UNIVERSITA' DEGLI STUDI DI PADOVA

Sede Amministrativa: Università degli Studi di Padova

Centro Interdipartimentale Studi e Attività Spaziali CISAS "G. Colombo"

SCUOLA DI DOTTORATO DI RICERCA IN: Scienze Tecnologie e Misure Spaziali (STMS)
INDIRIZZO: Misure Meccaniche per l'Ingegneria (MMI)

CICLO: XXI

METODI DI LOCALIZZAZIONE E SENSOR FUSION PER LA ROBOTICA MOBILE

LOCALIZATION METHODS AND SENSOR FUSION FOR MOBILE ROBOTS

Direttore della Scuola: Ch.mo Prof. Cesare Barbieri

Supervisore: Ch.mo Prof. Francesco Angrilli

Co-supervisore: Prof. Mariolino De Cecco

Dottorando: Enrico Marcuzzi

INDICE

Sommario

Nel corso del presente lavoro è stato sviluppato e testato un algoritmo di sensor fusion per la navigazione inerziale-odometrica. Sensori come encoder e piattaforme inerziali basate su giroscopi hanno caratteristiche molto diverse in quanto ad accuratezza di misura in relazione alla manovra attuale compiuta da un veicolo autonomo. È quindi possibile utilizzare tali sensori per ottenere una maggior accuratezza nella stima della posa, e limitare la propagazione dell'incertezza a un primo livello di sensor fusion, a cui poi va aggiunto una ulteriore misura proveniente da un sensore riferito all'ambiente, in modo da ridurre periodicamente la propagazione dell'incertezza che cresce ad ogni ciclo di acquisizione e stima della posa a causa della correlazione temporale dei parametri.

Nell'algoritmo proposto, gli encoder e il giroscopio vengono combinati tenendo in considerazione la rispettiva incertezza, stimata in funzione della manovra attuale che viene classificata a partire dai dati stessi.

Un'altra parte del lavoro ha riguardato lo sviluppo di un algoritmo di localizzazione basato sul matching di scansioni ottenute da un sensore LIDAR, per disporre di una misura di posizione riferita all'ambiente, che può essere integrata in un algoritmo di sensor fusion o in algoritmi di SLAM (Simultaneous Localization And Mapping).

Sempre basandosi sui dati provenienti dal LIDAR, è stato sviluppato un algoritmo di identificazione e localizzazione della posa di oggetti di forma nota descrivibili mediante un modello geometrico.

Di entrambi è stata condotta un'analisi di incertezza utilizzando un setup realizzato allo scopo.

È inoltre stato sviluppato un algoritmo real time di aggiramento ostacoli, denominato Reactive Simulation, che prende in considerazione la cinematica del veicolo, il modello dinamico e la sua incertezza, le condizioni iniziali, le misure dell'ambiente e l'incertezza dei sensori per calcolare la traiettoria che compierebbe il veicolo per raggiungere un target locale. Tale simulazione ha lo scopo di integrare una pianificazione dinamica della traiettoria, calcolando una predizione più accurata della traiettoria in presenza di ostacoli imprevisti e rendere più robusto l'aggiramento ostacoli. Tale algoritmo è stato implementato e ottimizzato per operare in real time su un robot mobile.

Summary

This work present a sensor fusion algorithm for a navigation system suited for Autonomous Guided Vehicles (AGV) that use two motion-sensing devices: an odometric one which uses encoders mounted on the vehicle wheels and an inertial one which uses a gyroscope. The accuracy of both sensors is estimated as a function of the actual manoeuvre being carried out, which is identified by navigation data, and then the output of both sensors are fused taking into account the accuracy ratios.

The algorithm proposed was implemented on a mock-up of a scaled industrial robot equipped with driver-steering motors, power and logical drivers. This mock-up was used to estimate experimentally the accuracy of the proposed sensor-fusion algorithm.

The discrete form of the Inertial-Odometric navigation equations is the following:

$$\begin{cases} x_{k+1}^E = x_k + \dfrac{2\pi}{n_0} n_k R \cos(\alpha_k + \alpha_0)\cos(\delta_k) \\ y_{k+1}^E = y_k + \dfrac{2\pi}{n_0} n_k R \cos(\alpha_k + \alpha_0)\sin(\delta_k) \\ \delta_{k+1}^E = \delta_k + \dfrac{2\pi}{n_0} n_k R \sin(\alpha_k + \alpha_0)\dfrac{1}{b} \end{cases} \quad (1)$$

$$\delta_{k+1}^G = \delta_k + T_C G(V_k^G) \quad (2)$$

The output of equations 1 and 2 are combined by the data-fusion algorithm in order to estimate the best guess of the two attitude increments. To formalise the data-fusion algorithm it is necessary to consider the increments of vehicle's pose obtained from the sensors:

$$\Delta x_k^E = x_{k+1}^E - x_k$$
$$\Delta y_k^E = y_{k+1}^E - y_k$$
$$\Delta \delta_k^E = \delta_{k+1}^E - \delta_k$$
$$\Delta \delta_k^G = \delta_{k+1}^G - \delta_k$$

as well as the accuracy of the attitude increments defined as: $\pm \varepsilon_{\Delta\delta}^{\eta}$, where η can be *E* (encoder) or *G* (gyroscope). The increments lead to the following equations of pose updating:

$$\begin{aligned} x_{k+1} &= x_k + \Delta x_k^E \\ y_{k+1} &= y_k + \Delta y_k^E \\ \delta_{k+1} &= \delta_k + DF(\Delta \delta_k^E, \Delta \delta_k^G) \end{aligned} \quad (3)$$

The data fusion (DF) equation which combines these increments is:

$$DF(\Delta \delta_k^E, \Delta \delta_k^G) = \frac{(\sigma_{\Delta\delta}^E)^2 \Delta \delta_k^G + (\sigma_{\Delta\delta}^G)^2 \Delta \delta_k^E}{(\sigma_{\Delta\delta}^E)^2 + (\sigma_{\Delta\delta}^G)^2}$$

In order to fuse the data, only the ratio of the two uncertainties $\sigma_{\Delta\delta}^E / \sigma_{\Delta\delta}^G$, defined as ξ_R, is needed:

$$\Delta \delta_k = \frac{\Delta \delta_k^G}{\dfrac{1}{\xi_R^2} + 1} + \frac{\Delta \delta_k^E}{\xi_R^2 + 1}$$

The uncertainty estimation at each step must be known to combine the relative system measurements (mainly affected by drift) with an absolute system. Uncertainty is expressed in terms of the covariance matrix

of the vector $X_k = [x_k , y_k , \delta_k]$. In order to achieve the above covariance estimation, equation 3 can be rewritten as:

$$X_{k+1} = X_k + \Phi(w_k)$$

where Φ is a nonlinear function of $[n_k , n_0 , R , b , \alpha_k+\alpha_0, G(V_k) , \delta_k]$ of which n_0 and b are considered constant. This last equation can be used to evaluate the standard uncertainty of the estimation X_{k+1} if the standard uncertainty of the estimates X_k and w_k are known. The uncertainty can be expressed using the covariance matrix:

$$C_{Xk+1} = C_{Xk} + \Im_{\Phi k} \cdot C_{wk} \cdot \Im_{\Phi k}^T + \Im_{\Phi k} \cdot S_{wk} \cdot I_k^T + I_k \cdot S_{wk} \cdot \Im_{\Phi k}^T$$

In order to estimate uncertainty, the covariance matrix C_{wk} has to be characterised as a function of the different manoeuvres the vehicle can make during travel. Four kinds of manoeuvres were defined for the purposes of this study as shown in next table: standing, straight path, curving, steering.

Table 1: qualitative consideration regarding different manoeuvre and sensor accuracy

Vehicle manoeuvre	Odometric accuracy: qualitative considerations	Inertial accuracy: qualitative considerations
Standing	High	Low
Straight path	High	Low
Steering with the steering angle blocked	Medium	High
Steering	Low	High

As summarized in Table 1,during standing the relative accuracy of the gyroscope is very low due to the drift of its bias; this is not the case for the encoders. The gyroscope increments were therefore not considered when the encoder driver wheel increments were zero. The same during a straight path or a path with the steering angle close to zero: vibrations affect the gyroscope's accuracy and its output is comparable to its bias. On the contrary, during steering the relative accuracy of the gyroscope is higher than the relative accuracy of the encoder-based estimation, which suffers from wheel deformations.

Several trajectories were performed to characterize ξ_R,, the ratio of the two uncertainties $\sigma_{\Delta\delta}^E / \sigma_{\Delta\delta}^G$, using the mock-up of a scaled industrial robot. In order to verify the data-fusion algorithm, a triangulation sensor based upon an optical scanner mounted on the robot was used.

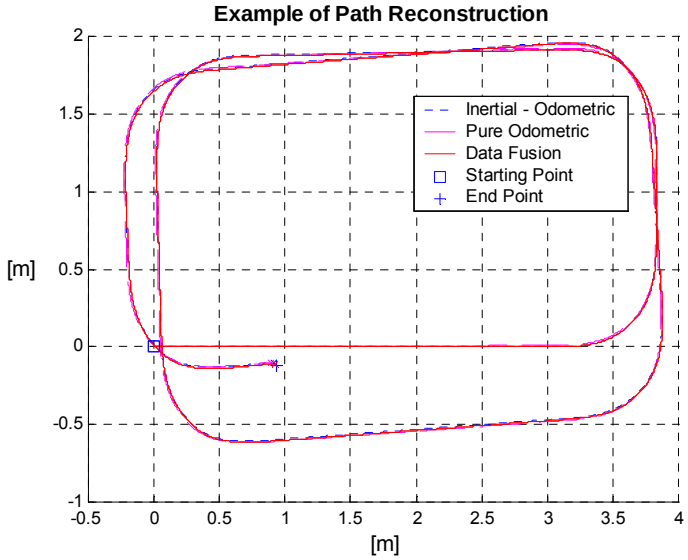


Figure 1: example of trajectories

Twenty rectangular paths, shown in Figure 1, in both clockwise and counter-clockwise directions at the velocity of 0.25 m/s were carried out as shown in next figure. These paths were followed keeping the robot load and the current fed to the motors constant (i.e. following the same trajectories in terms of steering and velocity). The initial and final positions (as shown in Figure 2) were estimated using the absolute triangulation system and then compared with the odometric, gyroscope and data-fusion estimates.

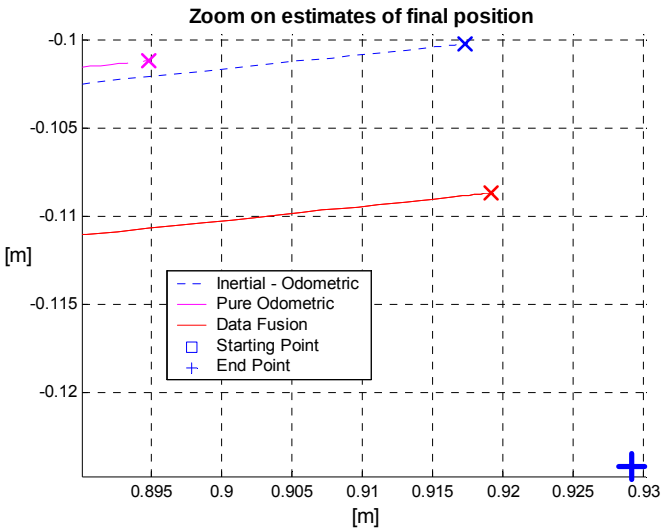


Figure 2: zoom on final position

The Figure 3 shows the uncertainty ellipses caused by the systematic effects as a function of the path. The correlation increment at the end of the path and in general on the left side of the path can be seen. The correlation increment that gives a thin ellipse (i.e. a preferred direction) is caused by the fact that the influence parameters give rise to the same direction in end-point estimation errors. This could be easily verified by computing the end-point estimation varying each influence parameter one at a time around the nominal value within twice the standard uncertainty: the end-point estimations of all the influence parameters give rise to roughly the same geometrical figure (pointing in the same direction).

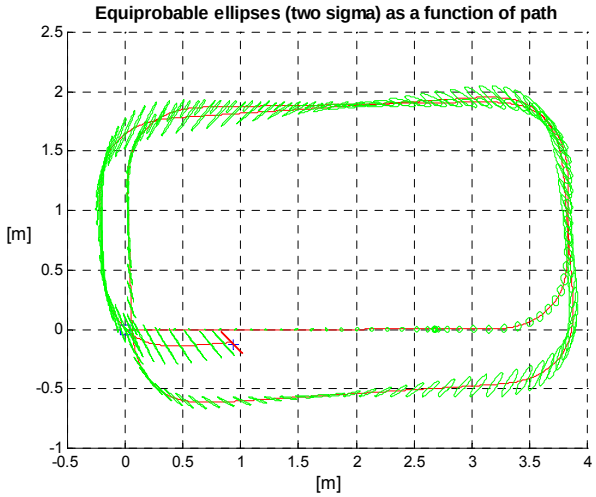


Figure 3: uncertainty ellipses plotted on trajectory

During this work it was also developed an algorithm of obstacle avoidance. The method is based on a very simple and very fast computationally path planning that has the drawback of a not negligible difference between the planned and the executed path; from this it was conceived the Reactive Simulation that allow to predict more accurately the trajectory. Reactive Simulation is computed every time that a new target position is planned during motion due to the detection of an obstacle: if the output of the simulation is the safe obstacle avoidance, the robot continues smoothly its path, otherwise it stops to avoid dangerous collisions and to plan a new safe path. The Reactive Simulation takes into account vehicle kinematics and dynamical

model and its uncertainty, initial conditions, measurement of the environment and sensors uncertainty. An advantage of this approach is that the use of a model for simulation, if integrated with an algorithm of parameters identification, permits to estimate on-line the kinematics parameters. This allows to take into account parametric variations like different diameters of wheels, inertia of masses, etc, that could affect sensibly vehicle's motion.

The algorithm was implemented on an autonomous vehicle with differential drive kinematics. A PXI (National Instruments) with an embedded real-time operating system (RTOS) was used to control the robot and implement the Reactive Simulation.

Many trajectories were tested with sudden and dynamic obstacles. It was taken into account the model and laser uncertainty and the vehicle linear and angular initial velocities corresponding to the time of laser scans arrival to perform the reactive simulation and therefore a safe robot motion. In fact, the trajectory is planned based on scans closer (with respect to the received scans) to the vehicle by a quantity proportional to the measurement uncertainty of the laser range scanner and simulation model uncertainty (see Figure 4 where the continue line is the received scan and the dashed one is the calculated closer scan).

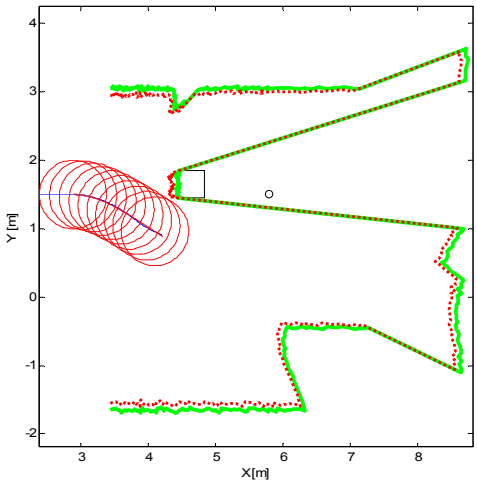


Figure 4: example of closer scan

In the Figure 5 it is shown a detail of Reactive Simulation where the continue line is the real trajectory made by vehicle after the Reactive Simulation (the dashed line).

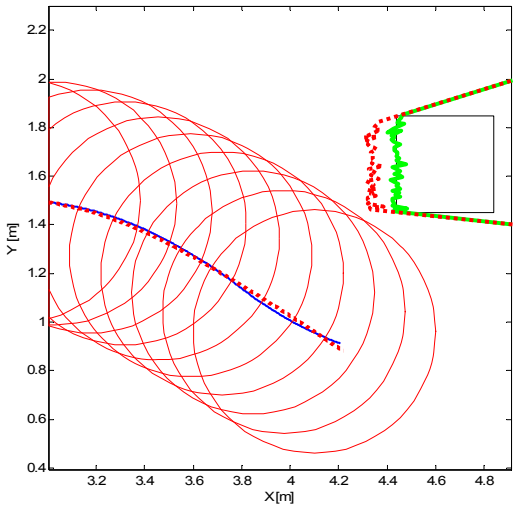


Figure 5: detail of Reactive Simulation

In this work it is also presented a localization algorithm based on LIght Detection And Ranging (LIDAR) data. A LIDAR gives a set of range measurement in a certain field of view and with a certain angular resolution, with this kind of sensor is possible for a mobile robot to "see" the environment where it moves.

This kind of localization is environment referred, so it is possible to fuse the so obtained data in a sensor fusion algorithm to minimize the propagation of uncertainty.

The algorithm tries to find the best matching between two scan, and so, knowing the pose of acquisition of the first scan, it is possible to understand the pose of acquisition of the second scan.

The algorithm was tested in post-processing, and now we're working for the real-time implementation, writing the code in C.

The scan matching is based on the Iterative Closest Point (ICP) algorithm, that is a widely used method for aligning three-dimensional point sets.

The ICP pseudocode is:

Given a model X and an object P:

- Initial transformation
- Iterative procedure to converge to local minima
  1. $\forall p \in P$ find the closest point $x \in X$
  2. Transform $P_{k+1} \leftarrow Q(P_k)$ to minimize distances between each p and x
  3. Terminate when change in the error falls below a preset threshold
- Choose the best among found solutions for different initial positions

The core of the algorithm is the minimization of a cost function:

$$\varepsilon^2(|x|) = \sum_i (E_i(a))^2 = \sum_i (w_i \varepsilon(P_{A,i} - O(a, P_{B,i})))^2$$

which is the sum of the distances of i-esim couple of points.

The minimization was investigated with two different method:

- Numerical minimization with Patternsearch algorithm (Matlab)
- Analitical minimization

Numerical and analitical minimization give the same results when the initial trasformation is near to the true trasformation, but the analitical method is faster. Numerical minimization is more robust to large displacements, where the analitical method fails.

To improve the robustness of the algorithm, it is necessary to perform some preprocessing of scans: we keep only the part of LIDAR data which are in the common field of view between the two different poses, as shown in Figure 6. The poses are estimated with some uncertainty by odometric data, and this information is also used as starting point for the minimization (if the algorithm needs it).

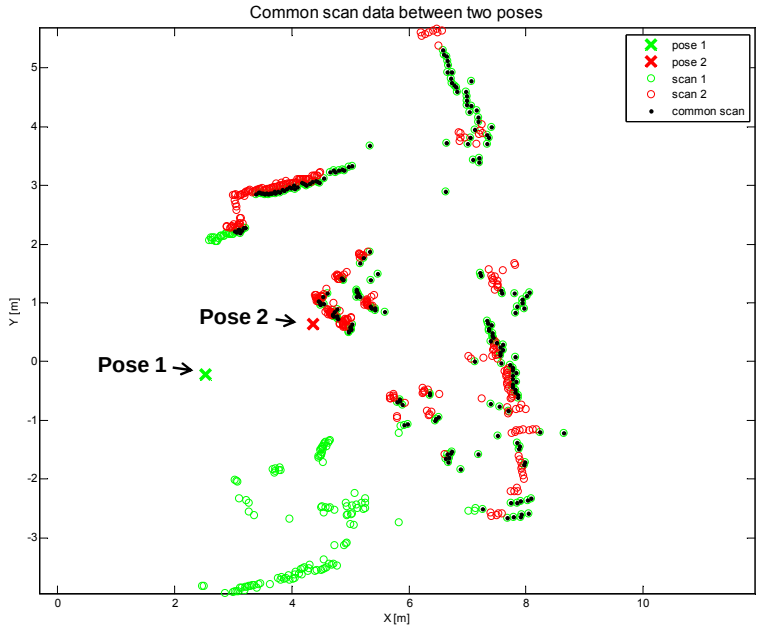


Figure 6: preprocessing of scans, keeping only the data in the common field of view between the two poses

We evaluated the effects of using other kernel because different cost functions can help the convergence also for wide displacements of the two scan.

Other common kernel are:

-Huber kernel $\varepsilon(|x|) = \log(1 + \frac{x^2}{\sigma})$

-Lorentzian kernel $\varepsilon(x) = \begin{cases} x^2 & x < \sigma \\ 2s|x| - \sigma^2 & x \geq \sigma \end{cases}$

where x is the distance between the couple of points, and σ is a experimental determined coefficient .

We also evaluated 5 different kinds of weights ω_i for the couple of points in: $E_i(a) = \omega_i \varepsilon(P_{A,i} - O(a, P_{B,i}))$.

1. Weights relative to max difference (inversely proportional to the distance): $\omega_i = 1 - \frac{d_i}{d_{\max}}$
2. Weights relative to a difference threshold: $\omega_i = \begin{cases} 1 & d_i < kd_{\max} \\ 10 & altrimenti \end{cases}$
3. Elimination of a percentual of values: ordered by distance the couples: ω_i = 1 if the i-esim couple is in *n*% of minimum distances, otherwise ω_i = 10;
4. Every couple with the same weight: ω_i = 1
5. Weights relative to the max differences (directly proportional to the distance): $\omega_i = \frac{d_i}{d_{\max}}$

where d_i is the distance of the i-esim couple, and $d_{\max}$ is the maximum distance between all the couples of points.

It was evaluated the uncertainty of the algorithm with a specific setup. The LIDAR has been put on a guide, shown in Figure 7, where it can translate and rotate.

Figure 7: calibration setup

We collect two scan, one for every pose of the LIDAR, the pose are summarized in Table 2:

Table 2: laser pose

	θ (deg)	d [cm]
Pose 0	-10	0
Pose 1	20	10

For every pose we collect 50 scans, of 15 different environment, and it was evaluated a numerical minimization, with 5 different weights and 3 different Kernel, and an analytical minimization.

In Figure 8 and Figure 9 some example of results obtained with Numerical Minimization, Weights: 1, Kernel: 2.

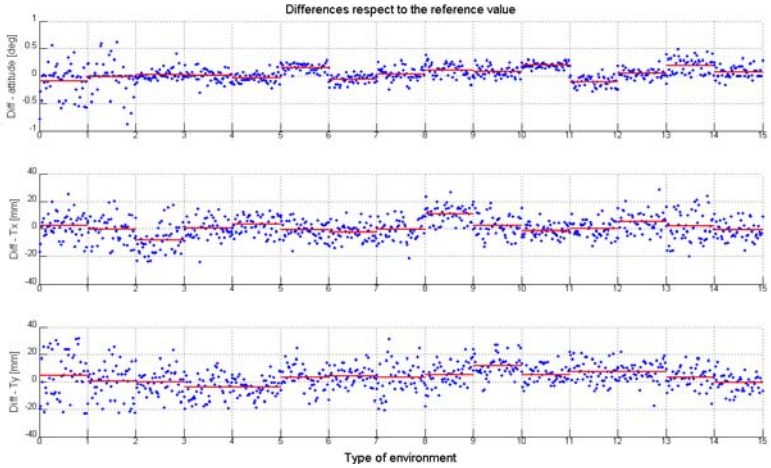


Figure 8: uncertainty analysis, differences respect to the reference value

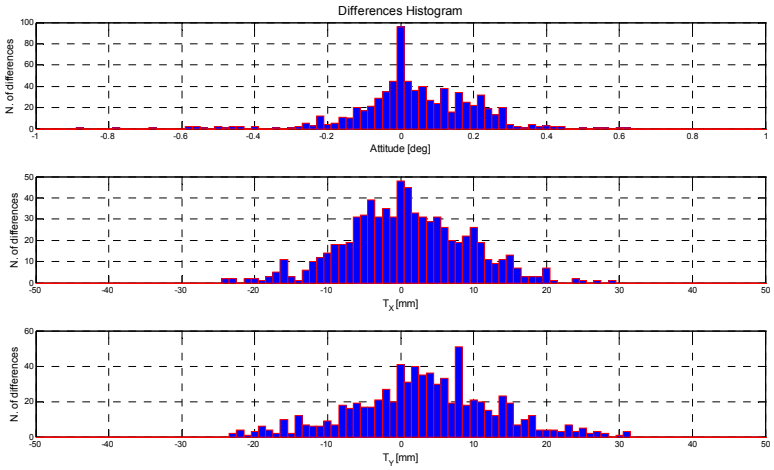


Figure 9: uncertainty analysis, histogram of differences respect to the reference value

Differences between ICP results and the reference value, that is the laser pose measured on the guide, have been evaluated. It was seen that different kernels with the same weight don't affect the result, that weights 1,2 give best and comparable results, weight 5 gives worst results.

The best result obtained (differences with the reference value) are summarized in Table 3:

Table 3: best results in uncertainty evaluation

Attitude = 0.3 deg
X = 16.5 mm
Y = 5.3 mm

Object Localization From Range Data is the problem of finding a best matching between an object model with laser scan measures of the object. In this work the object is an Europallet, see Figure 10, but it is possible to localize every kind of object which can be described with line segment.

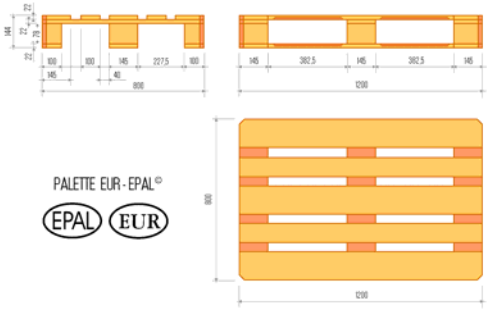


Figure 10: Europallet, the object modeled in this work

The algorithm pseudocode is:

1. Starting condition: p_0
2. Minimization of a cost function→p_{opt}
3. Rototranslation of the model with the optimized parameters
4. Simulation of LIDAR scan with the calculated pose of the model
5. Matching between real and simulated points and computation of the distance (similar to ICP)
6. If tolerance % of the distances is under a threshold: stop, otherwise $p_0 = p_{opt}$ and return to 1.

As in the previous algorithm the core is a minimization. Two kind of numerical minimization had been tried:

- Patternsearch
- Genetic algorithm

To improve the robustness of the algorithm three kind of preprocessing of data have been tried:

- Median filter to eliminate outliers, but it was seen that it can eliminate also points that belong to the object, so it was not used
- Clustering of points by distance
- Line extraction (total least square) permits to eliminate cluster of points that belong for example to wall (long lines with respect to object lines)

A validation test via simulation was performed: the object has been varied its distance (2..5 meters) and attitude (0°..180°) with respect to the laser sensor source within possible distance and attitude ranges and the estimation error and faults have been registered.

Results:

- **PATTERNSEARCH →** A 5% of convergence failures occurred in particular cases of high symmetry when the object attitude is near 0°, 180° or 90° relatively to the laser sensor mean direction.
- **GENETIC ALGORITHMS →** 99% of convergence.

With genetic algorithm, we obtained a result of mean errors and standard deviation after applying Chauvenet's criterion to eliminate outliers (not converging case) shown in Table 4.

Table 4: best results in validation test

Attitude [deg] = 0.4 +- 0.4
Trasl X [mm]= 4.0 +- 5.4
Trasl Y [mm]= 2.9 +- 3.1

Example of results are shown in Figure 11and Figure 12.

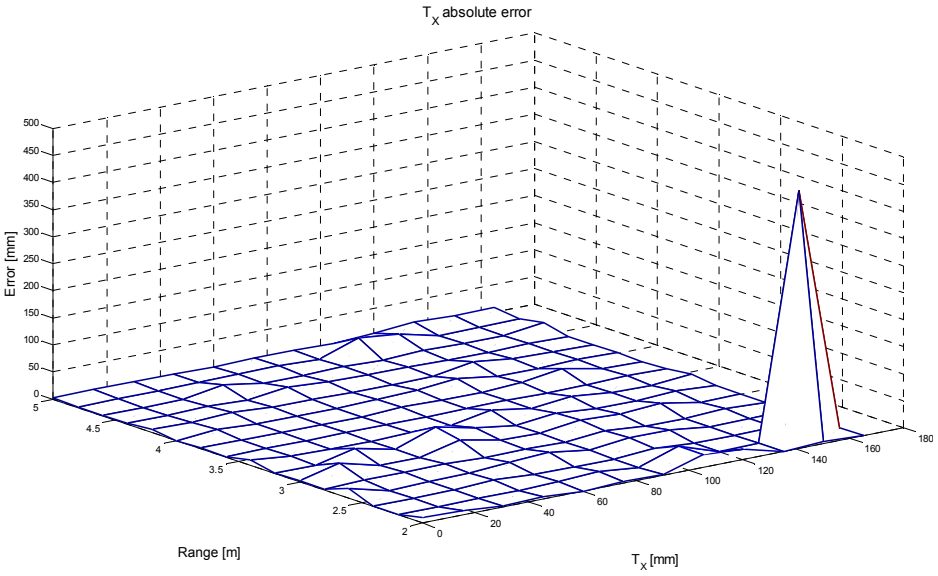


Figure 11: errors in X traslation obtained in simulation

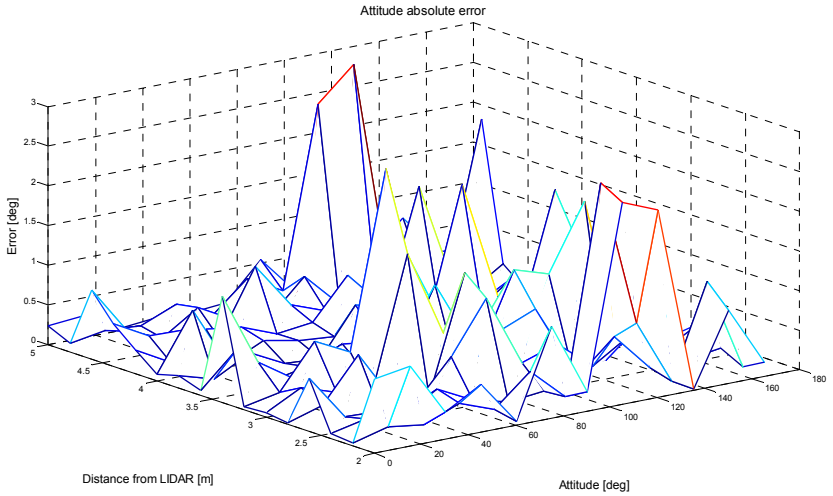


Figure 12: : errors in attitude obtained in simulation

It was made an uncertainty analysis using the same setup used for ICP uncertainty evaluation. The laser range finder was moved and these displacements were calculated from the object position estimated by the localization algorithm.

