \alpha_y \frac{w p_2^4 \sin \xi_3}{\beta_8^2} & \alpha_y \frac{w p_2^4 \beta_5}{\beta_8^2} & \mathbf{0} \end{array} \right)$$

## 2. Stabilizzazione con visual servoing

$$V = \frac{\partial h}{\partial v} = \begin{pmatrix} 0 & -H(1,2) & \mathbf{o} & \mathbf{o} & \mathbf{o} \\ \frac{\alpha_y}{\beta_2} & -H(2,2) & \mathbf{o} & \mathbf{o} & \mathbf{o} \\ -H(3,1) & 0 & -H(3,2) & \mathbf{o} & \mathbf{o} \\ -H(4,1) & \frac{\alpha_y}{\beta_4} & -H(4,2) & \mathbf{o} & \mathbf{o} \\ \mathbf{o} & -H(5,1) & 0 & -H(5,2) & \mathbf{o} \\ \mathbf{o} & -H(6,1) & \frac{\alpha_y}{\beta_6} & -H(6,2) & \mathbf{o} \\ \mathbf{o} & \mathbf{o} & -H(7,1) & 0 & -H(7,2) \\ \mathbf{o} & \mathbf{o} & -H(8,1) & \frac{\alpha_y}{\beta_8} & -H(8,2) \end{pmatrix}$$

$$V = \frac{\partial h}{\partial v} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

dove  $\mathbf{I}$  è la matrice identica  $3 \times 3$ ,  $\mathbf{0}$  è la matrice nulla  $3 \times 3$  e  $\mathbf{o}$  è il vettore  $[0 \ 0 \ 0]$ .

## 2.5 Modello Simulink del sistema in esame

In figura 2.3 è riportato lo schema del sistema in esame.

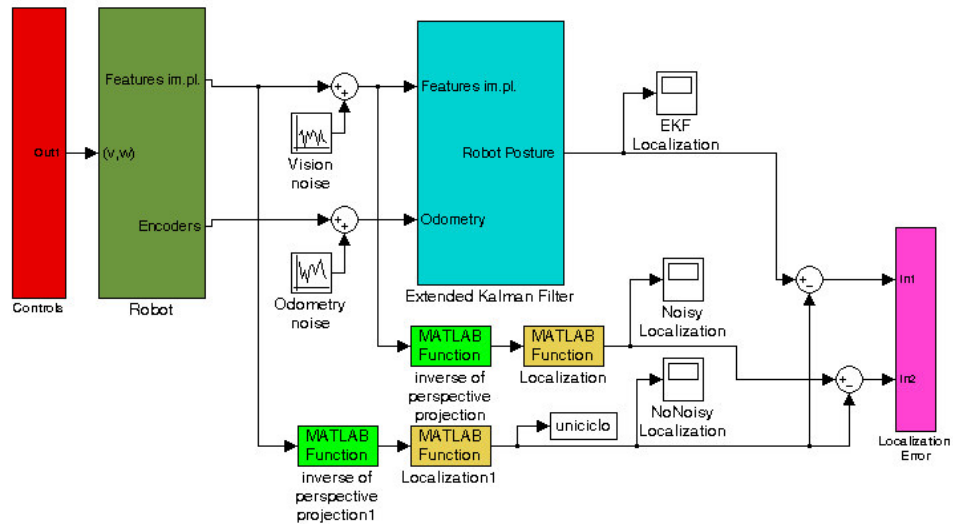


Figura 2.3: Schema di simulazione per la localizzazione tramite EKF.

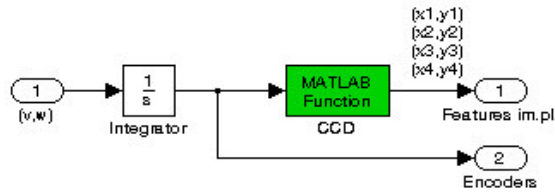


Figura 2.4: *Modello del Robot.*

### 2.5.1 Blocco *Robot*

Il blocco in questione (fig.2.4) simula il veicolo. L'integratore simula gli encoders, mentre il sottosistema *CCD* simula la dinamica delle feature sul piano immagine.