The nominal displacements and rotation are summarized in Table 5.

Table 5: nominal displacements and rotation of laser pose

d [cm]	Θ [deg]
0	0
0	-10
0	-30
0	10
0	30
10	0
10	-10
10	-30
30	0
30	-10
30	-30

For every laser pose, 20 scan had been collected and it was used the mean value to reduce noise.

We put the pallet at two different distances and also with other object around. The results are summarized in Table 6 and Table 7.

Table 6: Differences - pallet in front of LIDAR

	Mean Differences			Root Mean Square Differences		
	Θ [deg]	X [mm]	Y [mm]	Θ [deg]	X [mm]	Y [mm]
A	1.2	19	6	0.9	10	5
B	0.5	10	6	0.7	7	4
C	0.6	11	9	0.8	6	3
D	1.3	14	13	0.9	12	5
Mean	0.9	13	8	0.8	8	4

Table 7: Differences - pallet rotated with respect of LIDAR

	Mean Differences			Root Mean Square Differences		
	Θ [deg]	X [mm]	Y [mm]	Θ [deg]	X [mm]	Y [mm]
A	0.4	26.1	6	0.4	14	3.7
C	0.2	4.4	7.6	0.1	4.1	7.9
Mean	0.31	15.22	6.76	0.25	9.07	5.77

where:

- A = distance of pallet: 4 m, alone
- B = distance of pallet: 4 m,, with object around
- C = distance of pallet: 2 m, alone
- D = distance of pallet: 2 m, with object around

There's no great difference between the pallet alone or with object around, but in the case of 0° angle of the object and with other neighboring objects, the localization process was not robust due to bad incidence angle and too few points on the object. Differences to be noted are in the number of iteration of the algorithm to converge at the expected solution: with the pallet in front of the pallet and with other object around the number of iteration is greater than in the other cases.

Introduzione

La robotica è la scienza di percepire e manipolare il mondo fisico attraverso dei dispositivi controllati da computer. Si occupa cioè di sviluppare delle metodologie che permettano ad una macchina (robot), equipaggiata con opportuni sensori atti a percepire l'ambiente circostante e attuatori per interagire con esso, di eseguire dei compiti specifici. È una disciplina relativamente nuova, ma che risponde in realtà ad un antico desiderio e bisogno dell'uomo: quello di costruire delle macchine che lo liberino dai compiti più faticosi, noiosi o pericolosi. Pur essendo una branca dell'ingegneria è una scienza multidisciplinare poiché in essa confluiscono studi di molte discipline sia di natura umanistica che scientifica.

Esempi di sistemi robotici che sono impiegati con successo vanno dalle piattaforme mobili per l'esplorazione spaziale , ai bracci robotici per l'assemblaggio o la movimentazione di pezzi in linea nelle industrie, a macchine che si muovono da sole, ai manipolatori che assistono i chirurghi in particolari operazioni.

Una particolare classe di robot che ormai trova largo utilizzo in industrie, porti, ospedali, musei è quella dei veicoli a guida autonoma, o AGV (Autonomous Guided Vehicles) come vengono comunemente chiamati.

Le domande a cui un tale robot deve essere in grado di rispondere in maniera autonoma sono nella maggior parte dei casi: "dove mi trovo?", e quindi il problema della localizzazione del sistema rispetto all'ambiente, "dove devo andare e cosa devo fare?", cioè la pianificazione della traiettoria e più in generale dei compiti, e "come ci vado?", che si traduce negli algoritmi di inseguimento della traiettoria, di controllo del moto e delle varie azioni. In particolari applicazioni ci possono essere integrazioni con interventi umani, che formulano decisioni e compiono azioni in cooperazione col sistema.

Per compiere questi compiti, il robot deve essere in grado di interagire con l'enorme incertezza che esiste nel mondo fisico a causa di una serie di cause.

La prima e più importante è che l'ambiente in cui il robot si trova ad operare è molto spesso per sua natura non predicibile, in particolar modo l'incertezza è molto elevata quando i robot si trovino ad operare in presenza di umani.

Una ulteriore fonte di incertezza sono i sensori stessi, con cui il sistema acquisisce informazioni sull'ambiente. Tali informazioni sono per forza limitate dalle caratteristiche stesse dei sensori, in quanto a risoluzione, a massimo range di misura, a limiti in relazione alle condizioni di lavoro. Inoltre, i sensori sono soggetti al rumore, che degrada la qualità del segnale, e possono rompersi.

Altre fonti di incertezza sono gli attuatori che, per quanto vengano controllati da opportuni hardware e software, sono in parte non predicibili o poco accurati, (di solito in maniera inversamente proporzionale al loro costo).

Il software è un'altra causa di incertezza dal momento che i modelli interni del sistema sono per forza di cose approssimati, sono astrazioni del mondo reale. Inoltre spesso nei sistemi real time una causa di incertezza è l'approssimazione introdotta dagli algoritmi in sé, che possono gestire una quantità di informazioni limitata per soddisfare ai requisiti temporali.

Da queste considerazioni si capisce l'importanza di trattare l'incertezza di un sistema robotico con un approccio probabilistico, cioè rappresentando le informazioni con distribuzioni di probabilità e associando quindi alle scelte compiute il valore, l'intervallo e il livello di confidenza.

Il sensor fusion è il processo che combina, di solito con approccio probabilistico o con teorie alternative, le informazioni provenienti da un certo numero di diverse sorgenti per arrivare a una descrizione completa e robusta di un sistema o un processo di interesse. Nel campo della robotica e in particolare della navigazione autonoma, argomento di interesse del presente lavoro di tesi, il sensor fusion acquisisce una rilevante importanza in quanto i sistemi sono equipaggiati con una ridondanza di trasduttori in maniera tale da sfruttare i vantaggi di ogni tipologia di sensore integrando tra loro le varie misure.

Il problema della localizzazione di un sistema AGV (Autonomous Guided Vehicles) rispetto all'ambiente in cui si trova ad operare è di importanza fondamentale, in quanto le varie decisioni da prendere potrebbero essere inficiate da una stima della posizione errata, o le azioni portate a termine con esito negativo quando non pericoloso. Molto spesso infatti tali robot assolvono a compiti delicati e operano in presenza di umani, da cui l'importanza di poter disporre di dati completi e robusti su cui basare algoritmi di decisione a loro volta robusti. Per conoscere con elevata accuratezza e precisione una misura di posizione e assetto (*posa*), i sistemi di misura tipicamente utilizzati appartengono a tipologie differenti, complementari nelle caratteristiche di accuratezza e frequenza di aggiornamento. I sistemi di misura incrementali o di dead-reckoning, come ad esempio encoders, giroscopi, ultrasuoni, etc, hanno il vantaggio di essere auto-contenuti all'interno del robot, di essere relativamente semplici da usare e garantire un'elevata frequenza di

aggiornamento della misura. D'altro canto, poiché integrano incrementi relativi ad ogni ciclo di acquisizione, l'incertezza nella stima della posa cresce con il trascorrere del tempo. I sistemi di navigazione riferiti all'ambiente fanno uso di riferimenti esterni fissi posizionati nell'ambiente in cui il robot opera. Da qui il vantaggio rispetto ai sistemi di misura incrementali: poichè la misura di posizione e assetto è effettuata rispetto ai riferimenti fissi nell'ambiente, l'incertezza resta comunque limitata ed è garantita la ripetibilità della misura rispetto all'ambiente. Questi sistemi hanno una frequenza di aggiornamento più bassa rispetto a quelli incrementali e funzionano solo in determinate condizioni di visibilità dei riferimenti artificiali.

In base a queste considerazioni, è intuitivo comprendere come per ottenere una misura robusta e accurata si possano combinare sistemi di navigazione incrementali e assoluti, garantendo in questo modo un'elevata frequenza di aggiornamento della misura e un'incertezza limitata nella stima della posa (J. Borenstein, L Feng, 1996), (Adam, 1999).

Il problema più importante nella fusione dei dati è lo sviluppo di appropriati modelli che rappresentino l'incertezza associata sia allo stato che alle osservazioni del sistema. La maggior parte delle tecniche di sensor fusion impiega le teoria della probabilità, tipicamente il teorema di Bayes e il filtro di Kalman, per descrivere e manipolare l'incertezza.

Allo scopo di ottenere buone prestazioni dall'algoritmo di sensor fusion è fondamentale una stima accurata in *real-time* dell'incertezza sia delle misure di tipo incrementale che di quelle riferite all'ambiente. Mentre per i sistemi riferiti all'ambiente l'incertezza è esprimibile direttamente in funzione di alcuni parametri noti (il numero di targets visibili, la velocità lineare e angolare del robot etc), per quelli di tipo incrementale l'incertezza è una funzione di alcuni parametri cinematici, dello stesso modello cinematico e della storia precedente dello stato (in particolare della storia di assetto della traiettoria). Inoltre è necessario considerare la correlazione dell'incertezza dei parametri cinematici. Per i suddetti motivi la stima di incertezza sulla posa non può essere basata solo su metodi di combinazione dell'incertezza standard, ma deve essere una funzione dipendente dalla stato e ricorsiva.

Nel corso del presente lavoro è stato sviluppato e testato un algoritmo di sensor fusion per la navigazione inerziale-odometrica che tiene conto dell'incertezza in funzione della manovra attuale del veicolo. Il vantaggio di tale tecnica è che permette di sfruttare le migliori caratteristiche dei sensori in base all'incertezza di ognuno in relazione al tipo di manovra e di limitare quindi la propagazione dell'incertezza.

I Lidar sono tra i sensori più utilizzati in robotica mobile per la conoscenza del mondo esterno. I dati forniti da tale tipologia di sensori sono tipicamente utilizzati per evitare ostacoli, per mappare l'ambiente e localizzarsi rispetto ad esso. Inoltre sono utilizzati come dispositivi di sicurezza in molti sistemi automatici per la produzione o la movimentazione dei beni. Utilizzando i dati ottenuti da tale sensore è possibile localizzare la posizione di un robot mobile rispetto all'ambiente e individuare e localizzare la posa di oggetti descrivibili da modelli, mediante il matching di scansioni.

Tali abilità possono risultare estremamente utili in quanto conferiscono una notevole versatilità a un robot mobile, per esempio in applicazioni dove non sia possibile utilizzare dei sistemi di misura della posizione riferiti all'ambiente, mentre per quanto riguarda la localizzazione di oggetti, ad esempio per la movimentazione di merci in quei casi in cui i pallet vengano disposti in modo disordinato durante operazioni di carico e scarico e un transpallet autonomo si troverebbe impossibilitato a compiere le sue missioni.

Nel corso del presente lavoro sono stati sviluppati e testati diversi algoritmi per la ricerca di oggetti, in particolare di europallet, e per la localizzazione di un robot mediante Lidar 2D. E' stata inoltre effettuata una analisi dell'incertezza degli algoritmi sviluppati mediante un setup sviluppato allo scopo.

In relazione alle caratteristiche dei sensori e al tipo di misura che si intende effettuare, quale quella di un veicolo in movimento, è importante che non vi siano slittamenti delle ruote o movimenti bruschi di curvatura poiché in questo modo si deteriorerebbe la misura proveniente da vari sensori (encoder, giroscopi, laser a triangolazione). Per questo motivo, altri aspetti, quali la pianificazione e il controllo della traiettoria, così come la capacità di operare in real-time l'aggiramento di ostacoli imprevisti o in movimento, risultano di importanza fondamentale anche per l'efficienza e la robustezza dell'algoritmo di sensor fusion, e di solito vengono considerati a un livello gerarchico più alto rispetto alla stima della posa, in quanto si basano comunque in primo luogo sulla stima della posa attuale del veicolo.

Questi ultimi aspetti a loro volta devono tenere conto dell'incertezza nella stima della posa e nella conoscenza dell'ambiente, soprattutto nel caso di navigazione in ambienti non strutturati, angusti o ad alto rischio, e per operare con robustezza l'aggiramento degli ostacoli. I sensori che partecipano alla conoscenza dell'ambiente circostante un AGV sono di solito laser lidar, ultrasuoni, telecamere, etc, che peggiorano l'accuratezza all'aumentare della velocità e della distanza dell'oggetto misurato. Di queste caratteristiche va tenuto conto nello sviluppare algoritmi che guidino un AGV in condizioni di assoluta sicurezza.

Durante il lavoro sono stati quindi sviluppati algoritmi di pianificazione della traiettoria e di aggiramento ostacoli che tenessero conto dell'incertezza nella stima della posa e nelle misure dei sensori. In particolare è stata sviluppata la Reactive Simulation, un algoritmo real time di aggiramento ostacoli, che prende in considerazione la cinematica del veicolo, il modello dinamico e la sua incertezza, le condizioni iniziali, le misure dell'ambiente e l'incertezza dei sensori per calcolare in real-time una simulazione della traiettoria che compierebbe il veicolo per raggiungere un target locale.

La Reactive Simulation viene calcolata ogni volta che un nuovo target locale è pianificato durante il moto in risposta al rilevamento di un ostacolo: se l'output della simulazione è un aggiramento ostacoli sicuro, il veicolo continua a inseguire la nuova traiettoria impostata, in caso contrario si ferma per evitare pericolose collisioni e pianificare un nuovo percorso sicuro.

La presentazione del lavoro è organizzata come segue: nel capitolo 1 vengono descritti i fondamenti del sensor fusion, approfondendo la quantificazione dell'incertezza con modelli di tipo probabilistico e fornendo brevi cenni a modelli alternativi all'approccio probabilistico. Nel capitolo 2 viene proposto un algoritmo di sensor fusion per AGV che combina sensori diversi tenendo in considerazione la rispettiva incertezza stimata in funzione della manovra attuale compiuta dal veicolo. Nel capitolo 3 viene descritto un algoritmo di localizzazione basato sul matching di scansioni provenienti dal Lidar e viene presentata un'analisi di incertezza dell'algoritmo. Nel capitolo 4 viene presentato un algoritmo di ricerca di oggetti di forma qualsiasi purché descrivibile tramite segmenti, basato sul matching di scansioni del Lidar e viene presentata un'analisi di incertezza dell'algoritmo. Nel capitolo 5 viene proposto un algoritmo real time di aggiramento ostacoli, la Reactive Simulation.

Capitolo 1

Tecniche di sensor fusion

1.1 Fondamenti del “sensor fusion”

Il sensor fusion è il processo che combina le informazioni provenienti da un certo numero di sorgenti differenti per fornire una descrizione completa e robusta di un insieme di variabili di interesse. Il sensor fusion è di particolare utilità in ogni applicazione in cui molte misure devono essere combinate assieme per poter ottenere informazioni di qualità e integrità atte allo scopo dell’applicazione. Tecniche di sensor fusion sono utilizzate in molti sistemi industriali, militari, di monitoraggio, di sorveglianza civile, in processi di controllo e in sistemi informatici. Vengono di seguito descritti i fondamenti su cui si basa il sensor fusion secondo la quantificazione dell’incertezza con modelli di tipo probabilistico. Si fanno inoltre brevi cenni sui modelli alternativi all’approccio probabilistico.

1.1.1 Metodi probabilistici

Ogni processo di misura implica che nell’informazione ottenuta da un sensore sia contenuto un certo livello di incertezza e questa, ai fini pratici di utilizzo della misura, e a maggior ragione per il sensor fusion, va in qualche modo rappresentata analiticamente e quantificata. Importanti indicazioni vengono fornite in tal senso dalla norma ISO che definisce la misura come un insieme di elementi: un valore numerico, un intervallo di valori ad esso associato, un’unità di misura e un livello di confidenza che definisce la probabilità che la misura si trovi all’interno dell’intervallo indicato. Però, per combinare più misure/informazioni, ovvero per il sensor fusion, occorre anche la funzione densità di probabilità associata alla misura. Questa funzione è presa in considerazione prima della operazione di integrazione della densità di probabilità, da cui risulta un semplice intervallo, quello che viene poi associato alla misura insieme all’intervallo di confidenza. A valle di questa operazione, perciò, si perdono le informazioni relative a come la probabilità si distribuisce attorno al valore definito. Ad esempio , se si ipotizza una funzione densità di probabilità di tipo Gaussiano, la misura sarà costituita dal valor medio e da un intervallo di valori centrato attorno alla media μ e di ampiezza pari a un certo coefficiente moltiplicato per la deviazione standard σ della distribuzione. Se si vuole associare alla misura un livello di confidenza del 95% (come accade usualmente), l’intervallo avrà i propri estremi a $\pm 2\sigma$ dal valor medio. Una volta definiti gli elementi della misura, verrà omesso il tipo di distribuzione di probabilità utilizzato, informazione che resta fondamentale in un’applicazione di sensor/data fusion.

Sebbene vi siano molti metodi per rappresentare l’incertezza, a tutt’oggi i modelli più utilizzati sono quelli probabilistici. La loro vasta diffusione è giustificata dalla relativa semplicità nell’utilizzo rispetto ad altri metodi e dalla loro caratteristica di essere intuitivi per l’impiego nella fusione di informazione e nella logica decisionale (che viene a monte della fusione).

Non è detto, tuttavia, che la probabilità sia sempre il miglior modo per rappresentare l’incertezza, difatti la realtà suggerisce a volte l’utilizzo di metodi alternativi per ottenere un’informazione che sia più completa e aderente alla misura e alla sua incertezza.

1.1.2 Modello di stato ed osservazione

I concetti di stato e di osservazione di un sistema sono alla base dello sviluppo di varie teorie, tra cui, di nostro interesse, la teoria dei controlli e la teoria del data fusion. Una volta definito un modello fisico della natura di un sistema, più o meno vicino alla realtà, il concetto di *stato* di un sistema è associato al concetto di valore vero delle variabili di interesse di quel sistema, ossia dei valori che possono assumere le grandezze fisiche o le proprietà in genere che definiscono il modello del sistema. Il valore vero di uno stato è anche detto *misurando*, ossia grandezza fisica il cui valore è oggetto di indagine nel processo di misurazione. Ad esempio, si vuole costruire un modello cinematico di un veicolo. Si parte dall’assunzione che esso sia un corpo rigido, si sceglie un punto solidale al veicolo e ad esso si associa un asse di riferimento, un vettore

posizione (nel piano, ad esempio, due coordinate cartesiane e un angolo di assetto) e un vettore velocità (due velocità lineari e una velocità angolare). Lo stato sarà costituito dal vettore posizione e dal vettore velocità. Nel modello si descrive come evolve nel tempo (la derivata dello stato rispetto al tempo) il sistema in funzione del suo stesso stato e di altri possibili parametri.

Forma standard del modello lineare di stato di un sistema:

$$\dot{x}(t) = F(t)x(t) + B(t)u(t) + G(t)v(t) \tag{1.1}$$

Dove

$x(t) \in \Re^n$ è il vettore di stato, ossia il vettore di interesse,

$u(t) \in \Re^s$ è noto come vettore di controllo,

$v(t) \in \Re^q$ è una variabile casuale (detta rumore) da cui dipende l'incertezza nell'evoluzione dello stato nel tempo (nel caso trattato in precedenza potrebbe essere un errore di accelerazione, cioè della derivata della variabile di stato velocità),

$F(t)$ è la matrice di stato (o di modello) e ha dimensione $n \times n$,

$B(t)$ è la matrice di input e ha dimensione $n \times s$,

$G(t)$ è la matrice di rumore e ha dimensione $n \times q$

Forma standard del modello di osservazione (o modello di misura di un sistema):

$$z(t) = H(t)x(t) + D(t)w(t) \tag{1.2}$$

Dove

$z(t) \in \Re^m$ è il vettore di osservazione,

$w(t) \in \Re^r$ è una variabile casuale che descrive l'incertezza nell'osservazione

$H(t)$ è la matrice del modello di misura e ha dimensione $m \times n$

$D(t)$ è la matrice del modello di misura e ha dimensione $m \times r$

Le precedenti due equazioni definiscono l'evoluzione di un sistema a tempo continuo con osservazioni a tempo continuo dello stato. Chiaramente, per l'implementazione del modello di stato e di osservazione in un algoritmo di controllo o in un algoritmo di fusione (quale il filtro di Kalman), è necessario costruire i modelli a tempo discreto. In tal caso la variabile temporale continua t diventa l'insieme degli istanti di integrazione $t = \{t_0, t_1, ..., t_{k-1}, t_k\}$ fino al passo k-esimo e le equazioni (1.1) e (1.2) possono essere riscritte nel seguente modo:

$$x(t_k) = F(t_k)x(t_{k-1}) + B(t_k)u(t_k) + G(t_k)v(t_k) \tag{1.3}$$

$$z(t_k) = H(t_k)x(t_k) + D(t_k)w(t_k) \tag{1.4}$$

Si tralascia la trattazione che porta a calcolare le matrici a tempo discreto $F(t_k)$, $B(t_k)$, $G(t_k)$, $H(t_k), D(t_k)$ a partire dalla loro corrispondenti a tempo continuo. Se l'intervallo di tempo $\Delta t(k) = t_k - t_{k-1}$ tra due successive iterazioni resta costante è uso comune scrivere le equazioni (1.3) e (1.4) nella forma sintetica:

$$x(k) = F(k)x(k-1) + B(k)u(k) + G(k)v(k) \tag{1.5}$$

$$z(k) = H(k)x(k) + D(k)w(k) \tag{1.6}$$

In cui l'indice k rappresenta la variabile $t_k = t_0 + k \cdot \Delta t$.

Il modello di stato, se rispecchiasse esattamente la natura del sistema, fornirebbe una conoscenza completa della sua evoluzione, una volta note le condizioni iniziali e le variabili di input (controllo). Purtroppo, a causa di fattori di incertezza di varia natura, tra cui la stessa inevitabile approssimazione con cui il modello è costruito, spesso la conoscenza delle variabili di interesse derivante dal modello di stato non è sufficiente. Occorre perciò ottenere delle informazioni aggiuntive effettuando delle *osservazioni* (misure)

dello stato. Anch'esse rispondono a un modello più o meno completo che descrive come l'osservazione è legata allo stato da cui dipende e vi si può associare un'incertezza (di tipo probabilistico o di altro tipo). L'obiettivo fondamentale del sensor fusion, nel caso di veicoli autonomi, è quello di migliorare la conoscenza della posizione del veicolo in funzione del tempo allo scopo di poter utilizzare i veicoli per compiti complessi che richiedono una misura e un controllo di posizione molto accurati, cercando di ridurre l'incertezza introdotta dai fattori di disturbo; tra questi, oltre alle fonti di incertezza che caratterizzano tutti i sensori, è da considerarsi anche il non perfetto rispetto dei vincoli non olonomi (slittamento delle ruote e altri fenomeni), il che si traduce in una incertezza di modello oltre a quella di misura.

1.1.3 Teorema di Bayes

Il teorema di Bayes è un risultato di importanza fondamentale nello studio dei modelli probabilistici e di conseguenza nella progettazione di algoritmi di data fusion.

Si considerino due variabili casuali x e z. Ai fini applicativi del teorema di Bayes al problema del data fusion si attribuisce alla variabile x lo stato di interesse del sistema, o misurando, e a z l'osservazione di quello stato, o misura (ad esempio lo stato di interesse può essere la velocità lineare di un veicolo e l'osservazione può essere un valore misurato di velocità angolare delle ruote, che attraverso la conoscenza del raggio delle ruote, in genere del modello di osservazione, porta alla stima della velocità lineare, oppure può essere una misura diretta di velocità lineare).

La regola di derivazione delle probabilità condizionali può essere usata per esprimere la funzione densità in due modi diversi:

$$P(x,z) = P(x|z)P(z) = P(z|x)P(x) \tag{1.7}$$

Il teorema di Bayes si ottiene esprimendo la $P(x|z)$ grazie all'ultima equazione (1.7) scritta sopra:

$$P(x|z) = \frac{P(z|x)P(x)}{P(z)} \tag{1.8}$$

Nell'equazione (1.8) la distribuzione marginale $P(z)$ serve semplicemente a normalizzare la distribuzione posteriore (è un termine che non dipende dalla variabile x che è quella di interesse, e quindi si può considerare costante):

$$\int_x P(x|z)\cdot dx = \int_x \frac{P(z|x)P(x)}{P(z)}\cdot dx = \int_x \frac{P(x,z)}{P(z)}\cdot dx = \frac{P(z)}{P(z)} = 1 \tag{1.9}$$

L'equazione (1.8) mostra che $P(x|z)$ è una funzione densità di probabilità ed esprime quello che si cerca tipicamente in una operazione di misurazione, ovvero la probabilità associata con il misurando x, data la misura z dallo strumento di cui disponiamo. A valle di ciò ci si potrà riferire alla norma ISO per dare un valore ed un intervallo con confidenza oltre alla unità di misura impiegata.

Il teorema di Bayes fornisce perciò un metodo diretto per combinare l'informazione osservata con la stima precedente della densità di probabilità associata al misurando (ovvero la $P(x)$) per ottenere una nuova densità di probabilità, condizionata alla misura z, quindi migliore di quella precedente. Esso è pertanto alla base di molti algoritmi di data fusion.

In altri termini l'informazione contenuta nell'osservazione viene utilizzata per calcolare la nuova distribuzione di probabilità associata allo stato x rispetto alla funzione densità di probabilità precedente. Questa nuova distribuzione è chiamata *distribuzione posteriore* $P(x|z)$ e descrive le probabilità associate a x data l'osservazione z.

Per comprendere più a fondo il valore e l'utilità di questa (apparentemente) semplice manipolazione delle funzioni densità di probabilità occorre analizzare il significato delle funzioni $P(x|z)$, $P(z|x)$, $P(z)$ e $P(x)$.

La *pdf* $P(x)$ è la *funzione densità di probabilità precedente* e quantifica l'incertezza con cui si conosce il valore atteso di uno stato, prima della nuova osservazione z.

Al fine di ottenere maggiori informazioni circa la stato x, si effettua un'osservazione z. Le osservazioni sono modellate dalla funzione di densità di probabilità condizionata $P(z|x)$ che descrive, per ogni fissato stato $x \in X$, la probabilità che sarà fatta l'osservazione $z \in Z$, altrimenti detta come la probabilità di z dato x.

La distribuzione condizionata $P(z|x)$ ricopre il ruolo di quello che è chiamato *modello del sensore.* In tale modello x è l'ingresso/misurando e z è l'uscita/osservazione del sensore. Questa distribuzione può essere pensata in due modi distinti:

1. il primo è riferito alla *costruzione del modello del sensore* (Figura 1), in tal caso la distribuzione si ottiene fissando un particolare valore di $x = x_p$ trovando quale risulta la *pdf* relativa alla variabile z (coincide con il processo di taratura in cui si determina l'incertezza dello strumento in funzione dell'ingresso). Perciò, in questo caso $P(z|x_p)$ è considerata funzione di z. Per esempio si supponga di conoscere il valore vero della distanza da un target (x_t), allora $P(z|x_t)$ sarà la distribuzione delle osservazioni (misure) attorno a questo valore (Figura 2).
2. se invece esiste già un modello del sensore, supponiamo di fare un'osservazione e di fissare perciò $z = z_p$. Da questa osservazione si vuole inferire il valore dello stato x. In questo caso la distribuzione $P(z_p|x)$ è una funzione di x ed è nota come *Funzione di Verosimiglianza* $\Lambda(x) = P(z_p, x)$ (Figura 3).

Figura 1: Modello probabilistico del sensore.

In fase di taratura si opera in modo diretto come in Figura 1, però a noi interessa in realtà il processo inverso che è la $P(x \mid z)$ ovvero la distribuzione del valore del misurando x in base alla misura z.

Nell'implementazione pratica del teorema di Bayes (Equazione (1.8)), $P(z|x)$ è costruita come funzione di entrambe le variabili e nel caso di variabili discrete assume forma matriciale. Per ogni valore di x fissato si definisce una distribuzione di z. Quindi al variare di x viene creata una famiglia di distribuzioni in z.