#### Blocco *CCD*

Questo blocco prende in ingresso i controlli dati al robot e dà in uscita la posizione attuale delle feature sul piano immagine.

##### Ingressi:

- controlli  $(u_1, u_2)$ ;
- posizione iniziale delle feature sul piano immagine  $\begin{pmatrix} x_{i0} \\ y_{i0} \end{pmatrix}$   
per  $i = 1 \dots 4$ ;

##### Uscite:

- posizione attuale delle feature sul piano immagine  $\begin{pmatrix} x_i \\ y_i \end{pmatrix}$   
per  $i = 1 \dots 4$ ;

##### Elaborazione:

- Jacobiano immagine  $\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} \frac{x_i y_i}{h_i \alpha_y} & \frac{\alpha_x^2 + x_i^2}{\alpha_x} \\ \frac{y_i^2}{h_i \alpha_y} & \frac{x_i y_i}{\alpha_x} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$

dove  $(u_1, u_2)$  sono i controlli (velocità lineare e angolare) dati al robot. A seconda del contesto, tempo continuo o tempo discreto, le notazioni  $u_1$  e  $u_2$  assumono significati diversi: nel primo caso rappresentano velocità ( $[mm/sec]$  e  $[rad/sec]$ ), mentre nel secondo indicano i relativi spostamenti ( $[mm]$  e  $[rad]$ ) nell'intervallo di campionamento.

## 2. Stabilizzazione con visual servoing

---

### Condizioni iniziali *CCD*

Per calcolare la posizione iniziale delle feature sul piano immagine si esegue la seguente elaborazione:

- gli ingressi sono:
  - la posizione iniziale del robot  $(\xi_{10}, \xi_{20}, \xi_{30})$ ;
  - la posizione delle feature rispetto alla terna fissa  ${}^W(P_1, P_2, P_3, P_4)$ .

- si calcola mediante la matrice di trasformazione tra la terna fissa e la terna solidale al robot:
 
$$\begin{bmatrix} {}^C p_1 \\ {}^C p_2 \\ {}^C p_3 \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^W p_1 \\ {}^W p_2 \\ {}^W p_3 \\ 1 \end{bmatrix}$$
 dove per esteso:

$$- R = \begin{bmatrix} \sin \xi_{30} & 0 & -\cos \xi_{30} \\ 0 & 1 & 0 \\ \cos \xi_{30} & 0 & \sin \xi_{30} \end{bmatrix}$$

è la matrice di rotazione tra le due terne;

$$- T = \begin{bmatrix} \xi_{20} \cos \xi_{30} - \xi_{10} \sin \xi_{30} \\ 0 \\ -\xi_{10} \cos \xi_{30} - \xi_{20} \sin \xi_{30} \end{bmatrix}$$

è il vettore di traslazione tra le due terne;

- l'uscita si ottiene mediante la relazione prospettica:
 
$$\begin{bmatrix} x_{i0} \\ y_{i0} \end{bmatrix} = \begin{bmatrix} \alpha_x \frac{{}^C p_1^i}{{}^C p_3^i} \\ \alpha_y \frac{{}^C p_2^i}{{}^C p_3^i} \end{bmatrix}.$$

### 2.5.2 Blocco *Inverse of perspective projection*

**Ingressi:**

- posizione attuale delle feature sul piano immagine  $\begin{pmatrix} x_i \\ y_i \end{pmatrix}$   
per  $i = 1 \dots 4$ ;

**Uscite:**

- posizione delle feature rispetto alla terna solidale al robot  $\begin{pmatrix} {}^C P_1 \\ {}^C P_2 \\ {}^C P_3 \\ {}^C P_4 \end{pmatrix}$ ;

**Elaborazione:**

$$\bullet \begin{bmatrix} {}^C p_1^i \\ {}^C p_2^i \\ {}^C p_3^i \end{bmatrix} = \begin{bmatrix} \frac{\alpha_y}{\alpha_x} \frac{h_i x_i}{y_i} \\ h_i \\ \frac{h_i \alpha_y}{y_i} \end{bmatrix}$$

### 2.5.3 Blocco *Localization*

Questo blocco serve per effettuare la localizzazione statica del robot a partire dalle coordinate delle features nella terna  $\langle C \rangle$ .

**Ingressi:**

- Coordinate delle features rispetto alla terna solidale al robot:  ${}^C P_i$  per  $i = 1 \dots 4$ ;

**Uscite:**

- Localizzazione del robot  $(\xi_1, \xi_2, \xi_3)$ ;

**Elaborazione:**

$$\bullet \text{ si parte da } \begin{bmatrix} {}^C p_1^1 \\ {}^C p_3^1 \\ \vdots \\ {}^C p_1^4 \\ {}^C p_3^4 \end{bmatrix} = A * b$$

$$\text{dove } A = \begin{bmatrix} W_{p_3^1} & W_{p_1^1} & 1 & 0 \\ -W_{p_1^1} & W_{p_3^1} & 0 & 1 \\ \dots & \dots & \dots & \dots \\ W_{p_3^4} & W_{p_1^4} & 1 & 0 \\ -W_{p_1^4} & W_{p_3^4} & 0 & 1 \end{bmatrix} \text{ e } b = \begin{bmatrix} -\cos \xi_3 \\ \sin \xi_3 \\ \xi_2 \cos \xi_3 - \xi_1 \sin \xi_3 \\ -\xi_1 \cos \xi_3 - \xi_2 \sin \xi_3 \end{bmatrix};$$

$$\bullet \text{ da cui si ricava } b = A^\dagger \begin{bmatrix} {}^C p_1^1 \\ {}^C p_3^1 \\ \vdots \\ {}^C p_1^4 \\ {}^C p_3^4 \end{bmatrix};$$

$$\bullet \text{ le equazioni da risolvere sono quindi: } \begin{cases} -\cos \xi_3 = b_1 \\ \sin \xi_3 = b_2 \\ \xi_2 \cos \xi_3 - \xi_1 \sin \xi_3 = b_3 \\ -\xi_1 \cos \xi_3 - \xi_2 \sin \xi_3 = b_4 \end{cases}$$

$$\bullet \text{ in particolare: } \begin{cases} \xi_1 = b_1 b_4 - b_2 b_3 \\ \xi_2 = -b_1 b_3 - b_2 b_4 \\ \xi_3 = \text{atan2}(b_2, -b_1) \end{cases}$$

## 2. Stabilizzazione con visual servoing

---

### 2.6 Simulazioni

Nei prossimi paragrafi commenteremo le simulazioni effettuate per verificare il comportamento del sistema.

I parametri sono i seguenti per tutte le simulazioni:

- Tempo di simulazione: 0-40 sec
- Step size: 0.1 sec
- Solver: Ode1 (Euler)

Scegliamo una frequenza di campionamento di  $10Hz$  per tenere conto di un basso frame-rate della telecamera (vedi oss.2.1).

I dati relativi alle features ed alle condizioni iniziali del robot sono i seguenti per tutte le simulazioni:

- $\xi_{10} = -300mm$   $\xi_{20} = -255mm$   $\xi_{30} = \pi/3rad$
- ${}^W P_1 = [80 \ 25 \ 40]mm$   ${}^W P_2 = [100 \ -20 \ 80]mm$   ${}^W P_3 = [100 \ -30 \ 15]mm$   
 ${}^W P_4 = [80 \ 35 \ -15]mm$

Al veicolo viene dato un controllo ad anello aperto che lo avvicina all'origine secondo una traiettoria sinusoidale, in modo da simulare il controllore [7].