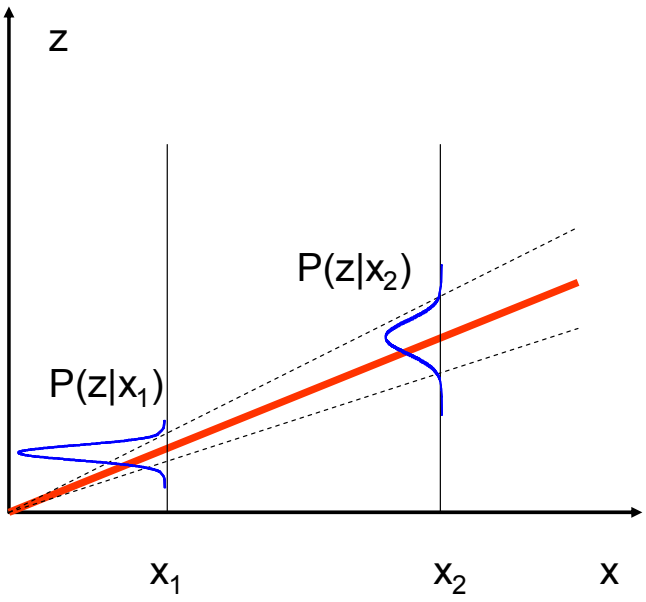


Figura 2: Modello del sensore: si suppone noto il valore vero dello stato e si osserva la distribuzione delle misure di questo valore. Equivale alla fase di taratura del sensore, in cui il valore vero del misurando è quello rilevato con uno strumento di riferimento di accuratezza superiore a quello da tarare

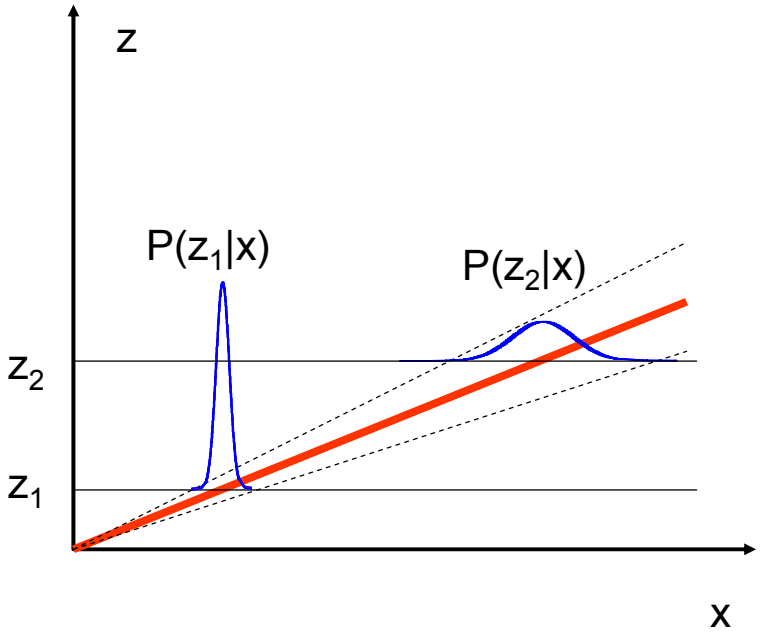


Figura 3: Funzioni di verosimiglianza: quello che si fa quando si effettua una misura è leggere un'uscita dello strumento (z_1 o z_2) e associare ad essa una distribuzione possibile del valore vero dello stato.

Esempio 1. Si consideri uno stato x a valori continui, ad esempio la distanza da un target, e un'osservazione z di questo stato. Il modello utilizzato per la distribuzione delle osservazioni del valor vero dello stato, di comune uso per questo tipo di osservazioni, è quello Gaussiano, ossia della Distribuzione Normale, in cui le osservazioni sono distribuite con media x e varianza σ_z^2:

$$P(z|x) = \frac{1}{\sqrt{2\pi}\sigma_z}\exp\left(-\frac{1}{2}\frac{(z-x)^2}{\sigma_z^2}\right) \qquad (1.10)$$

Questa è da considerarsi una funzione di entrambe le variabili z e x. Se si conosce il valore dello stato, allora la distribuzione è una funzione della sola z e descrive un modello del sensore. Se invece si effettua

un'osservazione, la distribuzione e funzione della sola x descrive la probabilità dello stato secondo una gaussiana con media z e varianza σ_z^2. In quest'ultimo caso essa è la Funzione di Verosimiglianza.

Si assuma di avere una stima precedente dello stato vero $x = x_p$, anch'essa basata sul modello di probabilità Normale, funzione della sola x

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma_x}\exp(-\frac{1}{2}\frac{(x-x_p)^2}{\sigma_x^2})$$

Il teorema di Bayes può essere applicato direttamente per combinare questa informazione precedente con l'informazione del sensore. In una implementazione pratica viene effettuata prima l'osservazione $z = z_p$, il suo valore viene sostituito alla variabile random z nell'Equazione (1.10). Poi le due distribuzioni vengono moltiplicate tra loro per produrre la distribuzione posteriore

$$\begin{aligned} P(x|z) &= K\frac{1}{\sqrt{2\pi}\sigma_z}\exp(-\frac{1}{2}\frac{(x-z_p)^2}{\sigma_z^2})\frac{1}{\sqrt{2\pi}\sigma_x}\exp(-\frac{1}{2}\frac{(x-x_p)^2}{\sigma_x^2}) \\ &= \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{1}{2}\frac{(x-\overline{x})^2}{\sigma^2}) \end{aligned}$$

La costante K è una costante indipendente da x e scelta in modo che la distribuzione posteriore sia normalizzata. La media e la varianza della distribuzione posteriore risultano essere

$$\overline{x} = \frac{\sigma_x^2}{\sigma_x^2+\sigma_z^2}z_p + \frac{\sigma_z^2}{\sigma_x^2+\sigma_z^2}x_p$$

$$\sigma^2 = \frac{\sigma_z^2\sigma_x^2}{\sigma_x^2+\sigma_z^2} = \left(\frac{1}{\sigma_x^2}+\frac{1}{\sigma_z^2}\right)^{-1}$$

La distribuzione posteriore è perciò anch'essa Gaussiana con una media (valore stimato dello stato) che risulta essere la media pesata tra le medie della distribuzione precedente e di verosimiglianza, e con una varianza (incertezza assegnata alla stima) uguale alla combinazione in parallelo tra le varianze originarie.

Il calcolo del fattore di normalizzazione K non ha in questo caso un'importanza fondamentale poiché, premesso che entrambe le distribuzioni tra loro moltiplicate siano di tipo Gaussiano, ai fini pratici si trascura l'espressione estesa delle distribuzioni passando direttamente al calcolo, noto, della media e della varianza risultante, date le medie e le varianze di partenza. Ai fini della chiarezza si riporta di seguito il calcolo del fattore di normalizzazione K. Sommando gli esponenti delle due distribuzioni si ottiene:

$$P(x|z) = K\frac{1}{2\pi\sigma_z\sigma_x}\exp\left(-\frac{1}{2}\left(\frac{\sigma_x^2(x-z_p)^2+\sigma_z^2(x-x_p)^2}{\sigma_x^2\sigma_z^2}\right)\right)$$

L'argomento dell'esponenziale può essere scritto come

$$-\frac{1}{2}\left(\frac{1}{\sigma^2}\left(x^2-2\sigma^2\left(\frac{x_p}{\sigma_x^2}+\frac{z_p}{\sigma_z^2}\right)x+\sigma^2\left(\frac{x_p^2}{\sigma_x^2}+\frac{z_p^2}{\sigma_z^2}\right)\right)\right)$$

$$= -\frac{1}{2}\frac{\left(x^2-2\overline{x}x+\overline{x}^2-\overline{x}^2+\sigma^2\left(\frac{x_p^2}{\sigma_x^2}+\frac{z_p^2}{\sigma_z^2}\right)\right)}{\sigma^2}$$

In cui è stato aggiunto e sottratto il termine $\overline{x}^2$ in modo da avere:

$$-\frac{1}{2\sigma^2}\left(\left(x-\overline{x}\right)^2-\overline{x}^2+\sigma^2\left(\frac{x_p^2}{\sigma_x^2}+\frac{z_p^2}{\sigma_z^2}\right)\right)$$

Ritornando all'espressione completa della distribuzione:

$$P(x|z) = K\frac{1}{2\pi\sigma_z^2\sigma_x^2}\exp\left(-\frac{1}{2\sigma^2}\left(\left(x-\bar{x}\right)^2-\bar{x}^2+\sigma^2\left(\frac{x_p^2}{\sigma_x^2}+\frac{z_p^2}{\sigma_z^2}\right)\right)\right)$$

Pertanto il termine noto risulta essere (ricordando che il fattore al denominatore della distribuzione normale deve essere $1/(\sqrt{2\pi}\sigma)$)

$$K = \sqrt{2\pi}\sqrt{\frac{\sigma_x^2+\sigma_z^2}{\sigma_x\sigma_z}}\exp\left(\frac{\bar{x}^2}{2\sigma^2}-\frac{1}{2}\left(\frac{x_p^2}{\sigma_x^2}+\frac{z_p^2}{\sigma_z^2}\right)\right)$$

Una volta nota la proprietà simmetrica delle distribuzioni Gaussiane (come dimostrato), ossia che il prodotto di due distribuzioni Gaussiane è ancora una distribuzione Gaussiana con media e varianza espresse dalle precedenti equazioni, non è necessario effettuare ogni volta i calcoli sulle distribuzioni ma è sufficiente utilizzare direttamente le espressioni già note della media e della varianza risultanti. Questo risultato consente il calcolo in tempo reale in un'implementazione di data fusion basata su questi modelli probabilistici. Più avanti si esporranno i risultati della combinazione, secondo il teorema di Bayes applicato a distribuzioni Gaussiane, delle informazioni di valori vettoriali e non scalari, in cui al posto della varianza si usa la matrice di covarianza.

In Figura 4 è riportato l'esempio grafico di quali possono essere, nel caso scalare, la distribuzione precedente, la funzione di verosimiglianza e la distribuzione posteriore. E' evidente che la distribuzione posteriore è ancora di tipo Gaussiano, come ci si aspetta, è situata in posizione interposta tra le altre due distribuzioni (rispetto alle rispettive medie) ed è una campana più stretta avendo una varianza minore delle due varianza originarie.

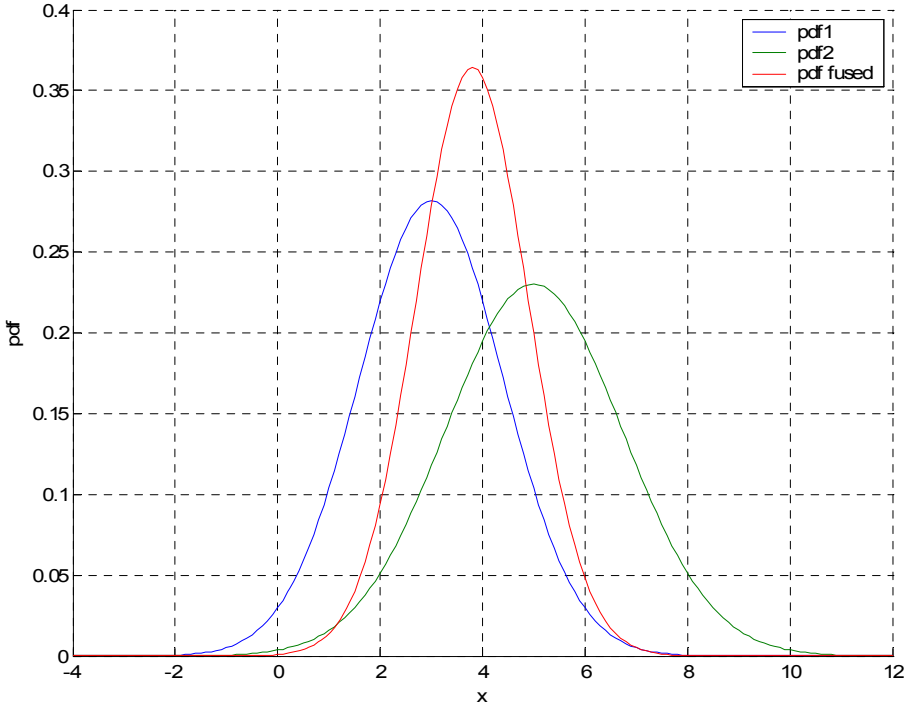


Figura 4: esempio grafico di quali possono essere, nel caso scalare, la distribuzione precedente, la funzione di verosimiglianza e la distribuzione posteriore. La distribuzione posteriore ha come media la media pesata tra le medie originarie e ha una varianza minore di entrambe quelle originarie, infatti possiede un contenuto di informazione maggiore dei singoli contributi.

Esempio 2. Si consideri il caso in cui lo stato *x* sia una variabile vettoriale continua e sia tale anche il vettore di osservazione *z*.

Per descrivere quantitativamente la dispersione della variabile aleatoria vettoriale *y* è necessario calcolare la *matrice di covarianza* definita come

$$C = \frac{1}{N}\sum_{i=1}^{N}(y_i - \overline{y})(y_i - \overline{y})^T$$

dove $\overline{y}$ è il valore medio degli N campioni y_i della variabile y. Se y_i è l'i-esimo campione di y del tipo:

$$y_i = [\eta_i, \xi_i]$$

Allora la matrice di covarianza assume la forma

$$C = \begin{bmatrix} \sigma_\eta^2 & \mathrm{cov}(\eta,\xi) \\ \mathrm{cov}(\xi,\eta) & \sigma_\xi^2 \end{bmatrix}$$

C è una matrice simmetrica e i suoi termini misti sono non nulli solo se le componenti del vettore y sono tra loro correlate.

Nell'ipotesi che lo stato segua una distribuzione Normale di probabilità, la distribuzione precedente $\mathrm{P(x)}$ potrà essere rappresentata con una gaussiana multi-dimensionale del tipo

$$P(x) = \frac{1}{\sqrt{2\pi^d \det(C_x)}} e^{-\frac{1}{2}(x-x_p)^T C_x^{-1}(x-x_p)}$$

Dove d è la dimensione del vettore e x_p il valor medio.

Se perciò si ipotizza di effettuare un'osservazione $z = z_p$, vettore a due dimensioni per esempio, con un sensore la cui funzione di verosimiglianza relativa z_p abbia matrice di covarianza C_z:

$$P(z|x) = \frac{1}{2\pi \det C_z} e^{-\frac{1}{2}(z_p-x)^T C_z^{-1}(z_p-x)}$$

E' infine possibile calcolare la distribuzione posteriore $P(x \mid z)$, nello stesso modo del caso precedente ma tenendo conto che le distribuzioni sono a valori vettoriali. Pertanto la distribuzione posteriore sarà ancora gaussiana, con matrice di covarianza pari al parallelo delle matrici di covarianza di partenza, ossia:

$$C = (C_x^{-1} + C_z^{-1})^{-1}$$

L'espressione di C si può scrivere anche in modo più efficiente calcolando una sola inversione matriciale:

$$C_x^{-1} + C_z^{-1} = C_x^{-1}C_zC_z^{-1} + C_z^{-1} = (C_x^{-1}C_z + I)C_z^{-1} =$$
$$= (C_x^{-1}C_z + C_x^{-1}C_x)C_z^{-1} = C_x^{-1}(C_z + C_x)C_z^{-1}$$

Da cui

$$\left(C_x^{-1} + C_z^{-1}\right)^{-1} = \left(C_x^{-1}(C_x + C_z)C_z^{-1}\right)^{-1} = C_z(C_x + C_z)^{-1}C_x$$

Quindi

$$\mathrm{C} = \mathrm{C_z}(\mathrm{C_x} + \mathrm{C_z})^{-1}\mathrm{C_x}$$

Mentre la media della distribuzione posteriore risulterà essere (analogamente al caso scalare)

$$\overline{\mathrm{x}} = \mathrm{C_x}(\mathrm{C_x} + \mathrm{C_z})^{-1}\mathrm{z_p} + \mathrm{C_z}(\mathrm{C_x} + \mathrm{C_z})^{-1}\mathrm{x_p}$$

Graficamente una distribuzione Normale bidimensionale è rappresentata da una campana di Gauss le cui sezioni trasversali sono delle ellissi che rappresentano il luogo dei punti caratterizzati da uguale probabilità. Nella matrice di covarianza sono contenute le informazioni sulle dimensioni degli assi principali delle ellissi (autovalori della matrice) e sulla loro orientazione (autovettori). Un esempio di quello che graficamente corrisponde alle equazioni appena ottenute è riportato in Figura 5.

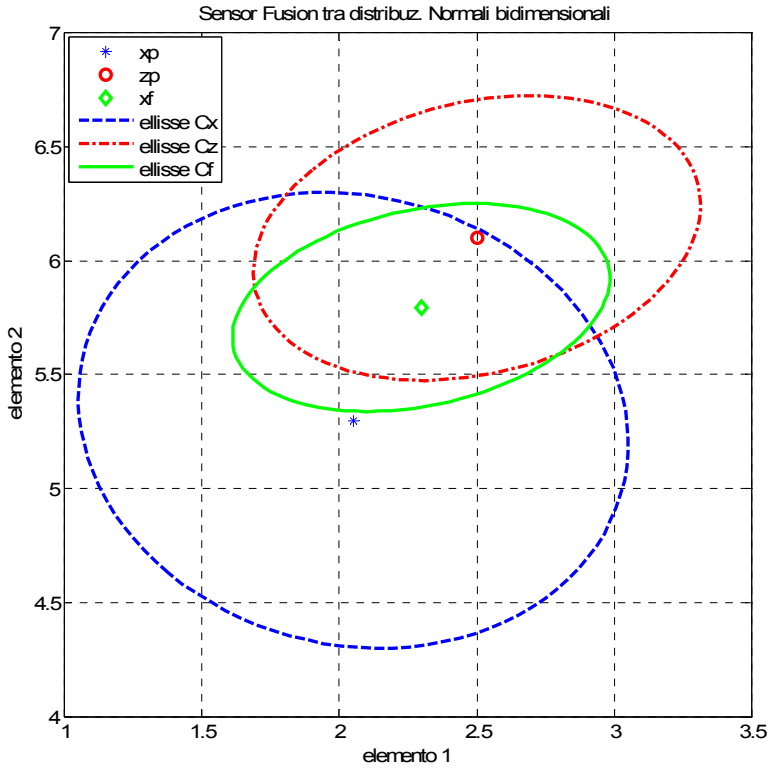


Figura 5: Fusione tra la distribuzione di probabilità precedente e la distribuzione di probabilità relativa all'osservazione zp.

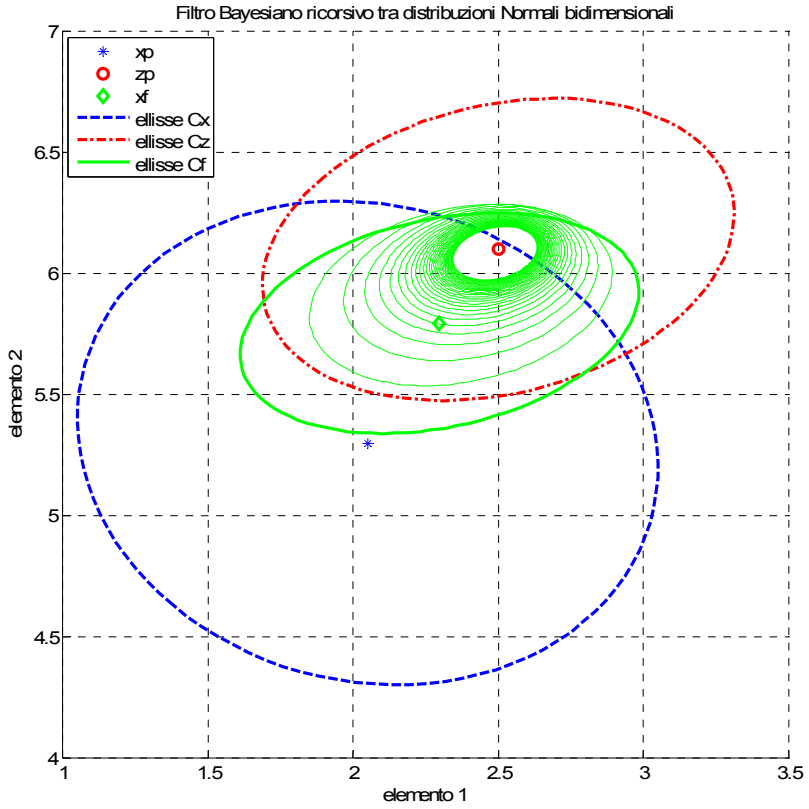


Figura 6: Ellissi di equiprobabilità nel caso in cui l'osservazione ottenuta e la relativa ellisse di incertezza restino sempre le stesse. L'ellisse della distribuzione posteriore converge con il proprio centro verso il valore osservato e va riducendosi sempre di più ad ogni iterazione.

In Figura 6 si è simulato il caso in cui l'osservazione z_p venga ripetuta 20 volte. Come atteso, l'ellisse della distribuzione posteriore converge con il proprio centro verso il valore osservato e va riducendosi sempre di più ad ogni iterazione nonostante l'ellisse di equiprobabilità dell'osservazione (ottenuto dalla funzione di verosimiglianza) resti sempre lo stesso.

1.1.4 Applicazione del teorema di Bayes al sensor fusion

Il teorema di Bayes si può applicare direttamente alla fusione delle informazioni provenienti da differenti sorgenti (sensori). Ciò che si vuole fare è ottenere la distribuzione posteriore

$P(x|Z^n)$, dove $Z^n \triangleq \{z_1 \in Z_1, ..., z_n \in Z_n\}$ è l'insieme delle osservazioni.

La distribuzione posteriore descrive le probabilità dei vari valori dello stato $x \in X$ data l'informazione ottenuta dalle n osservazioni. In linea di principio, il teorema di Bayes impiegato direttamente per calcolare la funzione di distribuzione fornisce

$$\begin{aligned} P(x|Z^n) &= \frac{P(Z^n|x)P(x)}{P(Z^n)} \\ &= \frac{P(z_1, z_2, ..., z_n|x)P(x)}{P(z_1, z_2, ..., z_n)} \end{aligned} \tag{1.11}$$

In pratica sarebbe difficili applicare questo calcolo perché si dovrebbe conoscere completamente la distribuzione congiunta e condizionata $P(z_1, z_2, ..., z_n|x)$, cioè la distribuzione congiunta di tutte le possibili combinazioni di osservazioni condizionata allo stato. Si ritiene ragionevole, sotto certe ipotesi, assumere che una volta dato lo stato $x \in X$, l'informazione ottenuta dalla i-esima sorgente di informazione sia indipendente dall'informazione delle altre sorgenti. Grazie a questa assunzione, estendendo la (1.6) al caso di n variabili,

$$P(z_i|x, z_1, ..., z_{i-1}, z_{i+1}, ..., z_n) = P(z_i|x)$$

$$P(z_1, ..., z_n|x) = P(z_1|x)...P(z_n|x) = \prod_{i=1}^{n} P(z_i|x) \tag{1.12}$$

Sostituendo la (3-11) nella (3-10),

$$P(x|Z^n) = \left[P(Z^n)\right]^{-1} P(x) \prod_{i=1}^{n} P(z_i|x) \tag{1.13}$$

Quindi la distribuzione posteriore su x, vale a dire le probabilità aggiornate nello stato, sono proporzionali al prodotto della distribuzione di probabilità precedente e alle distribuzioni di probabilità di ciascuna osservazione. La distribuzione marginale $P(Z^n)$ agisce come costante di normalizzazione. L'equazione (1.13) fornisce un metodo semplice per la fusione di informazioni da più sensori ed è chiamata *gruppo indipendente di probabilità*. Nell'implementazione di questo algoritmo, prima di tutto si memorizzano le probabilità condizionate $P(z_i|x)$ come funzioni di entrambe le variabili z_i e x. Quando viene effettuata una sequenza di osservazioni (di sensori diversi ma anche dello stesso sensore) i valori osservati sono sostituiti nelle rispettive distribuzioni di probabilità e si costruiscono le n funzioni di verosimiglianza $\Lambda_i(x)$. Il prodotto di queste funzioni per l'informazione precedente $P(x)$, normalizzato dalla $P(Z^n)$, fornisce l'aggiornamento dell'informazione sullo stato $P(x|Z^n)$ che è funzione della sola x per uno specifico insieme di n osservazioni $\{z_1, z_2, ..., z_n\}$.

L'efficacia di questo metodo è basata sull'ipotesi che le osservazioni siano indipendenti tra loro quando condizionate al valore vero dello stato. Tale assunzione è ragionevole se lo stato a cui le osservazioni si riferiscono è la sola cosa che esse hanno in comune, perciò una volta che lo stato sia stato specificato è ragionevole assumere che le informazioni siano condizionalmente indipendenti. Ciò non sarebbe sicuramente

corretto senza la condizionalità: sarebbe errato dire che le informazioni sono incodizionalmente indipendenti poiché

$$P(z_1, z_1, \ldots z_n) \neq \prod_{i=1}^{n} P(z_i)$$

ogni contenuto i-esimo dipende dallo stato comune $x \in X$.

L'ipotesi di indipendenza condizionale non è detto che sia sempre ragionevole, ad esempio nel caso in cui un'osservazione può modificare sensibilmente lo stato.

Esempio 3. Si consideri una possibile applicazione del teorema di Bayes al sensor fusion tra due sensori per stimare un parametro discreto. L'ambiente di interesse è modellato da un singolo stato x che può assumere uno dei seguenti tre valori:

x_1: *x è un target di tipo 1;*

x_2: *x è un target di tipo 2;*

x_3: *non ci sono target visibili;*

Un singolo sensore osserva x e restituisce tre possibili valori:

z_1: *osservazione di un target di tipo 1;*

z_2: *osservazione di un target di tipo 2;*

z_3: *nessun target osservato;*

Il modello del sensore A è descritto dalla matrice di probabilità:

$$P_A(z_A|x) = \begin{bmatrix} & z_1 & z_2 & z_3 \\ x_1 & 0.45 & 0.45 & 0.1 \\ x_2 & 0.45 & 0.45 & 0.1 \\ x_3 & 0.1 & 0.1 & 0.8 \end{bmatrix}$$

Il modello del sensore B è descritto dalla matrice di probabilità

$$P_B(z_B|x) = \begin{bmatrix} & z_1 & z_2 & z_3 \\ x_1 & 0.45 & 0.1 & 0.45 \\ x_2 & 0.1 & 0.45 & 0.45 \\ x_3 & 0.45 & 0.45 & 0.1 \end{bmatrix}$$

Queste matrici di probabilità sono funzione sia di x che di z. Per un fissato valore vero dello stato esse descrivono la probabilità che venga effettuata una particolare osservazione (questo è il modello del sensore e in tal senso la matrice va letta lungo le sue righe). Quando viene effettuata una particolare osservazione, la matrice descrive la distribuzione di probabilità del valore vero dello stato (questa è la funzione di verosimiglianza e in tal senso la matrice va letta lungo le sue colonne).

La distribuzione posteriore dello stato x condizionata all'osservazione $z_A = z_i$, effettuata ad esempio dal sensore A, è data da

$$P(x|z_{A,i}) = k \times P_A(z_i|x)P(x)$$

Dove k è il fattore di normalizzazione definito dalla condizione che la somma delle probabilità della variabile x sia unitaria.

Se assumiamo di non avere nessuna conoscenza precedente circa lo stato:

$$P(x) = [1/3, 1/3, 1/3] \simeq [0.333, 0.333, 0.333]$$

Supponendo che venga resa disponibile l'osservazione $z_A = z_1$ dal sensore A, in tal caso la distribuzione posteriore sarà data dalla prima colonna della matrice di probabilità del sensore A (la notazione $\otimes$ indica il prodotto elemento per elemento), cioè la funzione di verosimiglianza per l'osservazione $z_A = z_1$

$$P(x|z_{A,1}) = k \times P_A(z_1|x)P(x)$$
$$= k \times (0.45, 0.45, 0.1) \otimes (0.333, 0.333, 0.333) = (0.45, 0.45, 0.1)$$

Se successivamente usiamo la distribuzione posteriore come distribuzione precedente per la fusione con una nuova osservazione del sensore A, $P(x) = (0.45, 0.45, 0.1)$, e viene resa disponibile una nuova osservazione, supponiamo ancora una volta $z_A = z_1$, la nuova distribuzione posteriore sarà:

$$P(x|z_{A,1}) = k \times P_A(z_1|x)P(x)$$
$$= k \times (0.45, 0.45, 0.1) \otimes (0.45, 0.45, 0.1) = (0.488, 0.488, 0.024)$$

Riassumendo, nel caso in cui vengano fatte due osservazioni successive z_1 del sensore A, l'algoritmo di fusione delle informazioni porta ad aumentare la probabilità che uno dei due target sia presente a scapito dell'ipotesi di nessun target visibile.

Dall'osservazione delle matrici di probabilità dei due sensori appare evidente che mentre il sensore A è più efficace nello stimare la presenza di un target ma meno ad osservarne la presenza, il sensore B per contro è più efficace nel distinguere il tipo di target ma meno efficace a dire se il target è visibile o meno. Allora viene spontaneo utilizzare le informazioni di entrambi i sensori e combinarle tra loro in modo da ottenere un'informazione più completa e stimare con maggior probabilità se il target è visibile o meno e di che tipo di target si tratta.

Supponiamo dunque di avere una distribuzione precedente uniforme come nel caso precedente. Se osserviamo $z_A = z_1$ con il sensore A e $z_B = z_1$ con il sensore B, la distribuzione posteriore in x sarà

$$P(x|z_{A,1}, z_{B,1}) = k \times P_{AB}(z_1, z_1|x) \times P(x)$$
$$= k \times P_A(z_1|x)P_B(z_1|x)$$
$$= k \times (0.45, 0.45, 0.1) \otimes (0.45, 0.1, 0.45) = (0.6924, 0.1538, 0.1538)$$

Comparando questo risultato con il risultato del caso precedente in cui si effettuano due misure col sensore A è evidente come il sensore B fornisca una sostanziale informazione aggiuntiva sulla discriminazione del tipo di target a spese di una leggera perdita di efficacia nell'individuazione del target.

Ripetendo il calcolo per ciascuna coppia di osservazioni , risultano le seguenti matrici di probabilità

$z_A = z_1$

$$\begin{bmatrix} z_B = & z_1 & z_2 & z_3 \\ x_1 & 0.692 & 0.154 & 0.488 \\ x_2 & 0.154 & 0.692 & 0.488 \\ x_3 & 0.154 & 0.154 & 0.024 \end{bmatrix}$$

$z_A = z_2$

$$\begin{bmatrix} z_B = & z_1 & z_2 & z_3 \\ x_1 & 0.692 & 0.154 & 0.488 \\ x_2 & 0.154 & 0.692 & 0.488 \\ x_3 & 0.154 & 0.154 & 0.024 \end{bmatrix}$$

$z_A = z_3$

$$\begin{bmatrix} z_B = & z_1 & z_2 & z_3 \\ x_1 & 0.108 & 0.024 & 0.265 \\ x_2 & 0.0241 & 0.108 & 0.265 \\ x_3 & 0.868 & 0.868 & 0.471 \end{bmatrix}$$

Da cui risulta che la prestazione generale del sistema di osservazione è migliorata in ciascuno dei casi possibili. Se infatti si osserva un target di tipo 2 con il secondo sensore dopo aver osservato un target di tipo 1 con il primo sensore, la distribuzione posteriore risulta $P(x|z_{A,1}, z_{B,2}) = (0.154, 0.692, 0.154)$ che migliora l'individuazione del target 2. Se, ad ulteriore conferma dell'efficacia, non osserviamo nessun target con il secondo sensore e osserviamo un target di tipo 1 con il primo sensore, la distribuzione posteriore risulta $(0.488, 0.488, 0.024)$. Ossia si stima ancora bene che è visibile un target sebbene non si sappia quale in quanto il secondo sensore non è stato in grado di distinguerlo. Se poi il sensore A osservasse che non è visibile nessun target mentre il sensore B osserva il target di tipo 2, la distribuzione posteriore è $(0.108, 0.024, 0.868)$, il che indica che comunque si stima che non ci sia nessun target nonostante l'osservazione del sensore B, che però contiene poca informazione in quanto non è in grado si effettuare una buona individuazione del target ma solo a distinguere tra i due tipi di target.

1.1.5 Filtro Bayesiano ricorsivo

La fusione delle informazioni di più osservazioni o di più sensori che utilizza l'Equazione (1.13) richiederebbe, in linea di principio, di memorizzare tutta l'informazione passata e, all'arrivo di una nuova k-esima informazione nella forma $P(z_k|x)$, di ricalcolare la probabilità complessiva aggiornata. Il teorema di Bayes si presta facilmente all'implementazione ricorsiva o incrementale. Considerando infatti che si può scrivere $Z^k \triangleq \{z_k, Z^{k-1}\}$ e assumendo l'indipendenza condizionale delle osservazioni,

$$\begin{aligned} P(x, Z^k) &= P(x|Z^k)P(Z^k) \\ &= P(z_k, Z^{k-1}|x)P(x) \\ &= P(z_k|x)P(Z^{k-1}|x)P(x) \end{aligned}$$

Eguagliando i termini agli estremi di quest'equazione si ottiene:

$$P(x|Z^k)P(Z^k) = P(z_k|x)P(Z^{k-1}|x)P(x)$$
$$= P(z_k|x)P(x|Z^{k-1})P(Z^{k-1})$$

Ricordando che $P(Z^k)/P(Z^{k-1}) = P(z_k|Z^{k-1})$,

$$P(x|Z^k) = \frac{P(z_k|x)P(x|Z^{k-1})}{P(z_k|Z^{k-1})} \tag{1.14}$$

Il vantaggio di questo algoritmo ricorsivo è che ad ogni passo di iterazione è sufficiente calcolare e memorizzare solo la distribuzione posteriore $P(x|Z^k)$ la quale contiene una sintesi completa di tutte le informazioni passate. Quando si dispone di una nuova osservazione e quindi di una nuova funzione di probabilità $P(z_k|x)$, la distribuzione posteriore diventa quella precedente e le due funzioni, moltiplicate tra loro, diventano, dopo averne normalizzato il prodotto, la nuova distribuzione posteriore.

Esempio 4. Volendo applicare il filtro Bayesiano ricorsivo al caso di data fusion del §Esempio 1., la media e la varianza della distribuzione posteriore aggiornata al passo k, si possono calcolare immediatamente e risultano essere:

$$x_k = \frac{\sigma_{k-1}^2}{\sigma_{k-1}^2 + \sigma^2} z_k + \frac{\sigma^2}{\sigma_{k-1}^2 + \sigma^2} x_{k-1}$$
$$\sigma_k^2 = \frac{\sigma_{k-1}^2 \sigma^2}{\sigma_{k-1}^2 + \sigma^2} = \left(\frac{1}{\sigma_{k-1}^2} + \frac{1}{\sigma^2} \right)^{-1}$$

Dove σ^2 è la varianza associata alla nuova osservazione z_k, σ_{k-1}^2 è la varianza associata alla distribuzione precedente.

1.1.6 Filtro di Kalman

Tutti i problemi di data fusion includono una fase di stima. Dato un certo numero di misure (osservazioni) provenienti da un gruppo di sensori, vogliamo usare le informazioni ottenute per ottenere una stima che sia migliore di ognuna delle singole misure in termini probabilistici, dello stato di interesse. Uno stimatore è una regola decisionale che, data in ingresso una sequenza di misure di una grandezza di interesse, calcola il valore di quella grandezza (parametro, stato di un sistema, misurando). Il filtro di Kalman è un *algoritmo ottimo ricorsivo di elaborazione dati, o stimatore lineare ricorsivo*. La definizione di ottimo è riferita, sotto le ipotesi su cui si basa, alle prestazioni del filtro di Kalman rispetto a quelle di qualsiasi altro possibile algoritmo di data fusion (Maybeck P.S.). Esso elabora tutte le misure disponibili, qualsiasi sia la loro accuratezza, per stimare il valore corrente delle variabili di interesse. Ad esempio, supponiamo di voler misurare la velocità di un veicolo e di avere a disposizione le misure provenienti da un laser a triangolazione o da un gruppo di sensori odometrici o da una piattaforma inerziale (accelerometri e giroscopi). Piuttosto che essere costretti a scegliere solo una di queste misure e ignorare tutte le altre, è possibile costruire un filtro di Kalman che prenda in ingresso tutte le misure disponibili e combini tra loro le informazioni contenute in ciascun dato, compresa la conoscenza del modello dello strumento che lo ha fornito. Il risultato in uscita sarà una migliore stima della velocità.

Il temine *ricorsivo* è fondamentale per un'implementazione al calcolatore del filtro di Kalman, esso indica infatti che l'algoritmo non richiede la memorizzazione e la rielaborazione di tutta la storia dei dati ogni volta che è disponibile una nuova misura.

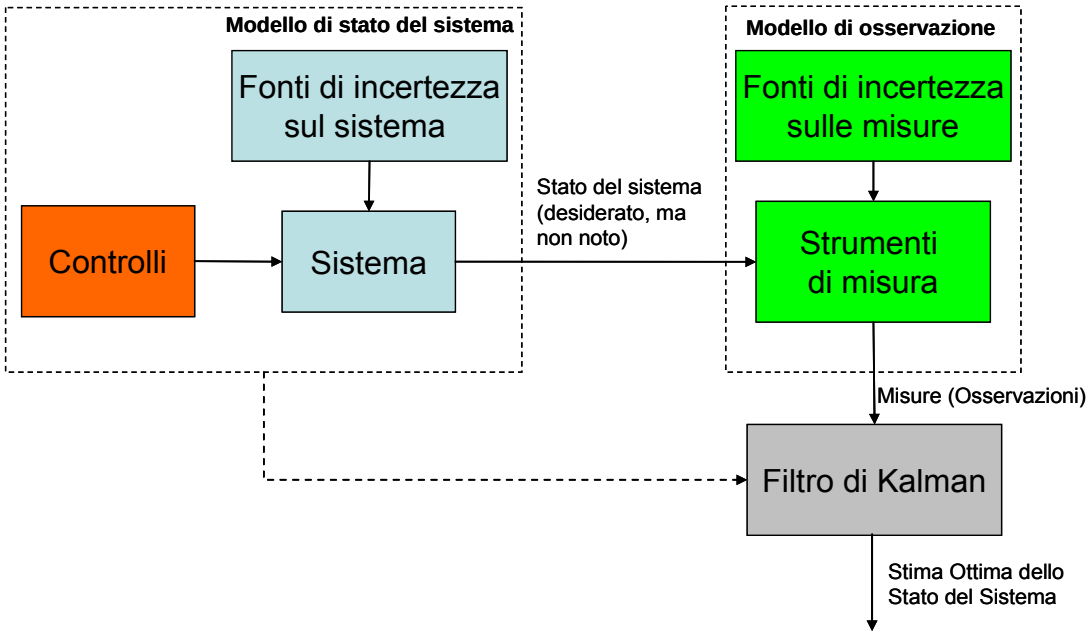


Figura 7: Schema a blocchi di una possibile applicazione del filtro di Kalman

La Figura 7 rappresenta una tipica applicazione in cui si può utilizzare il filtro di Kalman. Il sistema osservato può essere o meno asservito da alcuni input di controllo e gli strumenti di misura forniscono in output le informazioni per la retroazione e in generale per stimare certe variabili di interesse del sistema. La conoscenza di questi ingressi (controlli) e uscite (misure) è l'unica cosa disponibile del sistema fisico che può essere utilizzata per effettuare una stima ottima dello stato del sistema. Le fonti di incertezza sul sistema o *incertezza di modello* derivano dalla non completa conoscenza di tutte le variabili che influiscono sulla dinamica del sistema. Le fonti di incertezza sulle misure comprendono la non perfetta conoscenza di quali sono le relazioni tra le variabili di stato e le uscite misurate (incertezza di modello degli strumenti) e l'incertezza dovuta agli errori casuali e sistematici dei vari strumenti. E' perciò necessario un metodo per estrarre l'informazione utile dai segnali rumorosi. Ci possono essere diversi strumenti con differente caratteristica dinamica e di accuratezza che forniscono misure di una stessa variabile; è desiderabile, quando non necessario, un metodo per combinare tra loro le uscite dei vari strumenti per produrre una stima delle variabili di interesse in modo da minimizzare l'incertezza in senso statistico. Il concetto di filtro ottimo che definisce il filtro di Kalman è inteso in senso statistico ed equivale a dire che, se applicassimo dei filtri di prova molte volte per la stessa applicazione, allora la media dei risultati del filtro di Kalman sarebbe migliore della media dei risultati di qualsiasi filtro di prova.

Per minimizzare l'incertezza in senso probabilistico, ossia in senso Bayesiano, si vuole che il filtro propaghi la densità di probabilità condizionata del misurando (grandezza di interesse), condizionata alla conoscenza delle misure attuali degli strumenti. Ad esempio, la densità di probabilità indicata come $f_{x(i)|\boldsymbol{z}(1),\boldsymbol{z}(2),\ldots,\boldsymbol{z}(i)}(x \mid \boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_i)$ è la funzione che descrivere la probabilità che il misurando (supponiamo sia uno scalare) $x \in X$ assuma un particolare valore all'interno di X (per esempio, $X \equiv \mathbb{R}$) all'istante i ($x(i)$); questa probabilità sia condizionata alla conoscenza che il vettore delle misure $\boldsymbol{z}(1)$ all'istante 1 assuma il valore $\boldsymbol{z}_1$ e così via per l'istante 2 fino a i. Si supponga, ad esempio, che $x(i)$ sia la posizione monodimensionale di un veicolo all'istante di tempo i, e che $\boldsymbol{z}(j)$ sia il vettore bi-dimensionale delle misure della posizione al tempo j provenienti da due sensori laser. Tale densità di probabilità condizionata contiene tutta l'informazione disponibile su $x(i)$: essa indica, per ogni dato valore di tutte le misure ottenute fino all'istante i, qual è la probabilità che il misurando abbia un dato valore o intervallo di valori $x(i)$. La forma del grafico della funzione f contiene tutta l'informazione riguardo all'incertezza sul valore di x: se è configurata come un picco stretto e alto allora il maggior contenuto di probabilità si concentra in un intervallo ristretto di valori di x, se invece è configurata con un andamento poco pendente, l'incertezza su x è più elevata e la sua probabilità è distribuita in un intervallo più ampi di valori.

La stima ottima si ottiene direttamente una volta che sia stata propagata la densità condizionata poiché quest'ultima contiene le informazioni necessarie a scegliere un valore ottimo per il misurando. Possibili candidati alla scelta sono la *media*, la *moda* e la *mediana.* Tali valori, nel caso del filtro di Kalman, coincidono; infatti la propagazione della densità di probabilità condizionata avviene sotto le ipotesi che:

- il sistema sia descritto da un modello *lineare*
- il rumore associato al sistema e alle misure sia *bianco* e *Gaussiano* (ossia a media nulla, con densità di potenza spettrale distribuita uniformemente su tutta la banda di frequenze e con distribuzione di probabilità Normale).

Sotto queste tre ipotesi restrittive il filtro di Kalman è definito ottimo nel senso descritto sopra.

Il modello di sistema lineare è giustificabile per varie ragioni. Spesso un modello lineare si adatta bene allo scopo e, in presenza di non linearità, l'approccio ingegneristico più utilizzato è quello della linearizzazione del modello attorno a una qualche configurazione del sistema, ad esempio un punto o una traiettoria. I sistemi lineari sono maggiormente auspicabili perchè sono più facilmente gestibili con strumenti ingegneristici e la teoria dei sistemi lineari è molto più completa e pratica di quelli non lineari. Esistono poi dei metodi per estendere l'applicazione del filtro di Kalman ai sistemi non lineari qualora i modelli lineari dovessero rivelarsi inadeguati.

L'ipotesi di rumore bianco implica che i valori di rumore non sono correlati tra di loro nel tempo. In pratica, se si conosce quanto vale il rumore all'istante attuale, ciò non aggiunge nessuna informazione ai fini di una previsione su quale sarà il suo valore in un altro istante. Un rumore bianco ha uno spettro con uguale densità di potenza per tutte le frequenze. Ciò implica che un tale segnale ha potenza infinita, pertanto un rumore bianco non può esistere in natura. Poiché però ogni sistema fisico ha una banda passante limitata nello spazio delle frequenze ed è tipicamente affetto da un rumore a larga banda, dal punto di vista del sistema è equivalente ad assumere che vi sia rumore bianco giacché questo, all'interno della banda passante del sistema, è identico al rumore reale a banda larga e al di fuori della banda passante il rumore non ha influenza sul sistema. L'assunzione di rumore bianco semplifica notevolmente la matematica utilizzata nel filtro di Kalman e non aggiunge restrizioni al modello del sistema se questo è affetto da rumore a banda larga. Se invece all'interno della banda passante del sistema il rumore non ha densità spettrale uniforme oppure è correlato nel tempo, attraverso un filtro aggiuntivo è possibile riprodurre, partendo da un rumore bianco, un rumore correlato (il filtro consiste in un sistema lineare chiamato "*shaping filter*").

Mentre l'attributo "bianco" per un rumore è riferito alle sue caratteristiche temporali (o di frequenza), l'attributo "Gaussiano" è riferito alla sua ampiezza. Cioè, per ogni singolo istante temporale, la densità di probabilità dell'ampiezza di un rumore Gaussiano ha la nota forma di una campana. Questa assunzione è giustificata in senso fisico dal fatto che, tipicamente, vi è un gran numero di piccole sorgenti che contribuiscono a creare il rumore di misura. Matematicamente questo fenomeno è descritto dal teorema del limite centrale: all'aumentare del numero di variabili casuali indipendenti che si sommano tra loro, qualunque sia la distribuzione di probabilità di ciascuna, la distribuzione di probabilità della somma tende ad essere quella Gaussiana. Oltre a questa giustificazione se ne può aggiungere una di ordine pratico. Di solito di una variabile affetta da rumore si conoscono solo le statistiche del primo e del secondo ordine, rispettivamente la media e la varianza (o la deviazione standard), in tal caso, cioè quando non si conoscono altre statistiche di ordine superiore al secondo, non vi è altra distribuzione da assumere migliore di quella Gaussiana. Una Gaussiana è determinata completamente una volta noti i due parametri di media e varianza, a differenza di molte distribuzioni che richiedono la conoscenza di un numero anche infinito di statistiche. Il filtro di Kalman, che propaga la media e la varianza, include tutte le informazioni contenute nella densità di probabilità condizionata piuttosto che soltanto alcune di esse, come succederebbe con un'altra forma di distribuzione di probabilità. Le ipotesi su cui si basa il filtro di Kalman sarebbero eccessivamente restrittive in alcuni casi se l'obiettivo fosse quello di volere costruire un modello del sistema che si avvicini il più possibile alla realtà, quindi abbandonando le ipotesi di linearità e di rumore bianco e Gaussiano. L'obiettivo è invece quello di costruire un buon algoritmo stimatore (o di controllo), in tal caso le ipotesi fatte consentono di ottenere spesso una rappresentazione del sistema adeguata allo scopo e di utilizzare strumenti matematici facilmente trattabili. L'esigenza è quindi che il modello sia efficiente per effettuare una buona stima dei parametri di interessi, e tale esigenza condiziona anche l'estensione del modello a casi di maggior applicabilità e in cui le ipotesi citate non siano più valide.

Esempio 5. Viene riportato un esempio di applicazione del filtro di Kalman al caso della stima di una singola variabile, si supponga sia la posizione monodimensionale di un automobile ovvero la sua posizione

nel caso di traiettoria rettilinea; questo esempio consente di trattare il problema in modo intuitivo anche includendo il caso dinamico in cui il misurando vari nel tempo.

Si supponga di viaggiare su un percorso rettilineo e di non conoscere la propria posizione. A un certo istante t_1 si effettua un'osservazione z_1 della propria posizione stimandola ad esempio in base alla conoscenza, più o meno accurata, del tempo trascorso dalla partenza e della velocità media. Il risultato di questa misura sarà certamente affetto da un'incertezza che viene stimata e rappresentata dalla deviazione standard (la statistica del secondo ordine è la varianza $\sigma_{z_1}^2$). Si dispone perciò della funzione $f_{x(t_1)|z(t_1)}(x \mid z_1)$ che rappresenta la distribuzione di probabilità condizionata della posizione al tempo t_1, $x(t_1)$, condizionata al valore osservato z_1. Se la distribuzione è Gaussiana la deviazione standard σ_{z_1} è una misura diretta dell'incertezza della misura z_1, ed è tale per cui la probabilità che il valore vero di x si trovi nell'intervallo centrato in z_1 e con semi-ampiezza σ_{z_1} è pari al 68.3%. Parlando in termini di probabilità Bayesiana, se si suppone che la *distribuzione precedente* di $x(t_1)$ sia rettangolare con uguale probabilità per qualsiasi valore di x, la *distribuzione posteriore* dopo la misura (osservazione) z_1, sarà Gaussiana con varianza della stima:

$$\sigma_x^2(t_1) = \sigma_{z_1}^2$$

E la migliore stima della posizione sarà:

$$\hat{x}(t_1) = z_1$$

Si supponga poi di poter disporre della misura di posizione tramite GPS o, meglio ancora, DGPS, al tempo $t_2 \cong t_1$ (in modo che non sia cambiata la posizione del veicolo in modo apprezzabile), ottenendo una misura z_2 con varianza $\sigma_{z_2}^2$ inferiore a $\sigma_{z_1}^2$. A questo punto si dispone di un'ulteriore osservazione dello stato, cioè una seconda, più accurata, misura della posizione. Come visto nei paragrafi precedenti, un metodo per combinare tra loro in termini probabilistici due misure della stessa variabile con le rispettive distribuzioni di probabilità, è quello di fonderle in modo Bayesiano. Come visto nel §Esempio 1, se si considera la stima effettuata all'istante t_1 come stima precedente ($P(x)$) e si vuole ottenere una migliore stima dopo l'osservazione t_2, per ottenere la distribuzione posteriore, ossia la stima aggiornata, basta moltiplicare tra loro la funzione di distribuzione precedente e quella dell'osservazione attuale. Nel caso di distribuzioni Gaussiane non occorre moltiplicare tra loro le funzioni densità originarie in quanto le statistiche del primo e secondo ordine, media e varianza, si ottengono direttamente da quelle delle stime di partenza. In alternativa si può applicare il metodo di filtraggio ricorsivo secondo §Esempio 4. Il risultato della densità di probabilità della posizione $x(t_2)$, condizionata ad entrambe le misure effettuate agli istanti $t_1 \cong t_2$, z_1 e z_2, è la distribuzione Gaussiana con media μ e varianza σ^2 date da:

$$\mu = \frac{\sigma_{z_2}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2} z_1 + \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2} z_2$$

$$\sigma^2 = \left(\frac{1}{\sigma_{z_1}^2} + \frac{1}{\sigma_{z_2}^2} \right)^{-1}$$

La varianza risultante è inferiore ad entrambe le varianze originarie e ciò equivale a dire che l'incertezza nella stima di posizione è diminuita dopo la combinazione delle misure di partenza.

La migliore stima aggiornata è:

$$\hat{x}(t_2) = \mu$$

con varianza associata

$$\sigma_x^2(t_2) = \sigma^2 .$$

Questa stima è quella con la massima probabilità, il valore ottimizzato ai minimi quadrati. Si noti che, supponendo che la misura z_2 sia molto più accurata di z_1, cioè che $\sigma_{z_2}^2 \ll \sigma_{z_1}^2$, la varianza di questa stima combinata è comunque minore di $\sigma_{z_2}^2$; si deduce che, anche se una misura contiene poca informazione, comunque aumenta l'accuratezza del risultato del filtro.

Si può scrivere, manipolando l'ultima equazione scritta sopra,

$$\hat{x}(t_2) = \frac{\sigma_{z_2}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2} z_1 + \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2} z_2$$

$$= z_1 + \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2}(z_2 - z_1)$$

e, ricordando che $\hat{x}(t_1) = z_1$, nella forma classica in cui il filtro di Kalman è implementato:

$$\hat{x}(t_2) = \hat{x}(t_1) + K(t_2)[z_2 - \hat{x}(t_1)] \tag{1.15}$$

dove

$$K(t_2) = \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2} \tag{1.16}$$

Osservando le ultime due equazioni si evince che la stima ottima al tempo t_2, $\hat{x}(t_2)$, è uguale alla migliore predizione del suo valore prima che sia disponibile la misura z_2, $\hat{x}(t_1)$, sommata a un termine correttivo costituito dal prodotto di un coefficiente di combinazione ottima per la differenza tra z_2 e la migliore predizione del suo valore prima che questo sia effettivamente disponibile, $\hat{x}(t_1)$. Questo filtro ha pertanto una funzione detta di "predittore-correttore". Sulla base di tutte le precedenti informazioni, viene fatta una previsione del valore che le variabili di interesse e le misure a loro associate avranno nel prossimo istante di misura. Quando viene effettuata la prossima misura la differenza tra quest'ultima e il suo valore previsto è usata per correggere la previsione delle variabili di interesse.

La varianza della stima ottima può essere riscritta come:

$$\sigma_x^2(t_2) = \sigma_x^2(t_1) - K(t_2)\sigma_x^2(t_1)$$

E' importante notare che i valori $\hat{x}(t_2)$ e $\sigma_x^2(t_2)$ includono tutte le informazioni della funzione $f_{x(t_2)|\mathbf{z}(t_1),\mathbf{z}(t_2)}(x \mid \mathbf{z}_1, \mathbf{z}_2)$, il che equivale a dire che la densità di probabilità condizionata si ottiene propagando queste due variabili. Inoltre la soluzione di questo problema di fusione in caso statico con il filtro di Kalman è esattamente equivalente a quella che si otterrebbe applicando un filtro Bayesiano ricorsivo.

Consideriamo lo stesso problema nel caso *dinamico*.

Si supponga che il veicolo in questione percorra un certo tratto prima che sia disponibile una nuova misura di posizione di qualsiasi tipo. Si assume un modello di stato a velocità costante per il sistema della forma:

$$\frac{dx}{dt} = v_0 + \xi$$

Dove v_0 è una velocità nominale e ξ è un termine di rumore usato per rappresentare l'incertezza nella conoscenza del valore reale della velocità a causa dei disturbi da cui è affetta, di condizioni non nominali, di effetti non considerati nell'equazione differenziale semplice del primo ordine, etc. Il rumore ξ è modellato come rumore Gaussiano bianco con media nulla e varianza σ_ξ^2.

La densità di probabilità condizionata, date z_1 e z_2, all'istante t_2 è identica a quella ottenuta sopra. Al trascorrere del tempo, mentre il veicolo procede a velocità nominale v_0, l'incertezza dovuta al rumore sulla velocità aumenta e la funzione densità resta una Gaussiana centrata nei valori di $x(t)$ via via stimati dal

modello di stato, ma è caratterizzata da una varianza sempre maggiore, ossia è rappresentata da una campana sempre più larga e schiacciata. In altri termini, la densità di probabilità, immaginata scorrere sull'asse delle x all'aumentare del tempo, parte all'istante t_2 dalla stima ottima e si muove con velocità nominale secondo il modello dinamico allargandosi nel tempo perché si è sempre meno sicuri della posizione esatta del veicolo a causa dell'accumularsi di incertezza nel tempo. All'istante t_3^-, subito prima che sia disponibile una nuova misura all'istante t_3, la densità di probabilità $f_{x(t_3)|\mathbf{z}(t_1),\mathbf{z}(t_2)}(x \mid \mathbf{z}_1, \mathbf{z}_2)$ è una Gaussiana caratterizzata dalle seguenti media e varianza:

$$\hat{x}(t_3^-) = \hat{x}(t_2) + v_0(t_3 - t_2) \tag{1.17}$$

$$\sigma_x^2(t_3^-) = \sigma_x^2(t_3^-) + \sigma_\xi^2(t_3 - t_2) \tag{1.18}$$

La media si ottiene tramite semplice integrazione della velocità nominale nel tempo, secondo il modello. La varianza $\sigma_x^2(t_3^-)$ si ottiene propagando la varianza della stima ottima $\sigma_x^2(t_2)$ tra l'istante t_2 e l'istante t_3, o meglio calcolando l'incertezza sul valore di $\hat{x}(t_3^-)$ tramite la formula di Kleine-Mc Clintock applicata alla prima delle due equazioni riportate sopra e supponendo che la misura di tempo non sia affetta da incertezza.

$\hat{x}(t_3^-)$ è la **previsione** ottima di quale sarà il valore di x all'istante t_3^-, prima che sia effettuata una misura all'istante t_3, e $\sigma_x^2(t_3^-)$ è la varianza attesa in questa previsione.

All'istante t_3 viene effettuata una misura (ad esempio con GPS) z_3 con varianza $\sigma_{z_3}^2$. A questo punto si dispone di due densità di probabilità, una comprendente tutta la precedente informazione disponibile prima dell'ultima misura effettuata, l'altra contente l'informazione della misura stessa. E' ragionevole assumere che la sola cosa che le due osservazioni hanno in comune è lo stato del sistema, perciò le osservazioni saranno indipendenti una volta che lo stato si è realizzato. Secondo queste assunzioni le misure sono condizionatamente indipendenti tra loro. Allo stesso modo del caso statico, si combinano tra loro le due densità condizionate di probabilità producendo una distribuzione ancora Gaussiana con media

$$\hat{x}(t_3) = \hat{x}(t_3^-) + K(t_3)[z_3 - \hat{x}(t_3^-)] \tag{1.19}$$

e varianza

$$\sigma_x^2(t_3) = \sigma_x^2(t_3^-) - K(t_3)\sigma_x^2(t_3^-) \tag{1.20}$$

Dove il guadagno $K(t_3)$ è dato da

$$K(t_3) = \frac{\sigma_x^2(t_3^-)}{\sigma_x^2(t_3^-) + \sigma_{z_3}^2} \tag{1.21}$$

Questa seconda fase del filtro, in cui si calcolano la stima e la sua varianza a valle di tutte le informazioni disponibili, ossia quando si dispone dell'ultima misura, della sua varianza e della precedente previsione di stima e varianza, è detta **aggiornamento**.

L'espressione della stima ottima $\hat{x}(t_3)$ nel caso dinamico ha la stessa forma di quella che si è ottenuta nel caso statico, la differenza tra le due applicazioni del filtro è che nel caso dinamico si introduce la stima ottenuta dal modello del sistema per effettuare la previsione sulle variabili di interesse e sulle misure negli istanti in cui esse non sono disponibili, in questo intervallo di tempo l'incertezza della stima ottima, corretta nell'istante in cui si è ottenuta una misura, viene propagata fino all'istante in cui è disponibile una nuova misura.

E' interessante osservare l'espressione del guadagno $K(t_3)$. Se $\sigma_{z_3}^2$, la varianza del rumore di misura, è grande, allora il guadagno $K(t_3)$ è piccolo; semplicemente ciò significa che viene data poca confidenza a una misura molto rumorosa che perciò viene pesata poco nella fase di correzione del filtro. Se al limite $\sigma_{z_3}^2$ tendesse ad infinito il guadagno del filtro sarebbe nullo, la misura sarebbe ignorata e verrebbe tenuta in considerazione solo la stima del modello. Viceversa, se il rumore della stima del modello σ_ξ^2 ha varianza

elevata, oppure se è trascorso troppo tempo dall'ultima misura anche se il modello è accurato, $\sigma_x^2(t_3^-)$ sarà elevata e così anche il guadagno, perciò verrà data molta confidenza alla misura. Se al limite $\sigma_x^2(t_3^-) \to \infty$, allora $K(t_3) \to 1$ e perciò $\hat{x}(t_3) = \hat{x}(t_3^-) + 1 \bullet [z_3 - \hat{x}(t_3^-)] = z_3$. In questo caso l'output del modello viene ignorato e la stima ottima coincide con la misura.