#### 2.6.1 Simulazione 1

**Rumore sulle misure** Per ogni coordinata del punto sull'immagine si è aggiunto un rumore gaussiano di media nulla e varianza  $\sigma^2 = 4pixel$ ;

**Rumore sull'odometria**  $\sigma_l^2 = 1mm$  per lo spostamento lineare e  $\sigma_a^2 = 0.005rad$  per lo spostamento angolare;

**Matrici di covarianza :**

- $Q = \begin{bmatrix} 100 & 0 & 0 & 0 & \dots & 0 \\ 0 & 100 & 0 & 0 & \dots & 0 \\ 0 & 0 & 10 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix}$
- $R = 0.1 * eye(8)$ .

In fig.2.5 si vede la traiettoria dell'uniciclo che si dirige verso l'origine e si avvicina alle feature. Le fig.2.6, fig.2.7 e fig.2.8 rappresentano l'errore tra la localizzazione reale e la localizzazione stimata del veicolo. In particolare i grafici in grassetto rappresentano l'errore di stima commesso dall'EKF, mentre gli altri rappresentano l'errore commesso con la localizzazione in forma chiusa, per le feature  $\xi_1$ ,  $\xi_2$  e  $\xi_3$  rispettivamente.

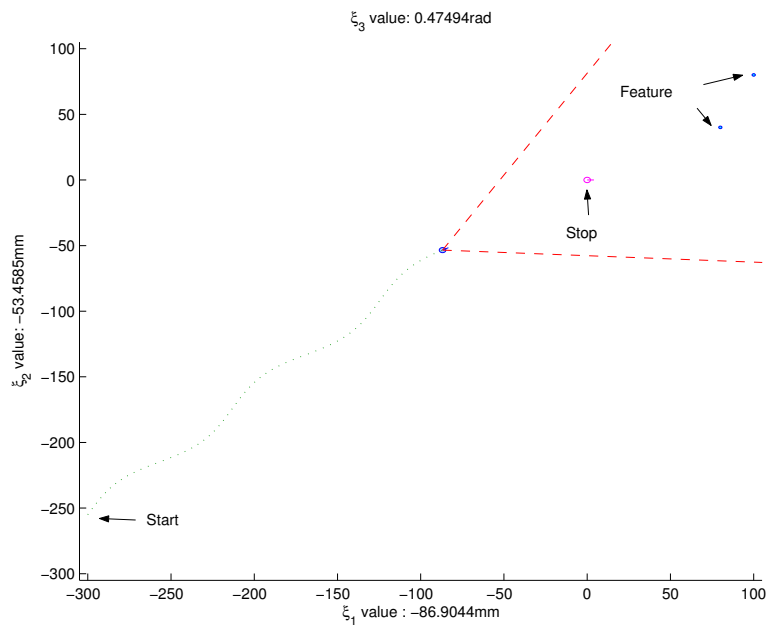


Figura 2.5: *Simulazione 1: traiettoria del veicolo.*

### Osservazioni sulla simulazione 1

- I risultati mettono in evidenza la migliore localizzazione che si ottiene facendo processare al filtro di Kalman esteso i dati provenienti dal piano immagine;
- quando il veicolo si avvicina alle feature i risultati dell'EKF diventano confrontabili con quelli provenienti dalla localizzazione in forma chiusa. Addirittura si notano dei picchi nelle stime del filtro. Ciò è dovuto a problemi di malcondizionamento dovuti alla particolare disposizione delle feature rispetto al cammino seguito dal robot. Il problema si risolve scegliendo più feature per la localizzazione.