1.1.6.1 Algoritmo del filtro di Kalman

Si è detto che il filtro di Kalman è uno stimatore lineare ricorsivo: calcola, ad istanti consecutivi, la stima di uno stato (misurando, a valori scalari o vettoriali) a valori continui che si evolve nel tempo, sulla base di osservazioni (misure) periodiche dello stato. Il filtro utilizza un modello statistico di come il vettore di interesse, lo stato $x(t)$, si evolve nel tempo e un modello statistico di come le osservazioni $z(t)$ sono legate a questo vettore di interesse. Il primo di questi modelli è costruito a partire dal modello di stato del sistema, lineare o linearizzato (filtro di Kalman "esteso"); il secondo modello statistico è costruito partire dal modello di osservazione che include il modello dello strumento/i di misura impiegato/i. I guadagni del filtro di Kalman sono scelti in modo che, sotto le ipotesi fatte sulle osservazioni e sul processo di stato, la stima risultante $\hat{x}(t)$ sia tale da minimizzare lo scarto quadratico medio definito da

$$L(t) = \int_{-\infty}^{+\infty} (x(t) - \hat{x}(t))^T (x(t) - \hat{x}(t)) P(x(t) \mid Z^t) dx$$

Se si differenzia $L(t)$ rispetto a $\hat{x}(t)$ e si pone l'espressione uguale a zero si ottiene il valore di $\hat{x}(t)$ che minimizza $L(t)$.

$$\frac{dL(t)}{d\hat{x}(t)} = \int_{-\infty}^{+\infty} \frac{d}{d\hat{x}(t)} [(x(t) - \hat{x}(t))^T (x(t) - \hat{x}(t)) P(x(t) \mid Z^t)] dx = 0$$

$$\Rightarrow \int_{-\infty}^{+\infty} 2(\hat{x}(t) - x(t)) P(x(t) \mid Z^t)] dx = 0 \Rightarrow \hat{x}(t) \int_{-\infty}^{+\infty} P(x(t) \mid Z^t) dx = \int_{-\infty}^{+\infty} x(t) P(x(t) \mid Z^t) dx$$

E poiché $\int_{-\infty}^{+\infty} P(x(t) \mid Z^t) dx = 1$,

$$\hat{x}(t) = \int_{-\infty}^{+\infty} x(t) P(x(t) \mid Z^t) dx$$

Quindi il valore ottenuto non è altro che la media condizionata $E(x(t) \mid Z^t)$, cioè il valore atteso dello stato all'istante t condizionato alla sequenza di osservazioni Z^t effettuate fino a quell'istante.

Nel §1.1.2 sono stati descritti i modelli di stato e osservazione a tempo continuo. Tuttavia il filtro di Kalman essendo implementato in un calcolatore, è costruito per passi discreti di tempo in modo iterativo, perciò anche i modelli di stato e osservazione devono essere trasformati in forma discreta prima dell'implementazione del filtro. Supponendo passi di integrazione (campionamento, per la misura), costanti, si utilizzano le equazioni di modello (1.5) e (1.6).

L' ipotesi di base per il filtro di Kalman in cui si assume che il rumore del processo di evoluzione dello stato e il rumore di misura siano a *media nulla* e *temporalmente scorrelati* si traduce rispettivamente con le equazioni

$$E\{v_k\} = E\{w_k\} = 0, \ \forall k$$

$$E\{v(i)v(j)^T\} = \delta_{ij} \cdot Q(i), \quad E\{w(i)w(j)^T\} = \delta_{ij} \cdot R(i)$$

in cui l'operatore $E\{\cdot\}$ è quello che calcola il valore atteso di una variabile casuale (ovvero la media). δ_{ij} è l'operatore di Kronecker e indica che le *matrici di covarianza* Q *e* R, rispettivamente del rumore v e w, sono non nulle solo se il valore atteso è calcolato rispetto a un solo istante temporale i e non a due istanti diversi.

Si indica con la stima dello stato all'istante k condizionata a tutte le informazioni ottenute fino all'istante k. La stima dello stato all'istante k condizionata alle sole informazioni ottenute fino all'istante $k-1$ è detta *previsione* di un passo in avanti e si indica con $\hat{x}(k|k-1)$, questa simbologia è estesa a tutti gli altri parametri dell'algoritmo. Si assume che al passo k-esimo siano noti i seguenti parametri:

- la stima $\hat{x}(k-1|k-1)$
- la covarianza condizionata (varianza per variabili scalari) $P(k-1|k-1)$

Il filtro di Kalman consiste nella ricorsione di due stadi (S. Blackman and R. Popoli.):

Previsione

Si calcola una previsione $\hat{x}(k|k-1)$ dello stato all'istante k e la sua covarianza $P(k|k-1)$:

$$\hat{x}(k|k-1) = F(k)\hat{x}(k-1|k-1) + B(k)u(k) \quad (1.22)$$

$$P(k|k-1) = F(k)P(k-1|k-1)F^T(k) + G(k)Q(k)G^T(k) \quad (1.23)$$

Aggiornamento

All'istante k viene resa disponibile un'osservazione $z(k)$. Si calcola dunque la stima aggiornata $\hat{x}(k|k)$ dello stato $x(k)$ insieme alla covarianza aggiornata $P(k|k)$ a partire dalla previsione appena calcolata:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + W(k)[z(k) - H(k)\hat{x}(k|k-1)] \quad (1.24)$$

$$P(k|k) = [1 - W(k)H(k)]P(k|k-1)[1 - W(k)H(k)]^T + W(k)R(k)W^T(k) \quad (1.25)$$

dove la matrice guadagno $W(k)$ è data da

$$W(k) = P(k|k-1)H(k)[H(k)P(k|k-1)H^T(k) + R(k)]^{-1} \quad (1.26)$$

Il filtro di Kalman è un ciclo ricorsivo (vedi Figura 8). La condizione di partenza è una stima ottenuta in qualche modo attraverso le conoscenze che si hanno del misurando fino all'istante iniziale del ciclo, dopo di che si effettua una previsione, in ultimo si utilizza la misura per aggiornare la previsione e produrre una nuova stima .

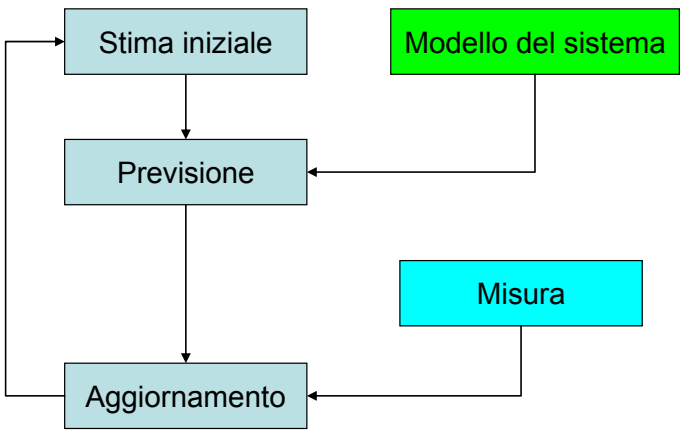


Figura 8: Schema logico della ricorsione del filtro di Kalman

La previsione è generata dal filtro utilizzando il modello di evoluzione del misurando mentre l'aggiornamento fa uso del modello di misura. Lo stadio di aggiornamento è quello di un tipico filtro lineare del primo ordine: si calcola una media pesata tra la previsione e la misura, in cui il peso $W(k)$ è associato alla misura $z(k)$ e il peso $1-W(k)H(k)$ è associato alla previsione $\hat{x}(k\,|\,k-1)$. Inoltre il filtro propaga la covarianza sia nella previsione, propagando l'incertezza dovuta al rumore associato al misurando (un semplice esempio è l'Equazione (1.18)), sia nell'aggiornamento della stima, propagando l'incertezza dovuta al rumore associato alla misura.

1.1.6.2 Misura delle prestazioni del filtro di Kalman

Per monitorare le prestazioni del filtro di Kalman si deve utilizzare una misura che sia indicativa di quanto bene il filtro riesce a "seguire" l'evoluzione dello stato nel tempo. E' possibile effettuare una previsione di quale osservazione sarà fatta all'istante k basata sulle osservazioni fatte fino all'istante $k-1$ calcolando il valore atteso dell'osservazione data dal modello ad essa associato (Equazione (1.6)) condizionatamente alle osservazioni precedenti:

$$\begin{aligned}\hat{z}(k\,|\,k-1) &= E\left\{z(k)\,|\,Z^{k-1}\right\}\\ &= E\left\{H(k)x(k)+W(k)\,|\,Z^{k-1}\right\}\\ &= E\left\{H(k)x(k)\,|\,Z^{k-1}\right\}\\ &= H(k)E\left\{x(k)\,|\,Z^{k-1}\right\}\\ &= H(k)\hat{x}(k\,|\,k-1)\end{aligned}$$

quindi, come si poteva intuire, la previsione dell'osservazione al passo k si ottiene semplicemente applicando il modello di osservazione alla previsione dello stato.

La differenza tra l'osservazione $z(k)$ e la previsione dell'osservazione $H(k)\hat{x}(k\,|\,k-1)$ è chiamata **innovazione** o residuo $\nu(k)$:

$$\rho(k) = z(k) - H(k)\hat{x}(k\,|\,k-1) \tag{1.27}$$

Non essendo possibile comparare direttamente le stime ottenute con il filtro e il valore vero dello stato, poiché molto spesso questo valore non lo si conosce, l'innovazione è un'importante misura della deviazione tra le stime del filtro e le osservazioni e può essere utilizzata come strumento per valutare l'efficienza del filtro. In particolare, ai fini della verifica delle ipotesi su cui si basano i modelli di stato e di osservazione e lo stesso filtro, si può verificare se sono soddisfatte le principali *proprietà delle innovazioni,* cioè che

$$E\left\{\rho(k)\,|\,Z^{k-1}\right\} = 0$$

$$E\left\{\rho(i)\rho^{T}(j)\right\} = \delta_{ij}S(i)$$

rispettivamente le due equazioni indicano che la sequenza delle innovazioni è bianca e scorrelata nel tempo. La matrice $S(k)$ è detta covarianza dell'innovazione e vale

$$S(k) = R(k) + H(k)P(k\,|\,k-1)H(k)^{T}$$

L'innovazione e la covarianza dell'innovazione possono esser utilizzate per esprimere in modo alternativo e più sintetico le equazioni di aggiornamento:

$$\hat{x}(k\,|\,k) = \hat{x}(k\,|\,k-1) + W(k)\rho(k)$$

$$P(k \mid k) = P(k \mid k-1) - W(k)S(k)W^T(k)$$

Dove

$$W(k) = P(k \mid k-1)H(k)S^{-1}(k)$$

Le equazioni di aggiornamento appena scritte si presentano nella stessa forma di quelle del semplice caso scalare a valori continui (§Esempio 5) (Equazioni (1.19), (1.20) e (1.21)), in cui la matrice di osservazione è la matrice identità, al posto della matrice P compare σ_x^2, al posto di R compare σ_z^2, al posto della matrice guadagno W compare il guadagno K.

Capitolo 2

Applicazione di sensor fusion per agv

2.1 Algoritmo di fusione per navigazione odometrico-inerziale in funzione della manovra attuale del veicolo

Sensori come encoder e piattaforme inerziali basate su giroscopi hanno caratteristiche molto diverse in quanto ad accuratezza di misura in relazione alla manovra attuale compiuta dal veicolo autonomo (Shoval). È possibile quindi pensare di utilizzare tali sensori per ottenere una maggior accuratezza nella stima della posa, e limitare la propagazione dell'incertezza a un primo livello di sensor fusion, a cui poi va aggiunto una ulteriore misura proveniente da un sensore riferito all'ambiente, in modo da azzerare periodicamente la propagazione dell'incertezza dovuta alla correlazione temporale dei parametri.

Nell'algoritmo proposto, gli encoder e il giroscopio vengono combinati tenendo in considerazione la rispettiva incertezza, stimata in funzione della manovra attuale che viene classificata a partire dai dati stessi.

2.2 Algoritmo di fusione e considerazioni qualitative

Il veicolo autonomo utilizzato per lo sviluppo dell'algoritmo è fornito di tre ruote, il modello cinematico è cioè quello di un triciclo, le cui equazioni in forma discreta sono:

$$\begin{aligned} x_{k+1}^E &= x_k + \frac{2\pi}{n_0} n_k R \cos(\alpha_k + \alpha_0)\cos(\delta_k) \\ y_{k+1}^E &= y_k + \frac{2\pi}{n_0} n_k R \cos(\alpha_k + \alpha_0)\sin(\delta_k) \qquad (2.1) \\ \delta_{k+1}^E &= \delta_k + \frac{2\pi}{n_0} n_k R \sin(\alpha_k + \alpha_0)\frac{1}{b} \end{aligned}$$

$$\delta_{k+1}^G = \delta_k + T_C G(V_k^G) \qquad (2.2)$$

dove i simboli rappresentano:

- x^E, y^E sono le coordinate del punto di riferimento P_r del veicolo, stimate con gli encoder, rispetto a un sistema di riferimento fisso
- δ^E è l'assetto del veicolo rispetto a un sistema di riferimento fisso, stimato cogli encoder
- δ^G è l'assetto del veicolo rispetto a un sistema di riferimento fisso, stimato con il giroscopio
- n_0 è il numero di conteggi in un giro dell'encoder che misura lo spostamento
- n_K è il numero effettivo di conteggi in un tempo di ciclo dell'encoder che misura lo spostamento
- α_0 è l'angolo di sterzo quando il centro di istantanea rotazione (ICR) è a infinito, cioè il cosiddetto angolo di zero
- α_k è l'angolo di sterzo rispetto a α_0
- R è il raggio della ruota anteriore, sterzante
- V^G è il voltaggio in output dal giroscopio
- $G()$ è la caratteristica del giroscopio
- T_C è il periodo di campionamento

Due sistemi di navigazione equipaggiano il veicolo:

- un sistema odometrico che utilizza due encoder sulla ruota di sterzo: uno misura l'angolo di sterzo, l'altro misura l'incremento angolare
- un sistema inerziale che utilizza un giroscopio per stimare l'assetto del veicolo

L'algoritmo di sensor fusion implementato fonde le misure ottenute dagli encoder e dal giroscopio tenendo conto dell'incertezza come funzione della manovra attuale del veicolo. La manovra viene riconosciuta basandosi sui dati stessi. L'algoritmo prevede i seguenti passi:

1. calcolo degli incrementi di posizione coi dati provenienti dagli encoder e dal giroscopio

2. riconoscimento del tipo di manovra attualmente compiuta dal veicolo secondo la Tabella 1

3. stima del rapporto tra le incertezze degli output forniti dai due sistemi di navigazione

4. fusione dei due incrementi in relazione al rapporto tra le incertezze

5. aggiornamento della posa del veicolo: l'output dell'algoritmo di sensor fusion viene sommato a δ_k per ottenere δ_{k+1}:

$$\delta_{k+1} = \delta_k + \Delta\delta_k$$

Per formalizzare l'algoritmo di sensor fusion è necessario considerare gli incrementi di posa:

$$\begin{aligned} \Delta x_k^E &= x_{k+1}^E - x_k \\ \Delta y_k^E &= y_{k+1}^E - y_k \\ \Delta \delta_k^E &= \delta_{k+1}^E - \delta_k \\ \Delta \delta_k^G &= \delta_{k+1}^G - \delta_k \end{aligned} \tag{2.3}$$

e anche l'accuratezza degli incrementi di assetto definita come $\pm \varepsilon_{\Delta\delta}^{\eta}$ dove η può essere *E* o *G*.

Utilizzando gli incrementi si arriva alle seguenti equazioni:

$$\begin{aligned} x_{k+1} &= x_k + \Delta x_k^E \\ y_{k+1} &= y_k + \Delta y_k^E \\ \delta_{k+1} &= \delta_k + DF(\Delta\delta_k^E, \Delta\delta_k^G) \end{aligned} \tag{2.4}$$

dove *DF* sta per Data Fusion.

Assumendo una distribuzione Gaussiana delle incertezze $\Delta\delta_k^E$ e $\Delta\delta_k^G$, la migliore stima in senso statistico dell'incremento di assetto che tenga conto degli incrementi provenienti dagli encoder e dal giroscopio è data dalla seguente equazione:

$$DF(\Delta\delta_k^E, \Delta\delta_k^G) = \frac{(\sigma_{\Delta\delta}^E)^2 \Delta\delta_k^G + (\sigma_{\Delta\delta}^G)^2 \Delta\delta_k^E}{(\sigma_{\Delta\delta}^E)^2 + (\sigma_{\Delta\delta}^G)^2} \tag{2.5}$$

che si può semplificare considerando solo il rapporto ξ_R tra le due incertezze:

$$\xi_R = \frac{\sigma_{\Delta\delta}^E}{\sigma_{\Delta\delta}^G}$$

nella seguente formula:

$$\Delta\delta_k = \frac{\Delta\delta_k^G}{\frac{1}{\xi_R^2}+1} + \frac{\Delta\delta_k^E}{\xi_R^2+1} \tag{2.6}$$

Per poter combinare i sistemi relativi, che sono affetti da deriva, con un sistema riferito all'ambiente, la stima dell'incertezza deve essere conosciuta ad ogni passo (S. I. Roumeliotis, 1999), (Y. K. Tham, 1999). L'incertezza è espressa in termini della matrice di covarianza del vettore posa $X_k = [x_k, y_k, \delta_k]$. Per ottenere la stima della covarianza, l'equazione (2.4) si può riscrivere come:

$$X_{k+1} = X_k + \Phi(\omega_k) \tag{2.7}$$

dove Φ è una funzione non lineare di $[n_k, n_0, R, b, \alpha_k + \alpha_0, G(V_k), \delta_k]$, dei quali n_0 e b sono considerati costanti. Il parametro n_0 è ovviamente costante poiché è il numero di impulsi dell'encoder per ogni rotazione. Il parametro b è la distanza tra l'asse di rotazione della ruota sterzante e l'asse delle ruote posteriori, ed è considerato costante perché nelle maggior parte delle applicazioni in robotica vengono usate strutture rigide, che in teoria potrebbero variare la distanza tra gli assi ma nella pratica l'effetto si può ritenere trascurabile, e non venire quindi considerato rispetto ad altri effetti. Il parametro n_k può essere tralasciato come fonte di incertezza poiché gli errori di lettura che vengono compiuti in un certo ciclo di campionamento vengono compensati in quello successivo, purchè il periodo di campionamento T_C sia sufficientemente piccolo se paragonato alla dinamica del sistema. In base a queste considerazioni, il vettore ω_k delle variabili che influenzano l'accuratezza nella stima della posizione e dell'assetto si può ritenere composto nel seguente modo:

$$\omega_k = [R, \alpha_k + \alpha_0, G(V_k), \delta_k] \tag{2.8}$$

L'equazione (2.7) si può utilizzare per valutare l'incertezza standard della stima X_{k+1} se sono conosciute le incertezze delle stime X_k e ω_k. L'incertezza può essere espressa utilizzando la matrice covarianza:

$$C_{Xk+1} = C_{Xk} + \Im_{\Phi k} C_{\omega k} \Im_{\Phi k}^T + \Im_{\Phi k} S_{\omega k} I_k^T + I_k S_{\omega k} \Im_{\Phi k}^T \tag{2.9}$$

dove il termine I_k può essere calcolato utilizzando la ricorsione:

$$I_k = I_{k-1} + \Im_{\Phi k-1} S_{\omega k-1} \tag{2.10}$$

e i vari termini sono definiti come segue:

$$C_{Xk} = \begin{bmatrix} \sigma_{xk}^2 & \sigma_{xy} & \sigma_{x\delta} \\ \sigma_{xy} & \sigma_{yk}^2 & \sigma_{y\delta} \\ \sigma_{x\delta} & \sigma_{y\delta} & \sigma_{\delta k}^2 \end{bmatrix} \tag{2.11}$$

$$C_{\omega k} = \begin{bmatrix} \sigma_R^2 & 0 & 0 & 0 \\ 0 & \sigma_{\alpha k}^2 & 0 & 0 \\ 0 & 0 & \sigma_G^2 & 0 \\ 0 & 0 & 0 & \sigma_{\delta k}^2 \end{bmatrix} \tag{2.12}$$

$$S_{\omega k}(i,j) = \sqrt{C_{\omega k}(i,j)} \qquad \forall i, \forall j \tag{2.13}$$

$$\Im_{\Phi k} = \begin{bmatrix} \frac{\partial \nabla x_k}{\partial R} & \frac{\partial \nabla x_k}{\partial \alpha_k} & \frac{\partial \nabla x_k}{\partial G} & \frac{\partial \nabla x_k}{\partial \delta_k} \\ \frac{\partial \nabla Y_k}{\partial R} & \frac{\partial \nabla Y_k}{\partial \alpha_k} & \frac{\partial \nabla Y_k}{\partial G} & \frac{\partial \nabla Y_k}{\partial \delta_k} \\ \frac{\partial \nabla \delta_k}{\partial R} & \frac{\partial \nabla \delta_k}{\partial \alpha_k} & \frac{\partial \nabla \delta_k}{\partial G} & \frac{\partial \nabla \delta_k}{\partial \delta_k} \end{bmatrix} \tag{2.14}$$

$$I_k = \begin{bmatrix} I_{XRk} & I_{X\alpha k} & I_{XGk} & I_{X\delta k} \\ I_{YRk} & I_{Y\alpha k} & I_{YGk} & I_{Y\delta k} \\ I_{\delta Rk} & I_{\delta\alpha k} & I_{\delta Gk} & I_{\delta\delta k} \end{bmatrix} \tag{2.15}$$

I vari termini $\sigma_{\delta k}$ devono essere aggiornati ad ogni iterazione nella matrice $C_{\omega k}$ dopo il calcolo di C_{Xk}.

Il primo termine dell'equazione (2.9) è la matrice covarianza del campione precedente. Il secondo termine è il contributo all'incertezza parametrica se i parametri non fossero correlati, cioè se $E\{\varepsilon_{\omega k}(i)\varepsilon_{\omega k}(j)\} = 0 \quad \forall i \neq j$, e se ogni parametro non fosse correlato con se stesso nel tempo (cioè non ci fossero effetti sistematici), quindi $E\{\varepsilon_{\omega k}(i)\varepsilon_{\omega h}(i)\} = 0 \quad \forall k \neq h$. Le incertezze standard espresse nella matrice $C_{\omega k}$ dipendono dalla manovra attuale che il robot sta compiendo, e si assume che siano scorrelate, portando quindi a una matrice ortogonale. Il terzo e il quarto termine considerano la piena correlazione dei campioni durante tutto il periodo di integrazione, cioè $E\{\varepsilon_{\omega k}(i)\varepsilon_{\omega h}(i)\}/(\sigma_{\omega k}(i)\sigma_{\omega h}(i)) = 1 \quad \forall k, \forall h$. In altre parole, questi ultimi due termini stimano la deriva in funzione del tempo. Su lunghi periodi questi due termini prevalgono, mentre per brevi intervalli sono i primi due termini a prevalere.

L'ipotesi di completa correlazione di ogni fattore di incertezza $\omega_k(i)$ con se stesso nel tempo, ma di non correlazione tra i diversi fattori deve essere spiegata per ogni fattore separatamente:

- è valida per R se si considera costante l'effetto del carico lungo tutto il percorso, mentre non vale se si considera l'effetto delle irregolarità del terreno sui raggi delle ruote
- la correlazione totale si può ritenere vera per $\alpha_k + \alpha_0$ solo quando si curva in una unica direzione, mentre sterzando prima in un senso e poi nell'altro si può introdurre una compensazione tale che considerando la correlazione totale si sovrastima l'incertezza sull'angolo
- si può ritenere vera per gli effetti sistematici di $G(V_k)$ se viene considerato costante durante il periodo di integrazione; in questo modo la parte casuale viene trascurata
- la correlazione vale per δ_k in tutto il periodo di integrazione, ma non è scorrelata dagli altri fattori di incertezza, cosicché l'incertezza dovuta a questo parametro risulta sottostimata.

2.3 Procedura di stima dell'incertezza

Per poter stimare l'incertezza, la matrice covarianza $C_{\omega k}$ deve essere caratterizzata in funzione delle differenti manovre che il veicolo può compiere durante un percorso. Si sono identificate quattro diverse manovre, come schematizzato in Tabella 1: veicolo fermo, tratto rettilineo, curve ad angolo di sterzo costante, curve ad angolo di sterzo variabile.

Tabella 1: riconoscimento della manovra effettuata in funzione dei dati degli encoder

Stato del veicolo	n_k	α_k	$\frac{\alpha_k - \alpha_{k-1}}{Tc}$
Stop	=0	Qualunque	Qualunque
Rettilineo	$\neq 0$	$< 10°$	$< 5°s^{-1}$
Curva con sterzo bloccato	$\neq 0$	$\geq 10°$	$< 5°s^{-1}$
Curva generica	$\neq 0$	Qualunque	$\geq 5°s^{-1}$

Considerazioni qualitative sull'accuratezza dei sensori in relazione alla manovra sono riassunte in Tabella 2.

Tabella 2: Considerazioni qualitative sulle incertezze dei sensori in base alla manovra effettuata.

Manovra del veicolo:	Accuratezza sistema odometrico: considerazione qualitativa	Accuratezza giroscopio: considerazione qualitativa
Stop	Alta	Bassa
Rettilineo	Alta	Bassa
Curva con sterzo bloccato	Media	Alta
Curva generica	Bassa	Alta

Gli elementi della matrice covarianza $C_{\omega k}$ sono stati stimati ad ogni passo secondo la seguente procedura

1. $\sigma_{\delta k}$ è stato stimato a partire dal campione precedente, chiaramente nell'istante iniziale il valore deve essere conosciuto
2. σ_R è stato stimato durante un percorso rettilineo e si è assunto costante indipendentemente dalla manovra attuale. Questa assunzione è stata fatta poiché durante traiettorie curvilinee le deformazioni dei pneumatici sono soprattutto torsionali portando quindi variazioni dei vari parametri α. Variazioni nel raggio delle ruote sono possibili, ma vengono considerate trascurabili. In generale, il raggio può subire dei cambiamenti in relazione a variazioni di carico, alle condizioni del pavimento e all'usura. In particolare l'usura porta a un effetto sistematico che deve essere compensato, ad esempio con una calibrazione on-line (De Cecco, 2002)
3. $\sigma_{\alpha_k+\alpha_0}$ è stato stimato in funzione della manovra. Questo valore insieme a σ_R è stato utilizzato per calcolare $\sigma^E_{\Delta\delta}$.
4. $\sigma_{G(Vk)}$ è stato stimato in funzione della manovra. È stato quindi utilizzato per calcolare $\sigma^G_{\Delta\delta}$.

2.3.1 Veicolo fermo

Quando il veicolo è fermo l'accuratezza del giroscopio risulta molto bassa a causa della deriva causata dall'offset. Gli encoder invece forniscono un output maggiormente accurato. Di conseguenza gli incrementi del giroscopio semplicemente non vengono considerati quando gli incrementi dell'encoder della ruota di guida sono zero. In base a queste considerazioni il rapporto tra le incertezze ξ_R è posto uguale a zero a veicolo fermo.

2.3.2 Tratti rettilinei

Durante tratti rettilinei o anche curvilinei con angolo di sterzo prossimo allo zero, le vibrazioni del sistema meccanico influenzano l'accuratezza del giroscopio e la sua uscita è comparabile con l'offset.

Per valutare la deviazione standard di σ_R, il robot ha percorso un tratto di 5 m per 20 volte, variando il carico in un range di valori possibili, e mantenendo lo sterzo allineato con l'angolo di zero. La posizione iniziale x_I e la posizione finale x_F sono state misurate utilizzando un laser a triangolazione, che è il sistema di riferimento. Il passi di integrazione sono definiti come *k=1…..N,* e σ_R è stato stimato con la formula:

$$\sigma_R = \left(\frac{2\pi}{n_0} \cdot \sum_{k=1}^{N} n_k \right)^{-1} \cdot \sigma_{xF-xI} \tag{2.16}$$

dove σ_{xF-xI} è la deviazione standard delle differenze tra le misure del sistema a triangolazione di riferimento e della stima odometrica.

Per valutare l'incertezza standard di $\sigma_{\alpha k+\alpha 0}$ e $\sigma_{G(Vk)}$, il robot ha compiuto un percorso rettilineo lungo 5 m per 10 volte, con un carico costante e mantenendo lo sterzo allineato con l'angolo di zero. L'assetto del veicolo iniziale δ_I e finale δ_F sono stati misurati con il sistema a triangolazione di riferimento, oltre che con gli encoder e il giroscopio. Calcolando le differenze rispetto alle misure di riferimento , sono state calcolate le deviazioni standard $\sigma^E_{\delta F-\delta I}$ e $\sigma^G_{\delta F-\delta I}$. Considerando la terza delle equazioni (2.1) è possibile calcolare $\sigma_{\alpha k+\alpha 0}$ e $\sigma_{G(Vk)}$:

$$\sigma_{\alpha k+\alpha 0} = \left(\frac{2\pi}{n_0} \cdot \frac{R}{b} \cdot \sum_{k=1}^{N} n_k \right)^{-1} \cdot \sigma^E_{\delta F-\delta I} \tag{2.17}$$

$$\sigma_{G(Vk)} = \left(N \cdot T_C \right)^{-1} \cdot \sigma^G_{\delta F-\delta I} \tag{2.18}$$

Sfruttando $\sigma_{\alpha k+\alpha 0}$ è possibile stimare $\sigma^E_{\Delta\delta}$ in condizioni di percorso rettilineo:

$$\left(\sigma^E_{\Delta\delta}\right)_k = \frac{2\pi}{n_0} \cdot \frac{R}{b} \cdot n_k \cdot \sigma_{\alpha k+\alpha 0} \tag{2.19}$$

Tramite $\sigma_{G(Vk)}$ è possibile calcoalre $\sigma^G_{\Delta\delta}$ in condizioni di percorso rettilineo:

$$\left(\sigma^G_{\Delta\delta}\right)_k = T_C \cdot \sigma_{G(Vk)} \tag{2.20}$$

È importante sottolineare che nella stima di $\sigma^G_{\Delta\delta}$ come mostrato nell'equazione (2.20) è stato ignorato l'effetto della velocità, che in realtà potrebbe essere non trascurabile. Questa semplificazione è stata fatta per due ragioni: la prima è che per la maggior parte delle applicazioni industriali in cui questo tipo di veicoli sono coinvolti, tratti rettilinei vengono percorsi a velocità costante e predeterminata, la seconda per semplificare l'approccio. Comunque, per certe particolari applicazioni potrebbe essere necessario stimare l'incertezza anche in funzione della velocità.

Durante le verifiche sperimentali effettuate con il prototipo di carrello elevatore, i valori di incertezza stimati sono stati $\sigma_R = 0.1\times10^{-3} m$, $\sigma_{\alpha k+\alpha 0} = 0.0013 rad$ e $\sigma_G = 0.0025\ rad\ s^{-1}$. Il rapporto tra le

incertezze standard ottenute con le equazioni (2.19) e (2.20) è stato calcolato ottenendo così il rapporto tra le incertezze ξ_R, circa uguale a 0.25.

A titolo di esempio si riporta in Figura 9 un output del giroscopio campionato durante un percorso rettilineo e la sua trasformata di Fourier (FFT). Dall'analisi spettrale è possibile notare la sensibilità del giroscopio alle vibrazioni indotte dalle irregolarità del pavimento, e anche che il valore medio è diverso da zero. L'incremento di assetto stimato col giroscopio è stato di 0.18°, mentre per il sistema di riferimento è stato di 0.05°.

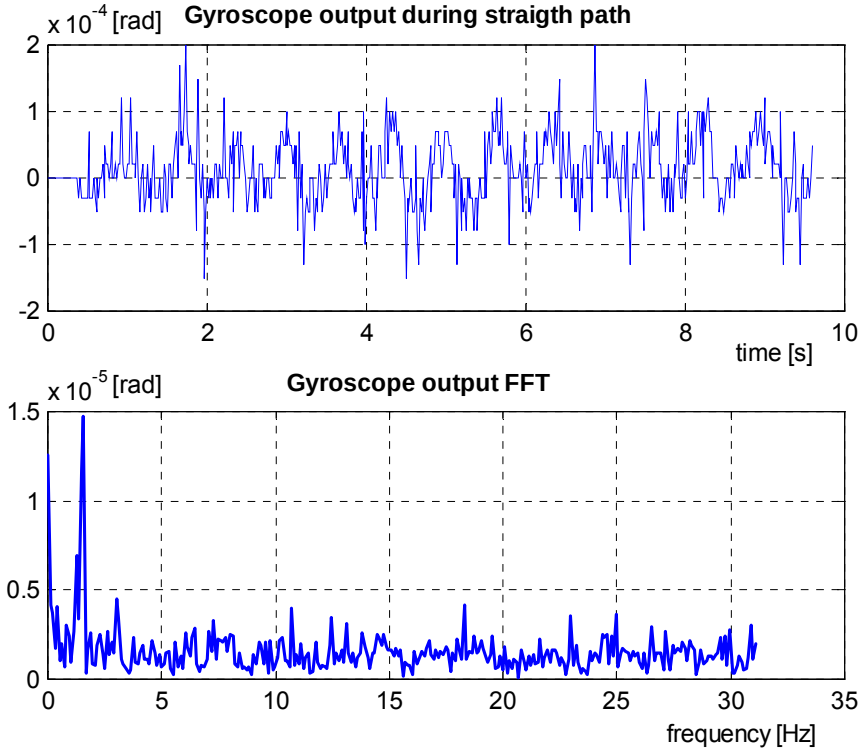


Figura 9: Output del giroscopio: incrementi misurati durante un tratto rettilineo con un tempo di campionamento di 16 ms (in alto), e FFT dell'output (in basso)

2.3.3 Curve ad angolo di sterzo costante

Per valutare l'incertezza di $\sigma_{\alpha k+\alpha 0}$ e $\sigma_{G(Vk)}$ durante la percorrenza di curve ad angolo di sterzo costante, sono stati compiute 10 traiettorie circolari con angolo di sterzo a 0.8 rad e 10 con angolo di sterzo a -0.8 rad. È da sottolineare il fatto che non sono state considerate variazioni in funzione di cambiamenti dell'angolo di sterzo, ma è stato considerato solamente un angolo di sterzo. Il carico è stato tenuto costante così come il raggio delle ruote. In ogni percorso circolare sono state portate a termine circa tre intere circonferenze. Al solito, l'assetto iniziale e finale è stato misurato con il sistema di riferimento a triangolazione laser, oltre che con encoder e giroscopio. Calcolando le differenze rispetto alle misure di riferimento , sono state calcolate le deviazioni standard $\sigma^E_{\delta F-\delta I}$ e $\sigma^G_{\delta F-\delta I}$. Considerando la terza delle equazioni (2.1) è possibile calcolare $\sigma_{\alpha k+\alpha 0}$ e $\sigma_{G(Vk)}$:

$$\sigma_{\alpha k+\alpha 0} = \left(\frac{2\pi}{n_0} \cdot \frac{R}{b} \cdot \sum_{k=1}^{N} \left(n_k \cdot \cos(\alpha_k + \alpha_0) \right) \right)^{-1} \cdot \sigma^E_{\delta F-\delta I} \tag{2.21}$$

$$\sigma_{G(Vk)} = \left(N \cdot T_C \right)^{-1} \cdot \sigma^G_{\delta F-\delta I} \tag{2.22}$$

Sfruttando $\sigma_{\alpha k+\alpha 0}$ e σ_R è possibile stimare $\sigma^E_{\Delta\delta}$ in condizioni di traiettoria curva:

$$\left(\sigma^{E}_{\Delta\delta}\right)_k = \sqrt{\left[\frac{2\pi}{n_0}\cdot\frac{n_k}{b}\cdot R\cdot\cos\left(\alpha_k+\alpha_0\right)\right]^2\cdot\sigma^2_{\alpha k+\alpha 0}+\left[\frac{2\pi}{n_0}\cdot\frac{n_k}{b}\cdot sin\left(\alpha_k+\alpha_0\right)\right]^2\cdot\sigma^2_R} \quad (2.23)$$

e sfruttando $\sigma_{G(Vk)}$ è possibile stimare $\sigma^{G}_{\Delta\delta}$ in condizioni di traiettoria curva:

$$\left(\sigma^{G}_{\Delta\delta}\right)_k = T_C\cdot\sigma_{G(Vk)} \quad (2.24)$$

Durante le traiettorie curve ad angolo costante le incertezze stimate col prototipo di carrello elevatore sono $\sigma_R = 0.1\times10^{-3}m$, $\sigma_{\alpha k+\alpha 0} = 0.00126\ rad$ e $\sigma_G = 0.0025\ rad\ s^{-1}$ (i calcoli sono stati fatti alla velocità nominale e all'angolo di sterzo nominale). Calcolate le incertezze con le equazioni (2.23) e (2.24) è stato calcolato il rapporto tra le incertezze ξ_R, circa uguale a 2.

2.3.4 Traiettorie curve ad angolo di sterzo variabile

Durante manovre di curvatura ad angolo di sterzo variabile, l'accuratezza del giroscopio è maggiore (più che durante curve ad angolo costante) che quella della stima effettuata con gli encoder, che soffre delle deformazioni delle ruote. Tali deformazioni durante le curve influenzano l'accuratezza della stima odometrica poiché le torsioni delle ruote fanno si che l'assetto attuale sia diverso da quello fornito dall'encoder della ruota sterzante (l'assetto reale risulta ritardato rispetto ai comandi di sterzatura). Questo effetto è messo in evidenza in Figura 10 dove è raffigurato il rapporto tra gli assetti stimati per via odometrica e dal giroscopio, e l'angolo di sterzo in funzione del tempo.

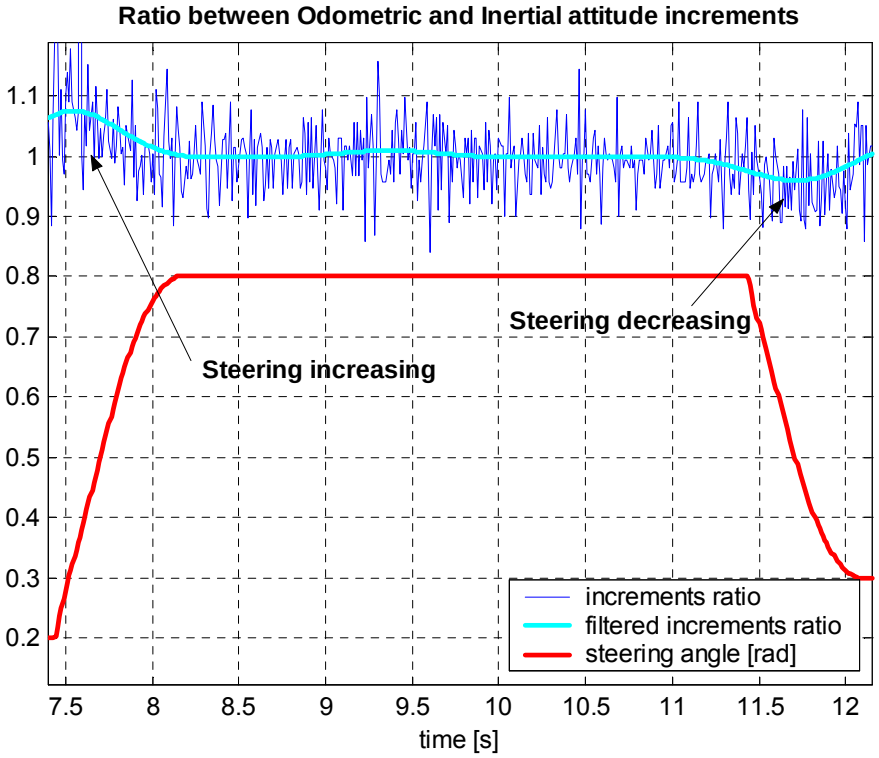


Figura 10: incrementi degli encoder rispetto a quelli del giroscopio e angolo di sterzo durante una curva di 90°

Si può notare come, quando la velocità di sterzatura è bassa o nulla, le due stime hanno valori medi molto simili e quindi si ottiene un rapporto quasi unitario. Quando la velocità di curvatura è diversa da zero le stime odometriche differiscono da quelle ottenute col giroscopio: sono maggiori mentre l'angolo di sterzo cresce e minori quando l'angolo diminuisce. Va sottolineato che questo effetto, che si può ritenere sistematico e costante, dipende in realtà dalla temperatura, dagli attriti, dall'usura e dall'età del sistema. Quindi è stato considerato nell'incertezza ma non caratterizzato (Guide to Expression of Uncertainty in Measurements 1993).

Il rapporto tra le incertezze dipende dalla velocità angolare della sterzatura e ha la tendenza a compensarsi quando il veicolo sterza da una parte e poi dall'altra. Per stimare l'incertezza standard di $\sigma_{\alpha k+\alpha 0}$ e $\sigma_{G(Vk)}$ si è eseguito il percorso raffigurato in Figura 11 per 20 volte. L'assetto è stato stimato all'inizio e alla fine della traiettoria sia col sistema di riferimento che con encoder e giroscopio. Per evitare l'effetto di compensazione, la sterzatura è stata effettuata 10 volte velocemente a destra e poi lentamente a sinistra, e poi viceversa per 10 volte.

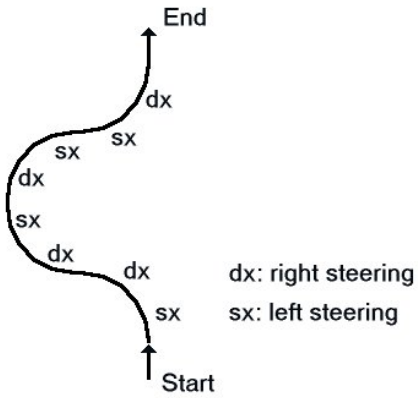


Figura 11: traiettoria percorsa per stimare l'incertezza durante curve ad angolo di sterzo non costante

Calcolando le differenze rispetto alle misure di riferimento , sono state calcolate le deviazioni standard $\sigma^{E}_{\delta F-\delta I}$ e $\sigma^{G}_{\delta F-\delta I}$. Considerando la terza delle equazioni (2.1) è possibile calcolare $\sigma_{\alpha k+\alpha 0}$ e $\sigma_{G(Vk)}$:

$$\sigma_{\alpha k+\alpha 0} = \left(\frac{2\pi}{n_0} \cdot \frac{R}{b} \cdot \sum_{k=1}^{N} \left(n_k \cdot \cos(\alpha k + \alpha 0) \right) \right)^{-1} \cdot \sigma^{E}_{\delta F-\delta I} \tag{2.25}$$

$$\sigma_{G(Vk)} = \left(N \cdot T_C \right)^{-1} \cdot \sigma^{G}_{\delta F-\delta I} \tag{2.26}$$

Sfruttando $\sigma_{\alpha k+\alpha 0}$ e σ_R è possibile stimare $\sigma^{E}_{\Delta\delta}$ in condizioni di sterzatura:

$$\left(\sigma^{E}_{\Delta\delta}\right)_k = \sqrt{\left[\frac{2\pi}{n_0} \cdot \frac{n_k}{b} \cdot R \cdot \cos(\alpha_k + \alpha_0) \right]^2 \cdot \sigma^2_{\alpha k+\alpha 0} + \left[\frac{2\pi}{n_0} \cdot \frac{n_k}{b} \cdot sin(\alpha_k + \alpha_0) \right]^2 \cdot \sigma^2_R} \tag{2.27}$$

e sfruttando $\sigma_{G(Vk)}$ è possibile stimare $\sigma^{G}_{\Delta\delta}$ in condizioni di traiettoria curva:

$$\left(\sigma^{G}_{\Delta\delta}\right)_k = T_C \cdot \sigma_{G(Vk)} \tag{2.28}$$

Durante le manovre di sterzatura le incertezze stimate col prototipo di carrello elevatore sono $\sigma_R = 0.1\times10^{-3}\, m$, $\sigma_{\alpha k+\alpha 0} = 0.0251\ rad$ e $\sigma_G = 0.0025\ rad\ s^{-1}$ (i calcoli sono stati fatti alla velocità nominale e all'angolo di sterzo e velocità di sterzo nominali). Calcolate le incertezze con le equazioni (2.27) e (2.28) è stato calcolato il rapporto tra le incertezze ξ_R, circa uguale a 4.

2.4 Verifiche sperimentali

Le verifiche sperimentali sono state compiute utilizzando il prototipo di carrello elevatore mostrato in Figura 12. La ruota motrice e sterzante è quella anteriore, guidata da due motori dc. Il veicolo è dotato di due encoder incrementali e un giroscopio a fibra ottica. L'encoder sull'asse della ruota ha 625 impulsi per giro (ppr), l'encoder che misura l'angolo di sterzo ha 10000 impulsi per giro, e il giroscopio ha una sensibilità nominale di 55 [mV $\deg^{-1}$ s^{-1}]. Il sistema di controllo e navigazione è affidato a un PC industriale inserito

in un sistema PXI fornito di un modulo analogico e un modulo digitale. La scheda analogica legge il giroscopio e fornisce i riferimenti di velocità e angolo di sterzo ai motori, la scheda digitale legge gli output degli encoder.

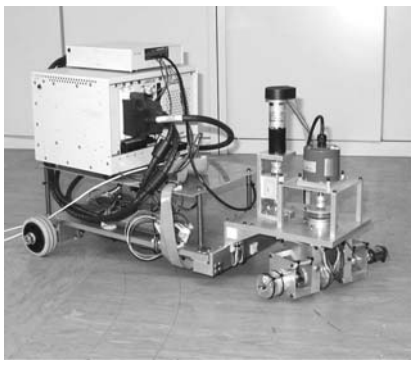


Figura 12: prototipo di veicolo industriale per la movimentazione merci con cinematica a tre ruote (triciclo)

Per verificare l'algoritmo di data fusion, è stato utilizzato come riferimento un sistema a triangolazione laser. L'incertezza angolare è ± 47 arcsec, e l'accuratezza della posizione al centro di una stanza quadrata di 5 m di lato è ± 1 mm (livello di confidenza $\pm 2\sigma$).

Sono state effettuate 20 traiettorie rettangolari sia in senso orario che antiorario alla velocità di 0.25 m/s, come mostrato in Figura 13.

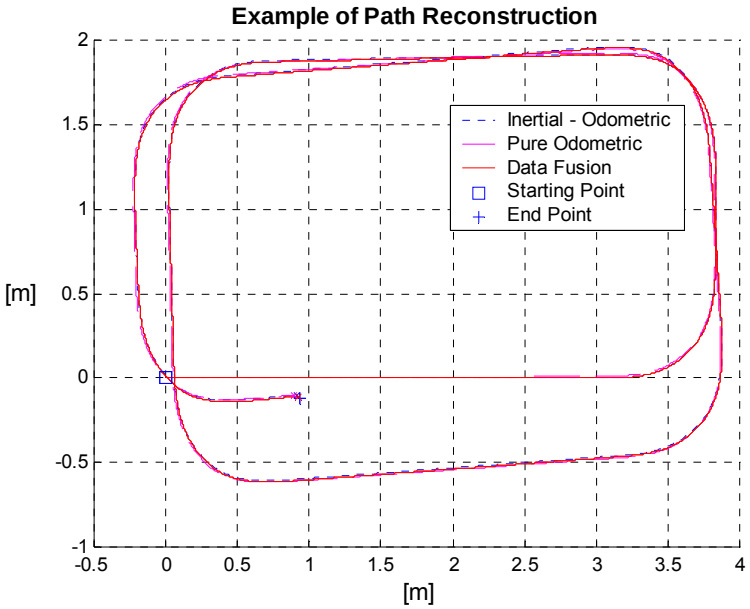


Figura 13: traiettorie rettangolari ricostruite coi singoli sensori e con l'algoritmo di data fusion, alla velocità di 0.25 m/s

Le prove sono state eseguite con il carico e la corrente di alimentazione dei motori tenuti costanti in modo da permettere la percorrenza della stessa traiettoria come velocità e sterzatura. Le posizioni iniziali e finali sono state stimate utilizzando il sistema a triangolazione e poi comparate con le stime effettuate con gli encoder, con il giroscopio e con l'algoritmo di data fusion. Le distanze tra i punti finali dei tre sistemi di navigazione e il sistema di riferimento rappresenta lo scarto stimato. Nel percorso mostrato, come in ogni punto finale stimato, il miglioramento dato dalla tecnica proposta è chiaro, come si può vedere nel dettaglio mostrato in Figura 14.

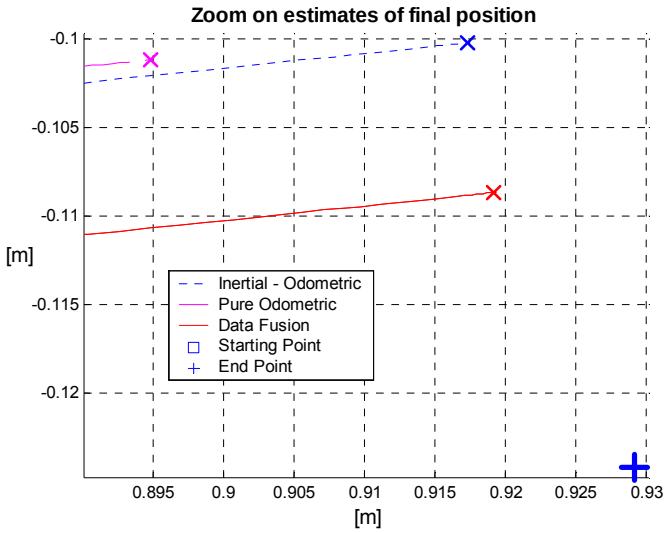


Figura 14: dettaglio sui punti finali stimati per via odometrica, inerziale più odometrica, algoritmo di data fusion. Il punto finale è stato misurato col sistema a triangolazione di riferimento.

Nell'istogramma in Figura 15 sono stati messi in evidenza gli scarti stimati nei punti finali per i tre sistemi di navigazione.

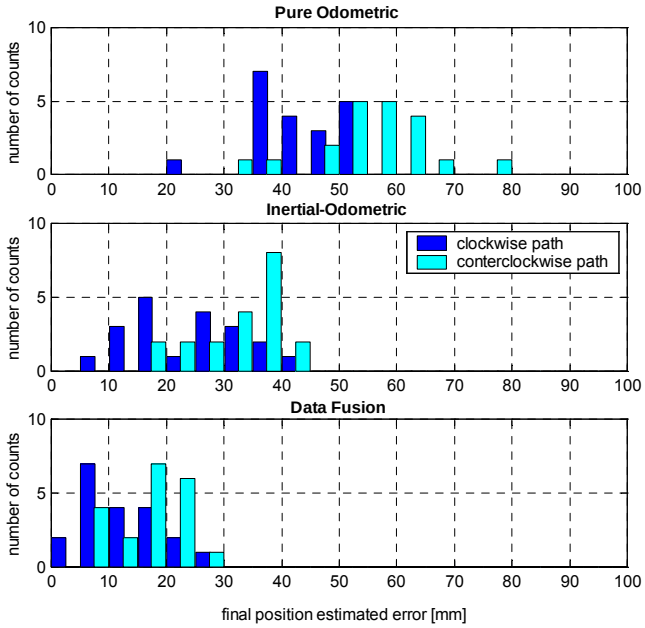


Figura 15: istogramma del valore assoluto degli scarti stimati nel punto finale dalla stima puramente odometrica, odometrica più inerziale e dall'algoritmo di data fusion

Si può notare una distribuzione binormale degli scarti stimati, che è dovuta a un'accuratezza minore nei percorsi in senso antiorario rispetto a quelli in senso orario. Ciò potrebbe essere spiegato considerando il fenomeno di azzeramento dell'angolo di zero (il cosiddetto autozero implementato nelle schede digitali) che risulta leggermente differente quando l'asse di sterzo incontra lo zero in senso orario e in senso antiorario, oppure considerando i residui della calibrazione (De Cecco, 2002), (ISO 1993).

Il valore medio assoluto de gli scarti stimati su distanze di oltre 25 metri è stato di 49 mm per la navigazione odometrica, di 28 mm per la navigazione odometrica più inerziale, e di 15 mm per la navigazione basata sull'algoritmo di data fusion proposto.

In Figura 16 sono raffigurate le ellissi di incertezza causate dagli effetti sistematici spiegati in precedenza, in funzione del percorso (le ellissi sono tracciate tenendo in considerazione le prime due righe e le prime due colonne della matrice C_{Xk}). L'aumento della correlazione con l'avanzare della distanza percorsa è chiaramente visibile. Inoltre l'incremento della correlazione che risulta in una ellisse molto schiacciata,

quindi con una direzione preferenziale, è causato dal fatto che i parametri di incertezza influenzano gli scarti sulla stima finale secondo una stessa direzione. Ciò si può verificare facilmente misurando il punto finale della traiettoria variando un parametro di influenza per volta intorno al valore nominale in un intervallo di due volte la deviazione standard: la figura geometrica ottenuta sarà molto simile, con un ellisse rivolta sempre nella stessa direzione.

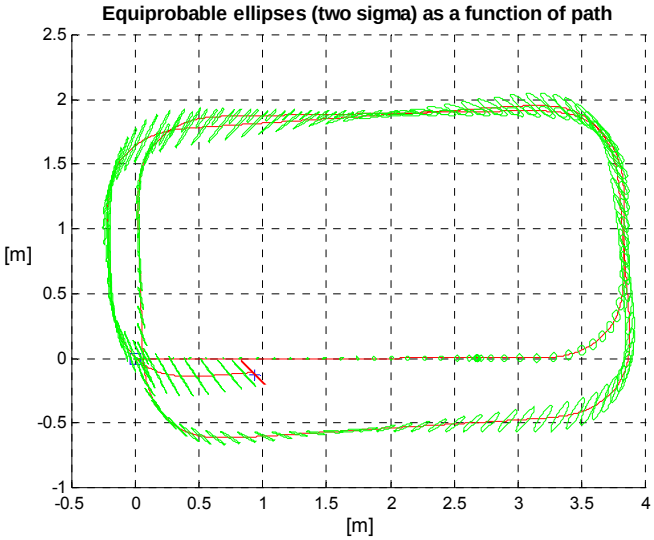


Figura 16: ricostruzione del percorso con l'algoritmo di data fusion in cui sono evidenziate le ellissi di incertezza in funzione del percorso

Per verificare la validità dell'algoritmo proposto nel valutare l'ellisse di incertezza, sono state effettuate ripetute traiettorie uguali cambiando il carico e la velocità: nella prima il carico era nominale e la velocità di 0.25 m/s, nella seconda il carico era quello nominale aumentato di 10 kg e la velocità di 0.25 m/s, nella terza il carico era nominale e la velocità di 0.5 m/s, nella quarta il carico era quello nominale aumentato di 10 kg e la velocità di 0.5 m/s. In Figura 17 sono raffigurati i punti finali stimati con l'algoritmo di data fusion e con il sistema di riferimento e l'ellisse di incertezza calcolata con un intervallo di ± 2 volte l'incertezza standard considerando una distribuzione gaussiana.

In Figura 18 è rappresentato un ingrandimento di una traiettoria ricostruita con l'algoritmo di data fusion e l'ellisse di incertezza in funzione del percorso. Si può notare una notevole variazione nella forma geometrica e nell'orientazione dell'ellisse lungo il percorso. Si evidenzia cioè la necessità di stimare la covarianza dell'incertezza e non solo le deviazioni standard σ_x e σ_y, il che potrebbe portare a una fusione non accurata con un sistema riferito all'ambiente.

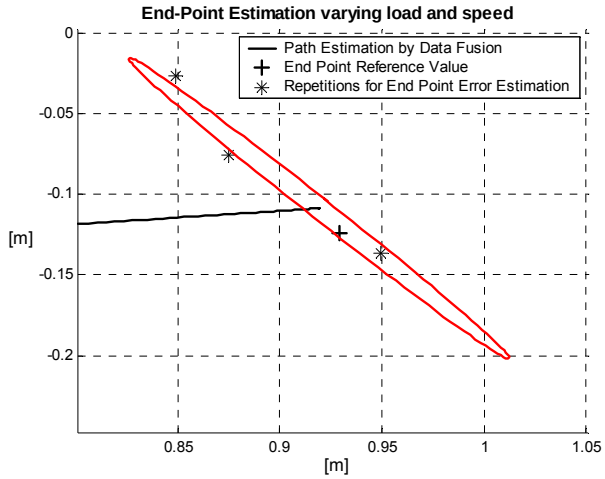


Figura 17: stima del percorso utilizzando l'algoritmo di data fusion (curva continua); punto finale misurato col sistema di riferimento (+); punti finali (*) di traiettorie ripetute per stimare lo spostamento finale rispetto alla misura di riferimento

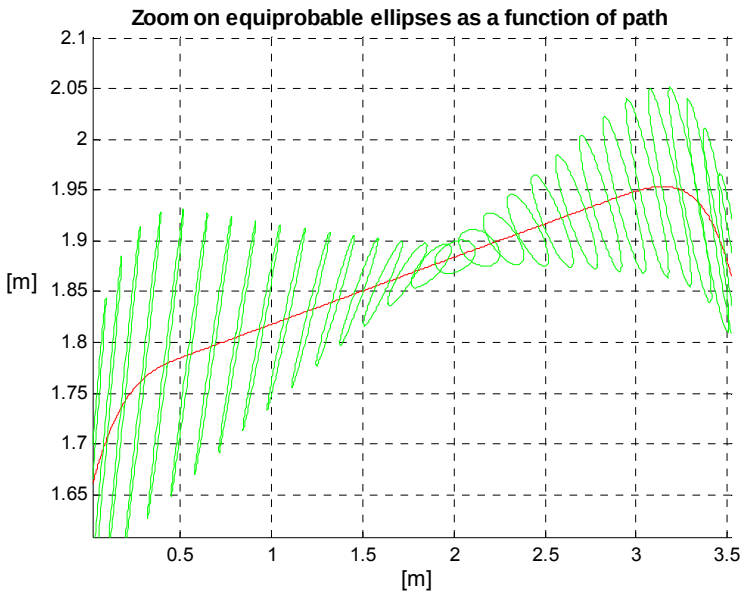


Figura 18: dettaglio di una traiettoria ricostruita con l'algoritmo di data fusion, con evidenziata l'ellisse di incertezza in funzione del percorso

2.5 Determinazione delle caratteristiche metrologiche del giroscopio laser a fibra ottica: aliasing e filtraggio

Preliminare all'implementazione dell'algoritmo di navigazione odometrico-inerziale su un prototipo di veicolo industriale, è stato necessario effettuare la taratura del giroscopio, con particolare attenzione al problema dell'aliasing e di un eventuale filtraggio.