## 2. Stabilizzazione con visual servoing

---

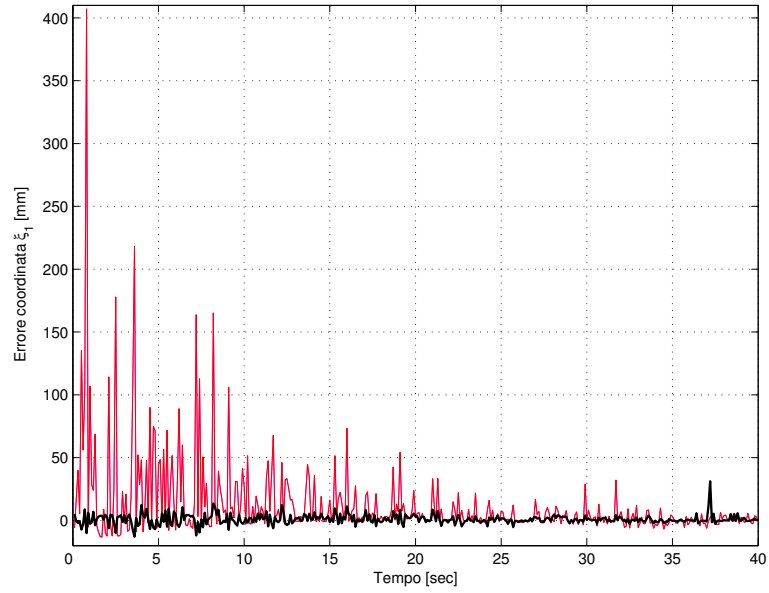


Figura 2.6: *Simulazione 1: Errore di stima sulla coordinata  $\xi_1$ .*

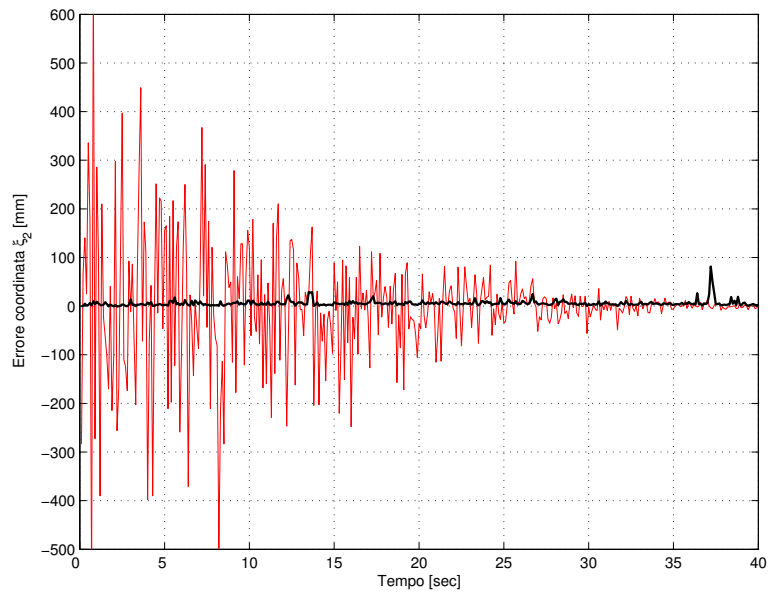


Figura 2.7: *Simulazione 1: Errore di stima sulla coordinata  $\xi_2$ .*



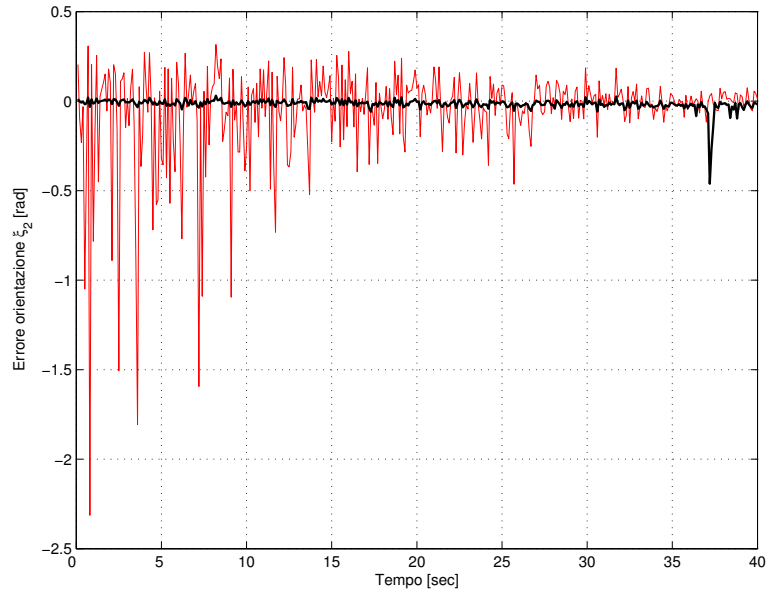


Figura 2.8: *Simulazione 1: Errore di stima sulla coordinata  $\xi_3$ .*

### 2.6.2 Simulazione 2

**Rumore sulle misure** Per ogni coordinata del punto sull'immagine si è aggiunto un rumore gaussiano di media nulla e varianza  $\sigma^2 = 1pixel$ ;

**Rumore sull'odometria**  $\sigma_l^2 = 2mm$  per lo spostamento lineare e  $\sigma_a^2 = 0.01rad$  per lo spostamento angolare;

**Matrici di covarianza :**

$$\bullet Q = \begin{bmatrix} 1000 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1000 & 0 & 0 & \dots & 0 \\ 0 & 0 & 100 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

$$\bullet R = 0.001 * eye(8).$$

Le fig.2.9, fig.2.10 e fig.2.11 rappresentano l'errore tra localizzazione reale e la localizzazione stimata del veicolo. In particolare i grafici in grassetto rappresentano l'errore di stima commesso dall'EKF, mentre gli altri rappresentano

## 2. Stabilizzazione con visual servoing

---

l'errore commesso con la localizzazione in forma chiusa, per le feature  $\xi_1$ ,  $\xi_2$  e  $\xi_3$  rispettivamente.

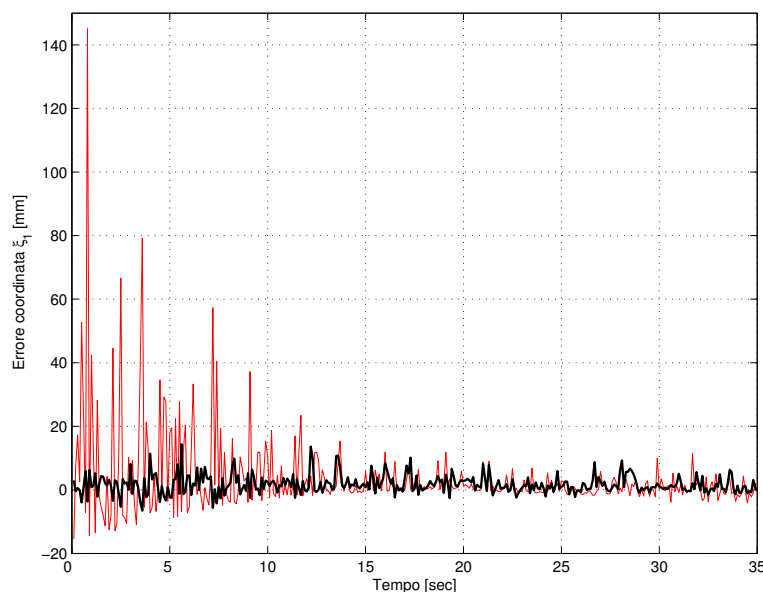


Figura 2.9: *Simulazione 2: Errore di stima sulla coordinata  $\xi_1$ .*

### Osservazioni sulla simulazione 2

- In questa simulazione, rispetto alla precedente è stato diminuito l'errore sulle misure ed amplificato l'errore sull'odometria. Abbiamo opportunamente variato le matrici  $Q$  ed  $R$  per dare meno peso all'odometria.
- Il filtro di Kalman esteso continua a dare migliori stime delle coordinate del veicolo.

## 2.7 Considerazioni sul problema

Dalle simulazioni si evince quanto segue:

- se il veicolo è relativamente lontano dalle feature, la localizzazione mediante filtro di Kalman esteso è da preferire rispetto alla localizzazione in forma chiusa ottenuta tramite le eq.(2.8) e eq.(2.7); un buon errore sull'odometria è ottimamente tollerato dal filtro;