Le traiettorie del veicolo su pavimento industriale presumibilmente indurranno un ripple sulla misura del giroscopio che avrà componenti armoniche superiori ai 50 Hz. Poiché il task di ricostruzione odometrico-inerziale avrà presumibilmente un tempo di ciclo di circa 10 ms e l'acquisizione del giroscopio avrà una frequenza di acquisizione di circa 100 Hz, si potrebbero avere dei problemi con le armoniche a più di 50 Hz.

Introducendo un filtro anti-aliasing si ridurrà l'effetto del ripple sulla misura giroscopica ma si potrebbe rischiare di introdurre un ritardo nella stima dell'assetto con conseguenze sulla ricostruzione incrementale sia di posizione che di assetto.

2.5.1 Acquisizioni ed elaborazione dati

Sono state effettuate alcune traiettorie campionando a 10000 Hz il solo giroscopio per valutarne le caratteristiche in frequenza. Di altre ripetizioni delle stesse traiettorie, Figura 19, sono stati acquisiti a 1000 Hz i dati forniti dal laser a triangolazione, utilizzato come sistema di riferimento, e da encoder e giroscopio.

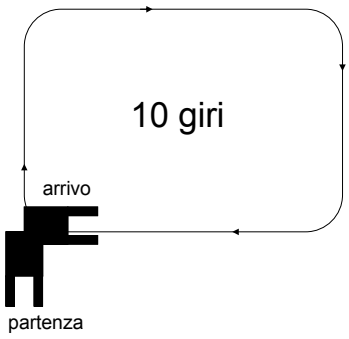


Figura 19: traiettoria per taratura giroscopio

2.5.1.1 Determinazione del fattore di taratura del giroscopio

Per determinare il fattore di taratura del giroscopio si è creato un funzionale che minimizza gli scarti quadratici delle differenze tra i dati di assetto forniti dal laser a triangolazione e la ricostruzione incrementale dell'assetto ad opera del giroscopio. Il funzionale tiene conto della velocità angolare del veicolo per pesare diversamente i residui quadratici: il sistema di triangolazione utilizzato come riferimento fornisce una stima della posa con un'accuratezza peggiore durante la percorrenza di curve mentre, come visto in precedenza, il giroscopio offre un'elevata accuratezza proprio durante manovre di curve o sterzata.

2.5.1.2 Determinazione dell'effetto dell'aliasing

Per determinare l'effetto dell'aliasing la procedura è stata di simularne le conseguenze sottocampionando a 100 Hz i dati acquisiti a 10 KHz, ed integrando le variazioni di assetto a partire dal segnale originario e poi dai dati ottenuti con la sottocampionatura. Occorre inoltre massimizzare l'effetto in funzione dello sfasamento (vedere Figura 20).

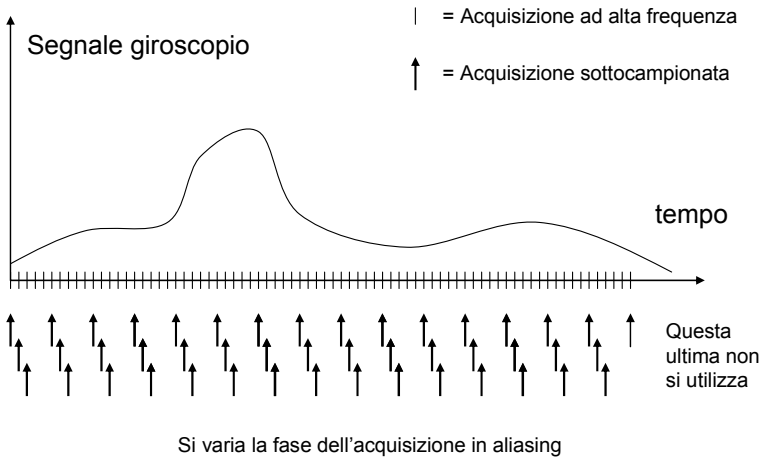


Figura 20: valutazione effetto aliasing, massimizzazione effetto in funzione dello sfasamento

Le ricostruzioni di assetto stimate prima e dopo il sottocampionamento sono state messe a confronto con la stima fornita dal sistema a triangolazione di riferimento, come schematizzato in Figura 21.

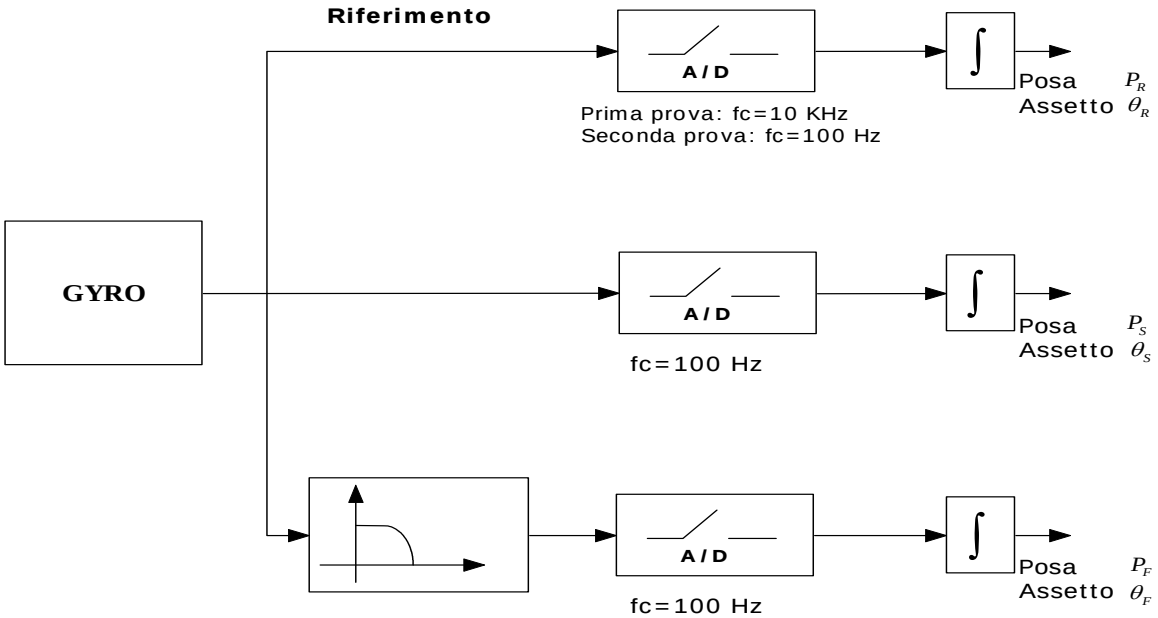


Figura 21: procedura seguita

2.5.1.3 *Ricostruzione dai dati originari*

La ricostruzione della traiettoria dalle letture degli encoder e dai dati del gyro è visualizzata in Figura 22.

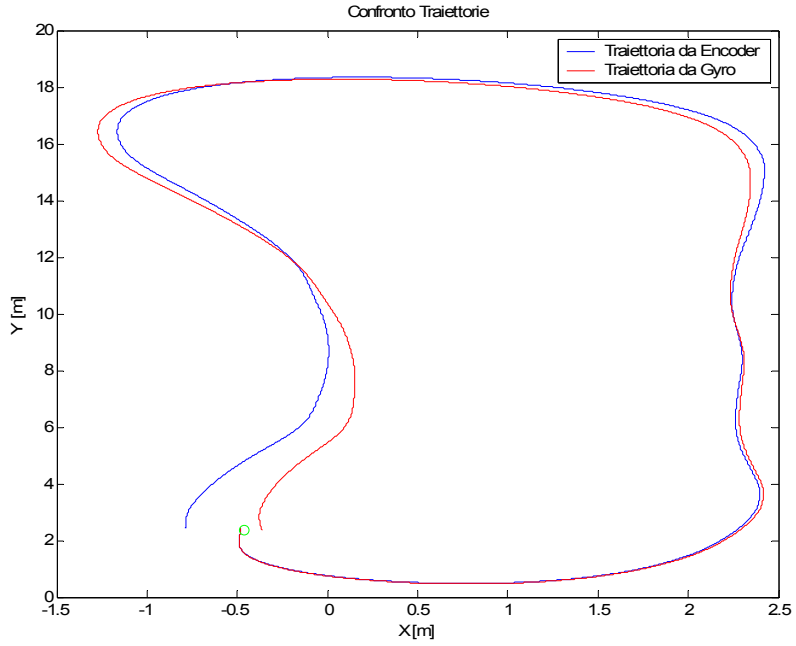


Figura 22: traiettorie stimate con encoder e con assetto fornito dal giroscopio

In figura il cerchietto verde rappresenta la posizione finale stimata dal NAV. Si può vedere che dai dati del gyro si ottiene una ricostruzione più accurata rispetto a quella ottenuta dai soli encoder.

In Figura 23 si ha il confronto relativo all'assetto:

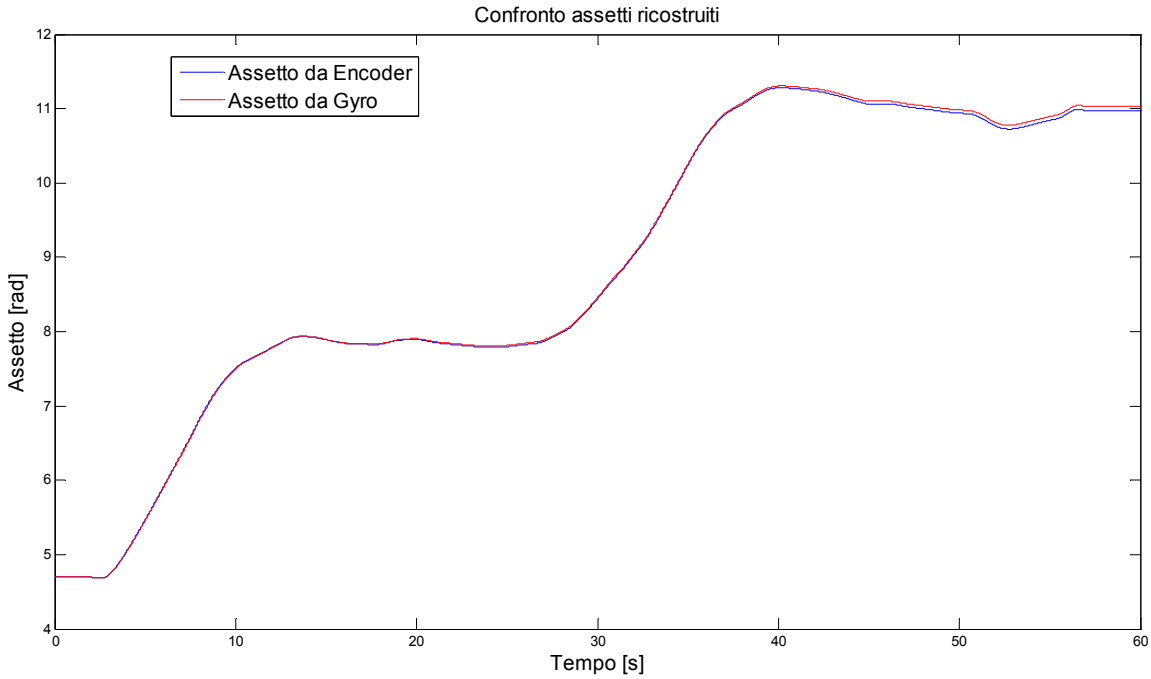


Figura 23: confronto assetto stimato con gli encoder e con il giroscopio

In Figura 24, sono evidenziate le differenze nella stima della posizione e dell'assetto tra i dati forniti dagli encoder e quelli forniti dal giroscopio.

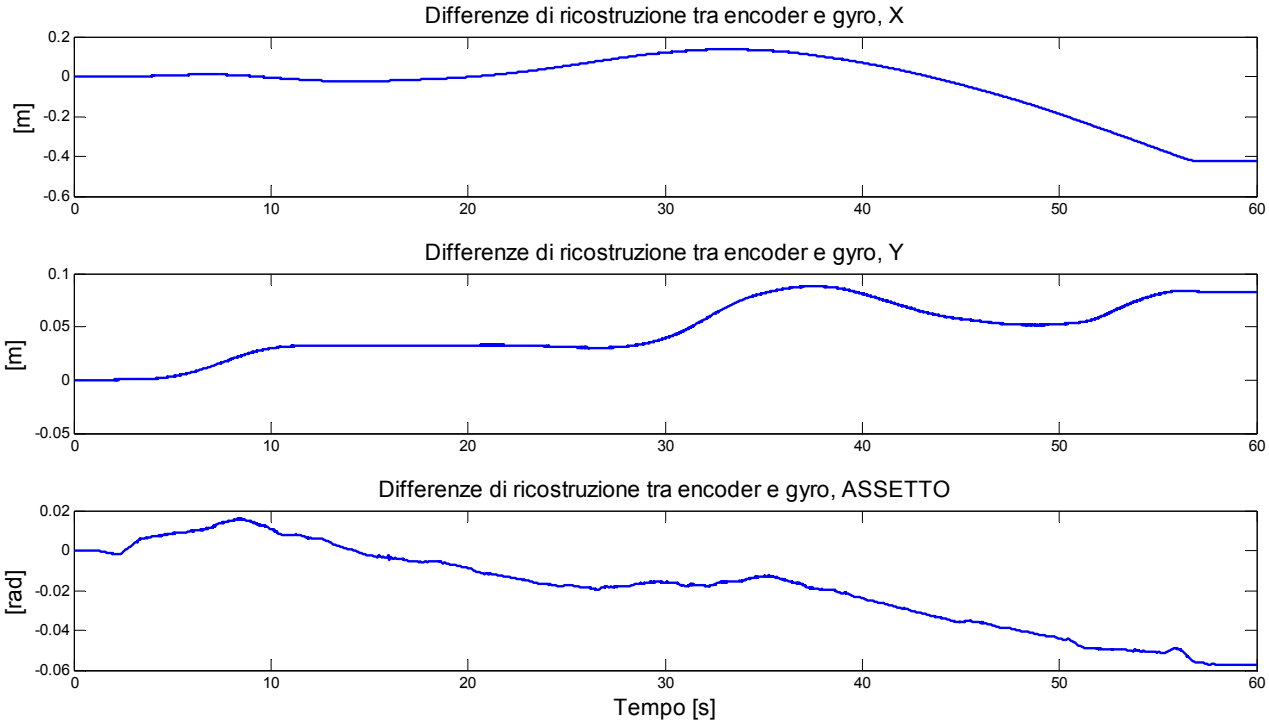


Figura 24: differenze di stima della posa tra encoder e giroscopio

2.5.1.4 Valutazione effetto dell'aliasing simulato

Dopo la sottocampionatura per simulare l'aliasing, gli assetti ricostruiti a partire dalle serie di dati ottenute risultano leggermente diversi, come si può notare in Figura 25 (si noti che due serie risultano quasi perfettamente sovrapposte in quanto non risultano più sfasate).

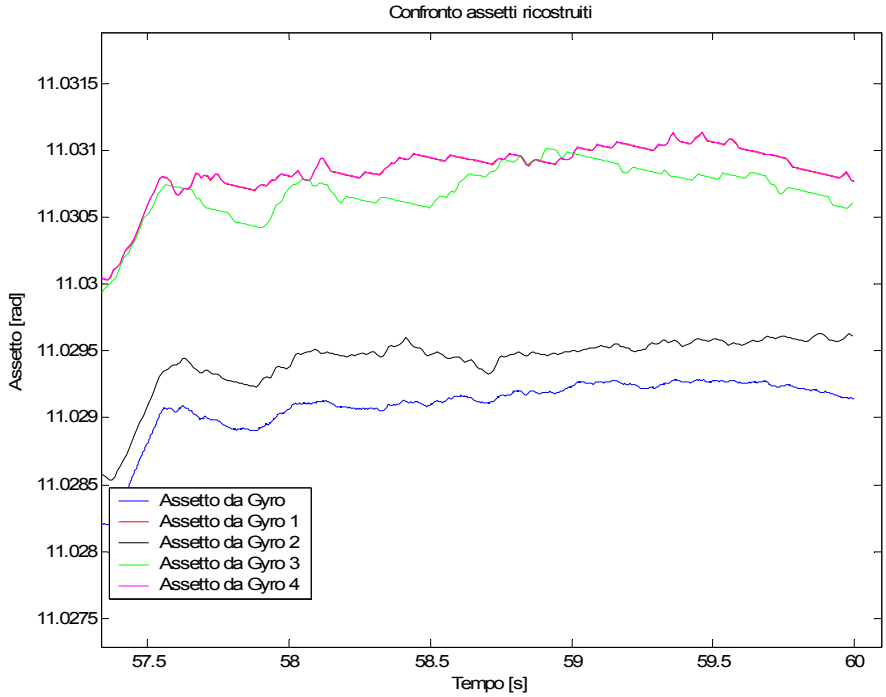


Figura 25: dettaglio della ricostruzione di assetto dopo la sottocampiontura

In Figura 26 è messo in evidenza gli scarti puntuali tra le serie di dati sottocampionati e il campionamento originale di riferimento (anche qui due serie, la prima e l'ultima, sono quasi perfettamente sovrapposte e si vede solo l'ultima).

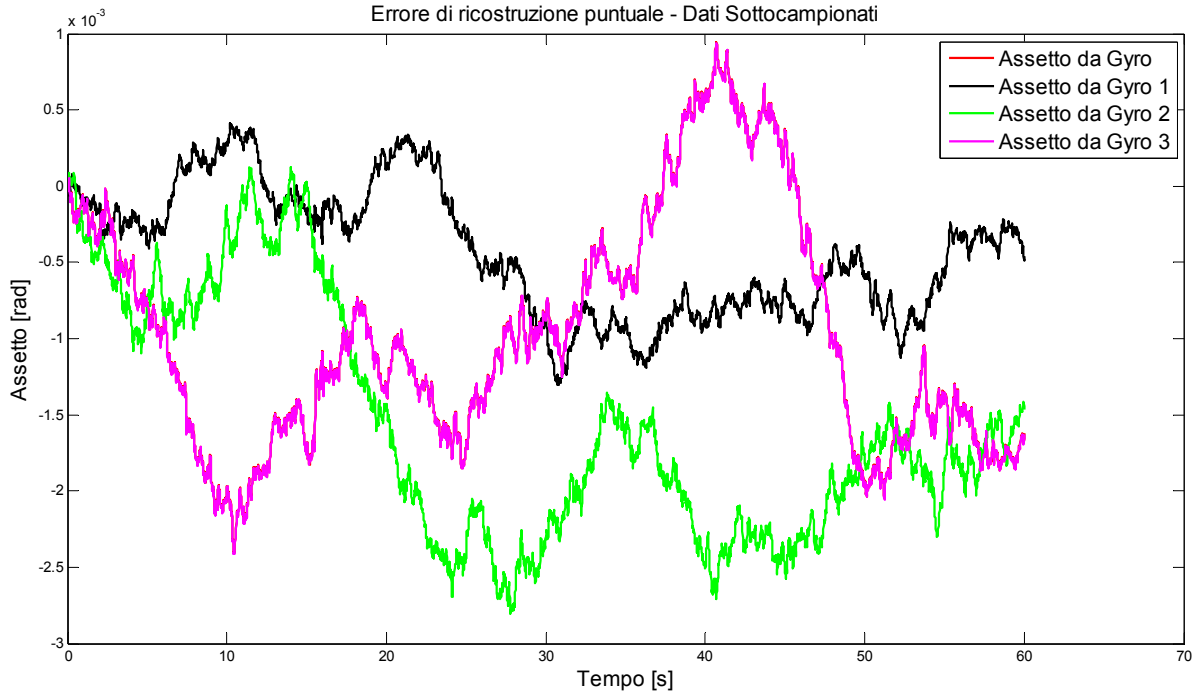


Figura 26: scarti nella ricostruzione dell'assetto tra le serie sottocampionate e il campionamento di riferimento

2.5.1.5 *Valutazione effetto filtraggio ideale (filtro senza ritardo di fase)*

Per evitare il fenomeno dell'aliasing, un sistema campionatore a frequenza Fc viene normalmente fatto precedere da un filtro passa-basso con frequenza di taglio $f_{\max}$ al più $Fc/2$, detto filtro anti-aliasing (Figura 27).

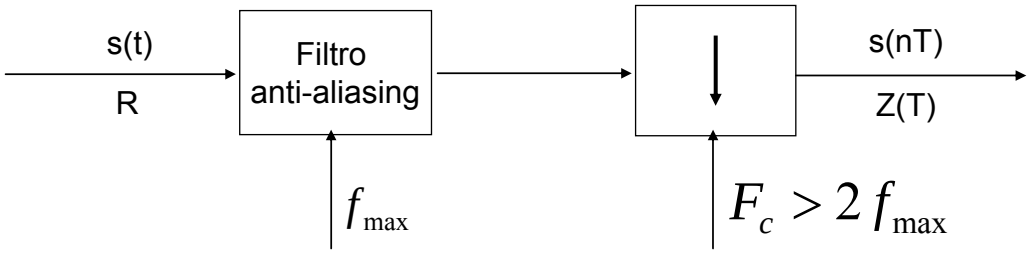


Figura 27: campionatore preceduto da un filtro anti-aliasing

Il filtro antialiasing in Figura 27 è un filtro passa-basso ed ha la funzione di porre in ingresso al campionatore un segnale a banda limitata la cui frequenza di Nyquist non superi la frequenza di campionamento. Quindi si è simulato il filtraggio con un filtro ideale per valutarne le conseguenze.

I dati campionati a 1 KHZ sono stati filtrati con un filtro Butterworth del 4° ordine e frequenza di taglio di 40 Hz, poi sono stati nuovamente sottocampionati per simulare l'aliasing. Dalle seguenti figure appare chiaro come lo scarto puntale di ricostruzione dell'assetto sia diminuito di circa 1 ordine di grandezza rispetto alle stesse serie di dati non filtrate, e di conseguenza anche la posa risulta avere degli scarti rispetto al valore di riferimento accettabili, nell'ordine di qualche millimetro (Figura 29).

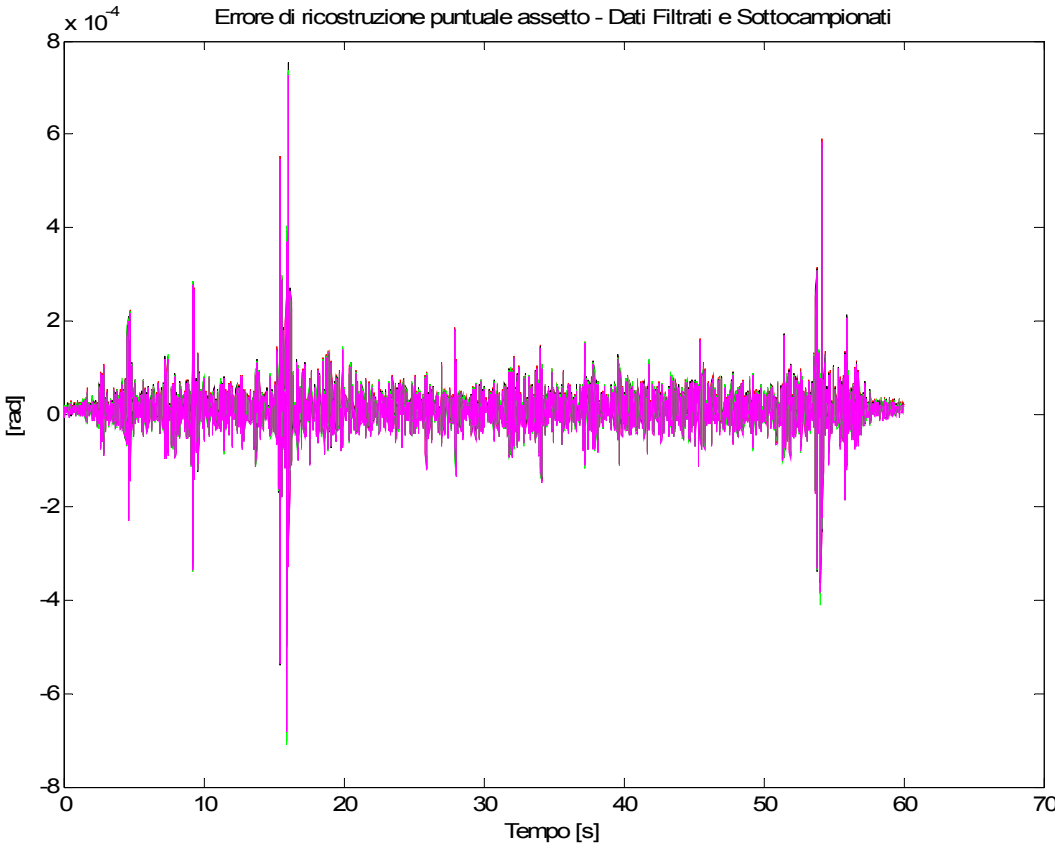


Figura 28: scarti di ricostruzione puntuale dell'assetto post-filtraggio e sottocampionatura

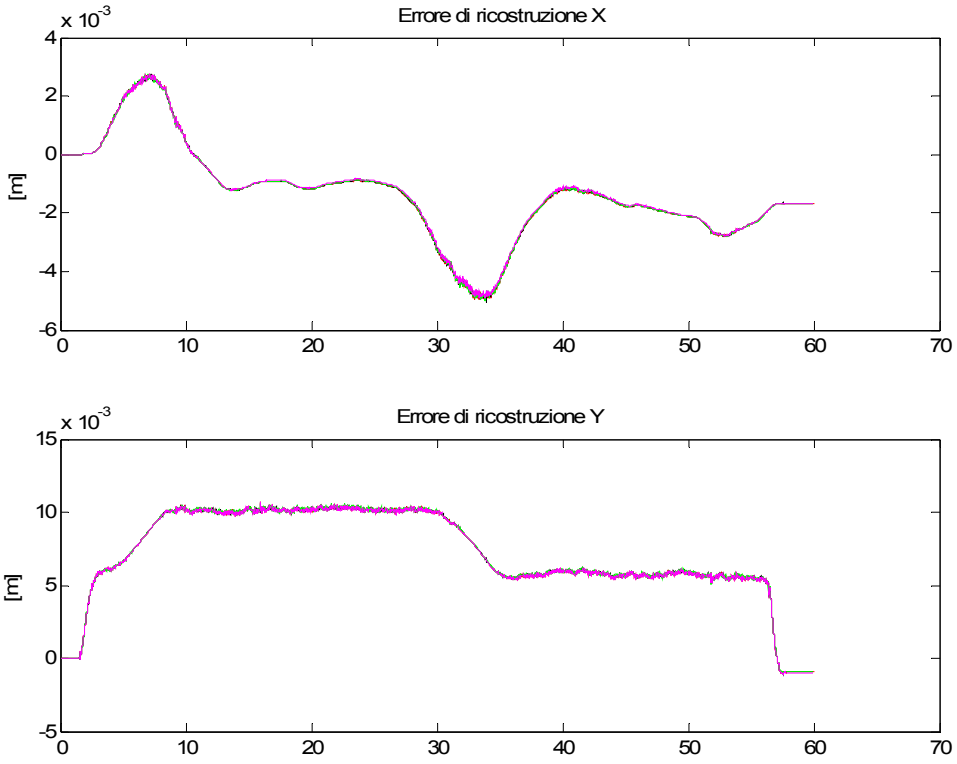


Figura 29: scarti di ricostruzione della posa post-filtraggio e sottocampionatura

In Figura 30 si nota chiaramente come rispetto alla ricostruzione della traiettoria stimata dai dati originari (linea blu più grossa) le traiettorie ricostruite post filtraggio (linee continue) risultino molto più vicine

rispetto a quelle stimate dopo la simulazione dell'aliasing (linee tratteggiate), quindi il filtraggio con un filtro ideale riduce significativamente gli effetti dell'aliasing e permette una ricostruzione più accurata della traiettoria.

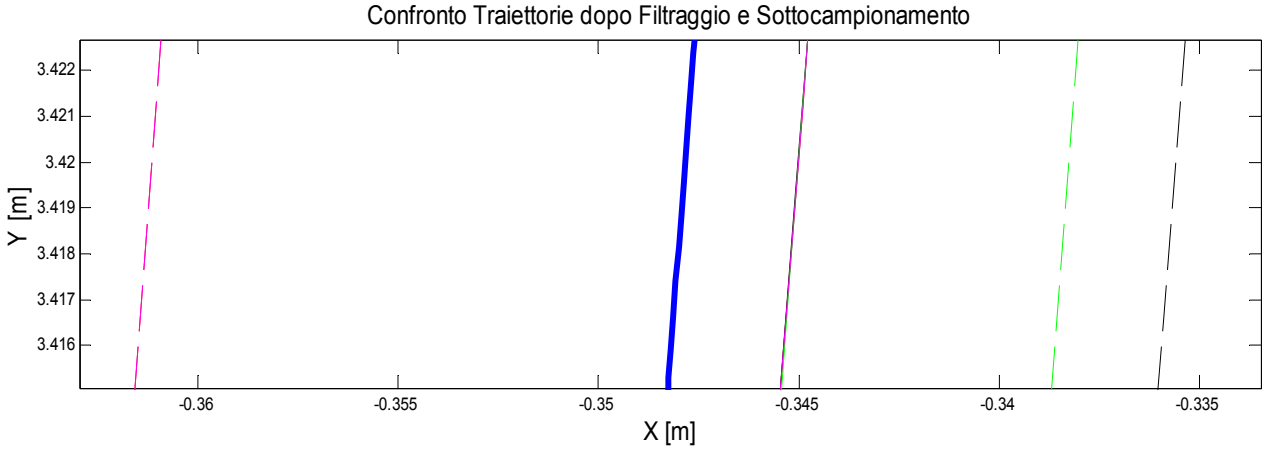


Figura 30: confronto traiettorie post filtraggio e sottocampionamento

Capitolo 3

Localizzazione mediante Lidar 2D

I Lidar sono tra i sensori più utilizzati in robotica per la conoscenza del mondo esterno. I dati forniti da tale tipologia di sensori sono tipicamente utilizzati per "osservare" l'ambiente esterno e poter evitare ostacoli, per mappare l'ambiente e localizzarsi rispetto ad esso. In questo capitolo verrà descritto un algoritmo di localizzazione riferita all'ambiente basato sul matching di scansioni provenienti dal Lidar.