## 2.7 Considerazioni sul problema

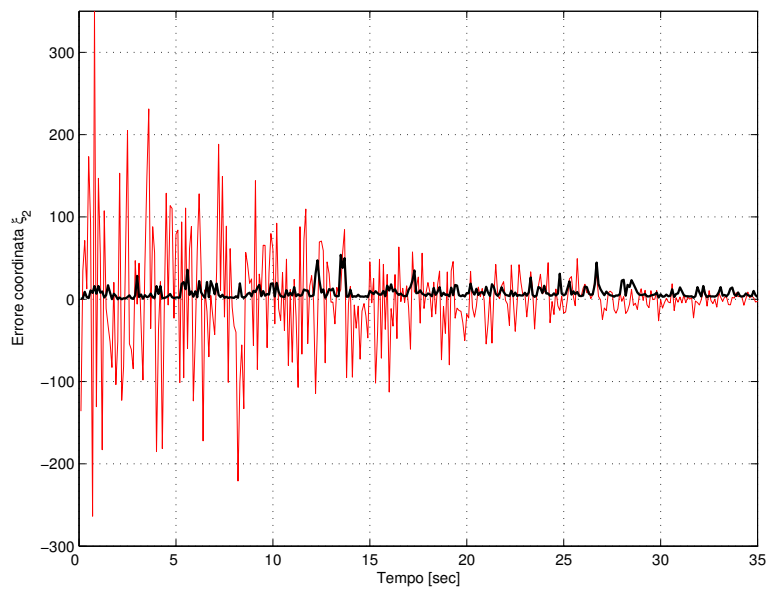


Figura 2.10: *Simulazione 2: Errore di stima sulla coordinata  $\xi_2$ .*

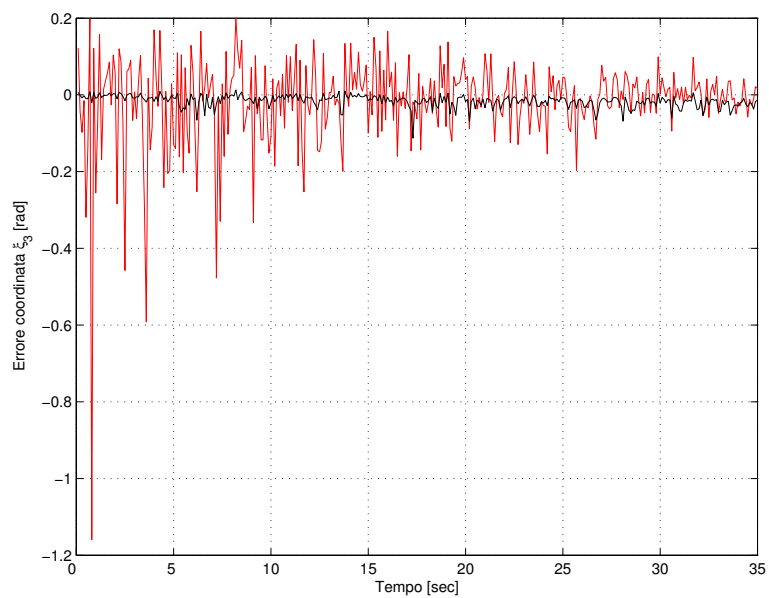


Figura 2.11: *Simulazione 2: Errore di stima sulla coordinata  $\xi_3$ .*

## 2. Stabilizzazione con visual servoing

---

- se le feature sono vicine, un piccolo errore sull'odometria, in particolare sullo spostamento angolare, comporta grossi errori nella stima. Anche variando opportunamente le matrici  $Q$  ed  $R$  non siamo riusciti ad ottenere sostanziali miglioramenti;
- nelle vicinanze delle feature abbiamo ottenuto buone stime dal filtro dando al robot piccole velocità angolari, approssimabili a zero nell'intervallo di campionamento.

Nella realtà c'è un compromesso nella scelta delle feature. La scelta è limitata ai determinati punti di cui si ha conoscenza geometrica: se questi punti sono vicini al robot, le manovre possibili sono limitate perchè le feature tendono ad uscire dal campo visivo, limitato e dipendente dal sensore utilizzato; se i punti sono lontani, un piccolo rumore sull'identificazione comporta grossi errori nella localizzazione.