3.1 Principio di funzionamento e caratteristiche del sensore LIDAR

Lo strumento principale utilizzato per lo sviluppo e il test degli algoritmi é un LIDAR (LIght Detection And Ranging), prodotto dalla SICK, modello PLS 101 (Proximity Laser Scanner) mostrato in Figura 31.

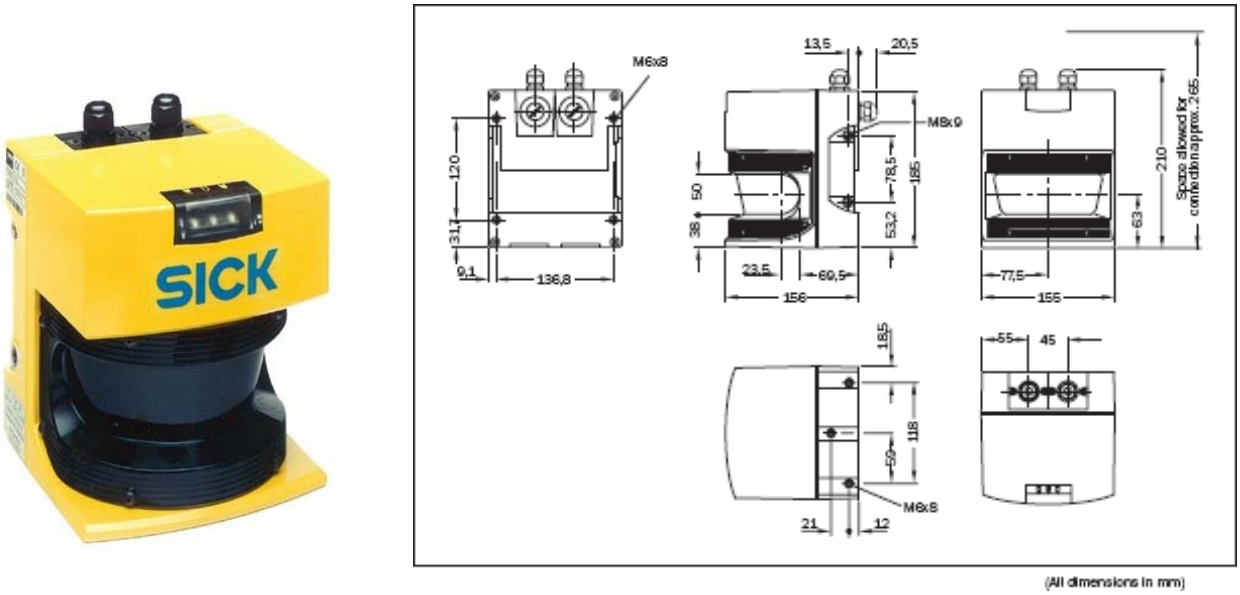


Figura 31: laser PLS 101 della SICK, utilizzato per testare gli algoritmi

É composto da un laser montato su una struttura e puntato verso uno specchio ruotante che direziona il raggio uscente, rendendo possibile una scansione a 180° dell'ambiente con una risoluzione angolare di 0.5°.

Il range di scansione arriva a 50 m e, dai datasheet, la risoluzione sulla distanza è minore di 7 cm con un'incertezza di 7 cm a 4 metri.

Il PSL 101 è in realtà un laser di sicurezza, non propriamente di misura, ed è utilizzato nell'industria per monitorare particolari zone ed eventualmente attivare allarmi o bloccare macchine. Di conseguenza per approfondirne la conoscenza come strumento di misura, è stata effettuata una taratura (Ervas, 2001) che ha dato risultati piuttosto diversi da quelli presenti nel datasheet. In particolare lo scarto quadratico medio in un range fino a 5 m si è attestato sui 3 cm, con una deviazione standard σ di 3 cm, mentre lo scarto quadratico massimo rilevato è stato di 10 cm.

Inoltre, si è osservato che nel campo 0 - 50 cm il funzionamento non è ottimale, in quanto la frequenza di clock del processore non è sufficiente a fronteggiare la velocità con cui il raggio torna indietro da così brevi distanze svolgendo tutti i calcoli richiesti. Come si può vedere in Figura 32, questo fatto influisce negativamente sulla scansione di oggetti particolarmente vicini (poche decine di cm): in particolare, nell'immagine di sinistra sono evidenziate con le lettere A e B, le zone da cui sono state riprese le due scansioni (quella raffigurata a sinistra e quella a destra) e con un cerchio le zone che mostrano notevoli differenze, nella seconda scansione infatti non si distinguono chiaramente i contorni dell'oggetto (un europallet)

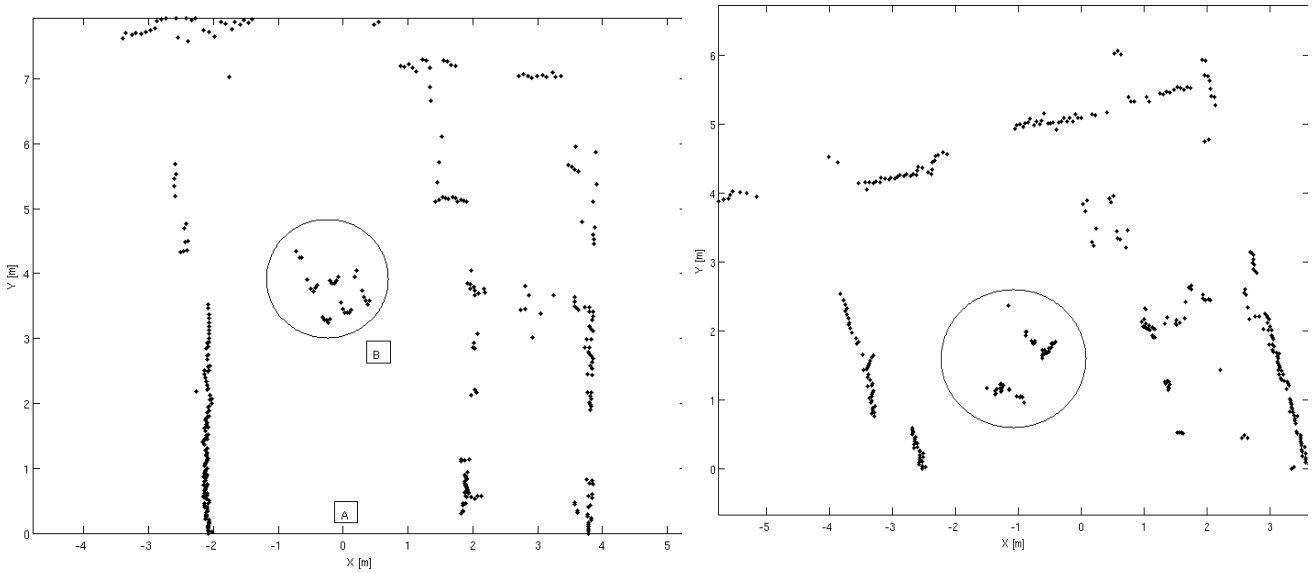


Figura 32: problema di misura del Lidar con oggetti troppo vicini: si noti nell'immagine di destra come i contorni dell'oggetto siano confusi

Per poter eseguire le prove e raccogliere i dati è stato osservato un tempo di warm-up di circa 3 ore, per poter far lavorare il sensore a temperatura di esercizio (Ye and Johann Borenstein, 2002).

In futuro si prevede di testare gli algoritmi su un nuovo laser di misura della Sick, che permette un angolo di visuale di 360°, una risoluzione angolare fino a 0.125° e un'accuratezza migliore sulla distanza.

3.1.1 Simulazione del sensore

È stato sviluppato un algoritmo che simula il sensore per poter testare gli algoritmi senza dover ogni volta utilizzare il robot, e in questo modo si possono ricreare gli scenari più disparati.

La funzione richiede in input 3 variabili:

- Dati: matrice Nx6 dell'ambiente da scansionare, dove N rappresenta il numero di linee di cui é composto l'ambiente; le colonne 1 e 2 rappresentano i coefficienti Theta [rad] e Rho [m] delle linee, secondo l'espressione $\cos(\theta)X + \sin(\theta)Y = \rho$; le colonne 3 e 4, 5 e 6 rappresentano i punti iniziale e finale in coordinate cartesiane [m, m] di ogni segmento;
- err_theta: incertezza angolare del laser [rad];
- err_dist: incertezza sulla distanza del laser [m].

L'ambiente deve essere descritto da segmenti, rappresentati dai punti di inizio e fine e dai coefficienti della linea congiungente. L'output fornito è una matrice contenente le coordinate cartesiane dei punti scansionati. È possibile sogliare i dati a una distanza massima di scansione oltre la quale il valore trovato viene posto uguale al valore di soglia.

L'algoritmo calcola per ogni segmento, per ogni raggio, l'intersezione (sempre esistente a meno di parallelismi), e se questa cade tra gli estremi del segmento (la distanza tra il punto trovato e i 2 estremi deve essere minore della lunghezza del segmento), allora un nuovo punto é stato trovato.

L'intersezione si calcola risolvendo rispetto ad X e Y il seguente sistema:

$$\begin{cases} \cos(\theta_a) * X + \sin(\theta_a) * Y = \rho_a \\ \cos(\theta_r) * X + \sin(\theta_r) * Y = \rho_r \end{cases}$$

dove il pedice *r* indica i coefficienti relativi ai raggi di scansione del laser, mentre il pedice *a* é riferito ai segmenti dell'ambiente. Il punto trovato viene confrontato con i punti precedentemente trovati (stesso raggio, altre linee) e viene memorizzato solo quello piú vicino.

Vengono prese in considerazione due fonti di incertezza: l'incertezza sulla risoluzione angolare del Lidar, e l'incertezza sul range dei punti trovati. Entrambe vengono modellate con una distribuzione gaussiana.

Rispetto alle scansioni ottenute con un laser reale, nel modello non vengono considerati i parametri fisici della superficie colpita, come l'indice di diffusione o di riflessione, e nemmeno l'angolo di incidenza del

raggio. Come mostrato in (Ye, Borenstein, 2002), questo è una delle maggiori fonti di incertezza delle scansioni: si é preferito utilizzare un modello semplificato per non appesantire il codice, preferendo snellezza e velocitá ad una modellizzazione più approfondita delle fonti di incertezza, considerando comunque incertezze maggiori per i parametri presi in considerazione, in modo da includere nei valori finali anche le fonti di incertezza non modellate.

3.1.2 Media delle scansioni

Puó risultare utile dover mediare piú scansioni, sia acquisite che simulate per diminuire l'incertezza sulle misure (confrontare le immagini in Figura 33).

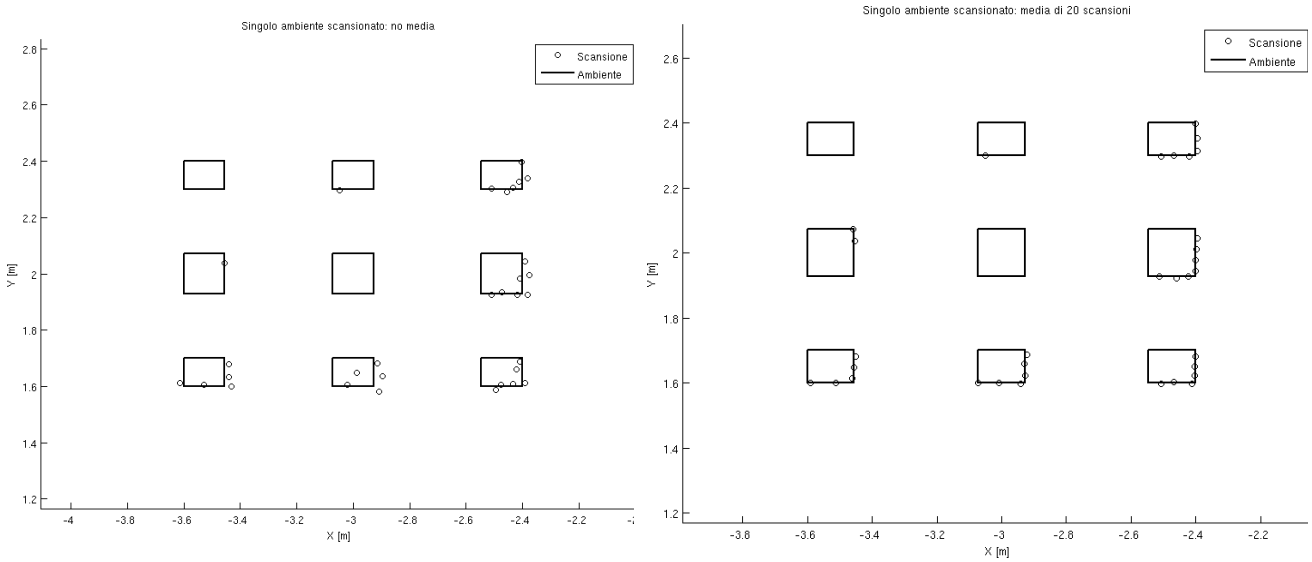


Figura 33: confronto scansione singola e media di 20 scansioni

Si puó dimostrare infatti (vedi (Adams, 2002)) che, definito σ_r lo scarto quadratico medio di *n* misure, la loro media avrà uno scarto quadratico medio pari a:

$$\bar{\sigma}_r = \frac{\sigma_r}{\sqrt{n}}$$

Per fare questo, a partire da N scansioni, é stato scritto un algoritmo che simula delle scansioni di un modello e corregge eventuali errori che si avrebbero facendo la media semplice.

Nell'eseguire l'operazione di media, sia di scansioni reali che di quelle simulate, bisogna prestare particolare attenzione a quei raggi di scansione che a causa dell'incertezza sullo step angolare colpiscano spigoli di oggetti diversi a distanze diverse. Mediando semplicemente si otterrebbero punti dove nella realtà non vi è presenza di oggetti, di conseguenza le medie vengono effettuate dopo un'operazione di clusterizzazione dei punti che valuta la distanza tra punti catturati dallo stesso raggio in scansioni differenti.

Si può notare il risultato tra i differenti modi di mediare i punti nelle immagini di Figura 34, dove nella prima é stata eseguito un processo di media classico, e si possono notare dei punti ben distanti dalle linee che rappresentano le superfici dell'ambiente: i raggi a volte colpiscono il pallet, a volte il muro retrostante. Nell'immagine di destra si nota come il problema venga risolto con la clusterizzazione descritta.

Nella realtà vi sono altri principi fisici fonti di incertezza legati alle caratteristiche del Lidar: a causa della divergenza del fascio laser che aumenta all'aumentare della distanza tra il rivelatore e l'oggetto scansionato un raggio può finire in parte sullo spigolo, in parte su quello che c'é dietro,. Di conseguenza la potenza che arriva al rivelatore risulta una media tra quelle ritornate dalle riflessioni in diversi punti: il risultato è un valore di range intermedio tra i due. Inoltre, quando il software interno del laser restituisce una acquisizione, questa in realtà é la media di 2 scansioni successive, distanti temporalmente qualche centesimo di secondo, e quindi con ottima approssimazione effettuate dalla stessa posizione ed assetto. Questo processo di media può tuttavia subire lo stesso fenomeno prima spiegato, da cui i risultati.

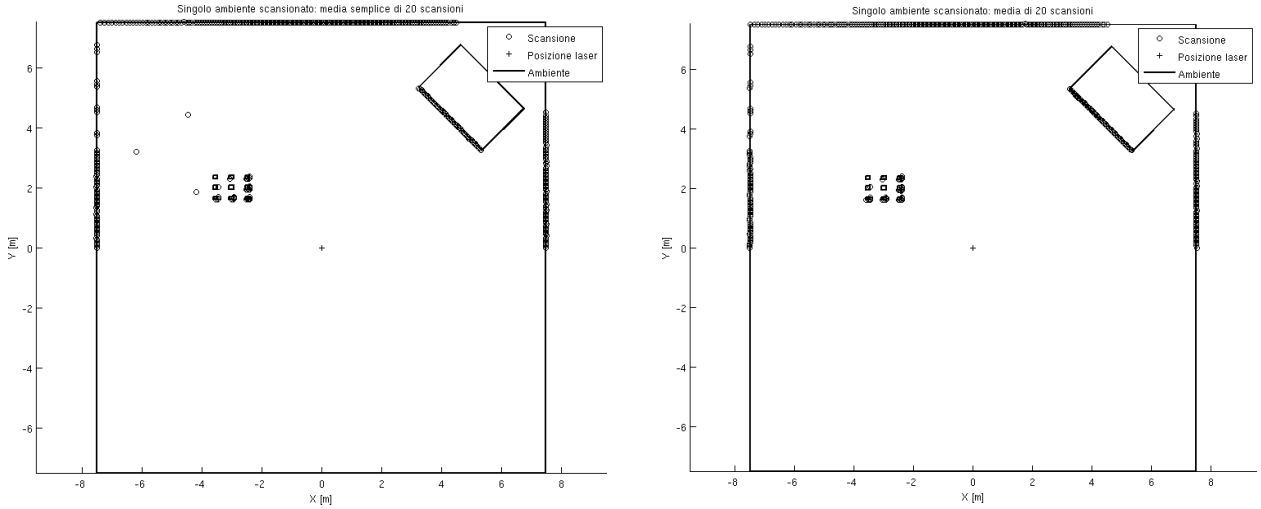


Figura 34: Media tradizionale dei raggi di scansione (sx) e media con clusterizzazione (dx)

3.2 Stima di posizione tramite algoritmo ICP

È stato sviluppato un algoritmo di localizzazione riferito all'ambiente a partire dai dati del sensore LIDAR. Viene calcolata la rototraslazione tra due scansioni dell'ambiente cercando il miglior matching tra le due, ed in questo modo, conoscendo la posa di acquisizione della prima scansione in ordine temporale, è possibile determinare la posizione in cui è stato acquisita la seconda. È così possibile ricostruire la traiettoria sia in modo incrementale calcolando ad ogni ciclo il segmento di traiettoria da aggiungere al precedente, sia in modo assoluto riferito all'ambiente.

L'algoritmo è stato testato in post-processing e attualmente si sta lavorando sulla traduzione in C per l'implementazione real-time.

3.2.1 Descrizione algoritmo

L'Iterative Closest Point (ICP) (Besl, McKay 1992) è un algoritmo molto utilizzato per l'allineamento di insiemi di punti nello spazio 3D.

In pseudocodice, l'algoritmo è composto dai seguenti passi:

Algoritmo ICP (Iterative Closest Point) → Dato un modello X e un oggetto P:

- A partire da una trasformazione iniziale
- Procedura iterativa per convergere a un minimo locale:
  1. $\forall p \in P$ trova il punto più vicino (the **closest point)** $x \in X$
  2. calcola la trasformazione $P_{k+1} \leftarrow Q(P_K)$ che **minimizzi le distanze** tra ogni p e x
  3. termina la procedura quando il cambiamento negli scarti si mantiene sotto una determinate soglia.
- Scegli la miglior soluzione possible a partire da differenti pose iniziali.

In questo modello e oggetto sono semplicemente due scansioni dell'ambiente effettuate dal LIDAR in pose diverse.

L'algoritmo si basa su una funzione costo da minimizzare. Data una serie di punti A appartenenti alla prima scansione, si vuole sovrapporre i punti di B, seconda scansione, su questa, come in Figura 35 .

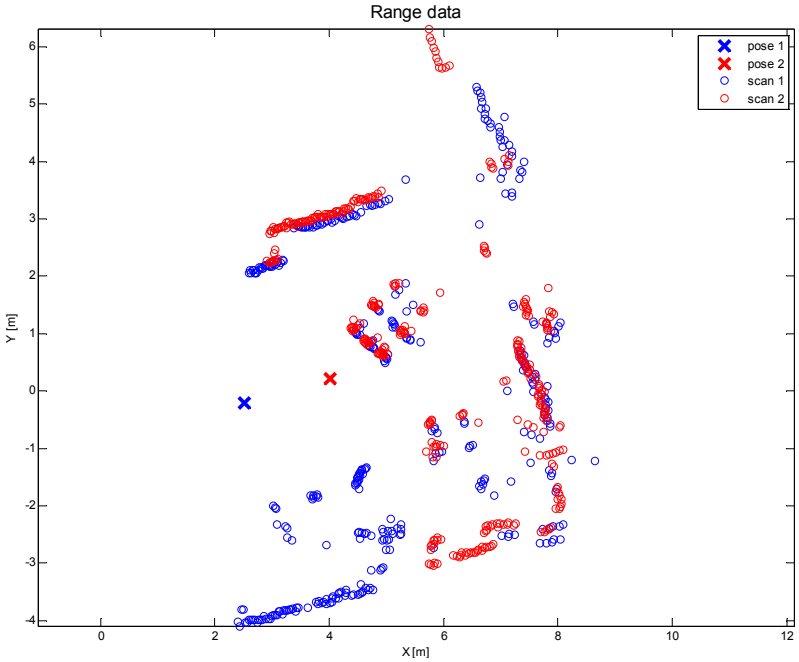


Figura 35: scansioni dell'ambiente da due differenti pose

Si definisce una trasformazione nel piano come:

$$O(a,P_B)=\begin{bmatrix}\cos t & -\sin t\\ \sin t & \cos t\end{bmatrix}\begin{Bmatrix}x_b\\ y_b\end{Bmatrix}+\begin{Bmatrix}t_x\\ t_y\end{Bmatrix}$$

in cui P_B sono le coordinate cartesiane dei punti di B, mentre $a=\{\theta,t_x,t_y\}$ é il vettore dei coefficienti della roto-traslazione da eseguire.

Si definisce errore di allineamento la funzione $\varepsilon^2(|x|)$, ed il più classico esempio é:

$$\varepsilon^2(|x|)=\|x\|^2$$

cioè la somma dei quadrati delle distanze tra coppie di punti; queste vengono specificate tramite la funzione $\phi(i)$, che associa ad ogni punto *k* della scansione B il corrispettivo *i* nella scansione A in modo che la distanza *d(PB(k), PA(i))* sia la minima (fissato *k*) rispetto agli altri possibili punti *i*, come si può vedere in Figura 36.

Si nota come ogni punto di B abbia il suo corrispettivo in A e non il viceversa: si vedrà nel seguito come questo aspetto possa essere modificato ed in quali casi convenga farlo.

Ottenute tutte le coppie, si definisce il vettore $E_i(a,P_A,P_B,\phi)$ (d'ora in poi solo *Ei(a)*) come:

$$E_i(a)=\omega_i\varepsilon(P_{A,i}-O(a,P_{B,i}))$$

cioè ogni componente rappresenta la distanza della coppia di punti di indice i. Il termine ω_i é un peso, ed il suo utilizzo verrà chiarito nel seguito.

Si tratta quindi di minimizzare la funzione:

$$\varepsilon^2(|x|)=\sum_i(E_i(a))^2$$

ottimizzando il parametro *a*.

Con questo fine, sono state intraprese due diverse strade:

- minimizzazione numerica con algoritmo Pattern Search
- minimizzazione analitica (Lu, Milios, 1994).

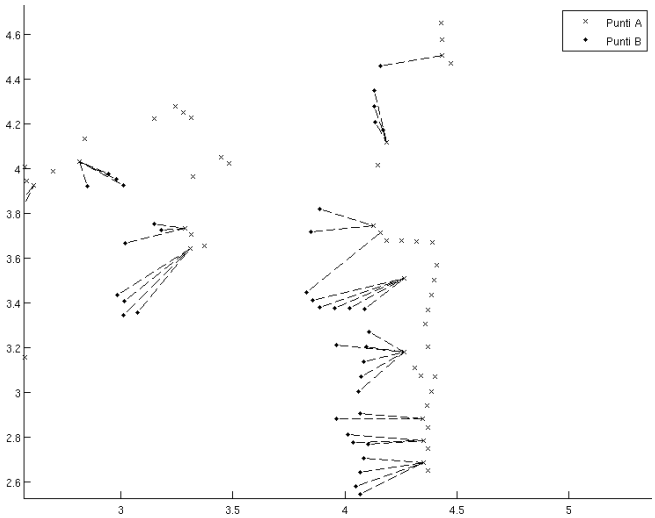


Figura 36: ICP: creazione coppie tra i punti A e i punti B

3.2.1.1 Minimizzazione numerica: algoritmo Pattern Search

L'algoritmo utilizzato per la ricerca numerica di minimo si chiama Pattern Search, e fa parte del gruppo di algoritmi detti di ricerca diretta. In Matlab é inserito nel toolbox Genetic Algorithm and Direct Search toolbox, e per richiamarlo si usa il comando *patternsearch*. Rispetto ai metodi classici (Steepest Descent, Gradiente Coniugato...) non necessita di conoscere lo Jacobiano (numerico o analitico) della funzione.

L'algoritmo trova una sequenza di punti che si avvicinano al punto di minimo. Il valore della funzione puó decrescere o rimanere pari al suo valore iniziale, mai aumentare. Il flowchart dell'algoritmo è rappresentato in Figura 39.

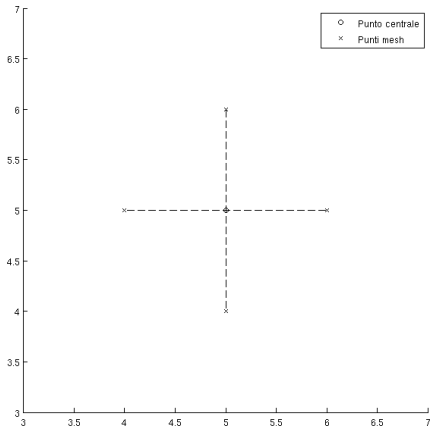


Figura 37: mesh creata intorno al pto centrale, caso con 2 variabili

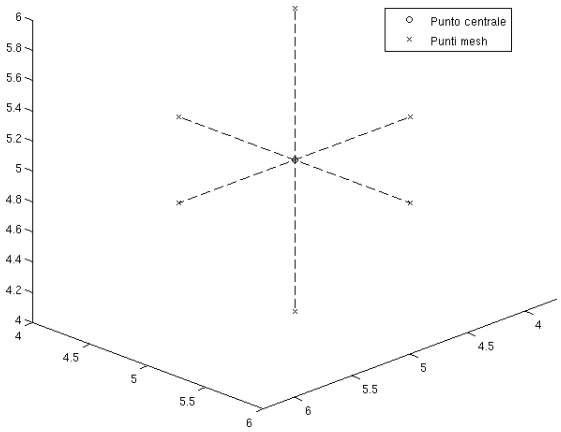


Figura 38: mesh creata intorno al pto centrale, caso con 3 variabili

Verrà fatto il caso di una ottimizzazione di 2 variabili per spiegare l'algoritmo. Fornito un punto iniziale, viene creata una mesh intorno a questo, cioè si individuano 2*N punti, dove N è il numero di variabili da ottimizzare, in cui viene valutata la funzione, come mostrato in Figura 37 per 2 variabili e in Figura 38 per 3 variabili. In questo caso, la distanza tra il punto centrale ed i suoi estremi vale 1; per ogni punto della mesh vengono calcolati i valori della funzione, e viene individuato il minore (compreso il valore del punto centrale).

A questo punto possono verificarsi due casi: la funzione nel punto centrale ha valore minore rispetto ai punti della mesh (ricerca senza successo), oppure uno dei punti della mesh ha minimizzato la funzione (ricerca con successo). Nel primo caso, la grandezza della mesh viene moltiplicata per un fattore 0.5 (quindi dimezzata), senza cambiare il punto centrale e si ripete la ricerca sulla nuova mesh. Nel secondo caso, si ripete la ricerca prendendo come nuovo punto centrale quello che minimizza la funzione tra quelli valutati, e la dimensione della mesh viene moltiplicata per un fattore 2 (quindi raddoppiata).

La ricerca si ferma quando viene verificato uno dei seguenti criteri di stop:

- la dimensione della mesh é minore di una tolleranza prefissata;
- è stato raggiunto il numero massimo di iterazioni, di valutazione di funzioni o il tempo massimo di esecuzione dell'algoritmo;
- La distanza tra due punti in due iterazioni successive, o il loro valore all'interno della funzione da minimizzare é inferiore ad una tolleranza prefissata.

I parametri che indicano i moltiplicatori della mesh ad ogni iterazione (sia con successo che senza) possono essere modificati, come pure la dimensione iniziale. Nel nostro caso, sono stati utilizzati i valori di default.

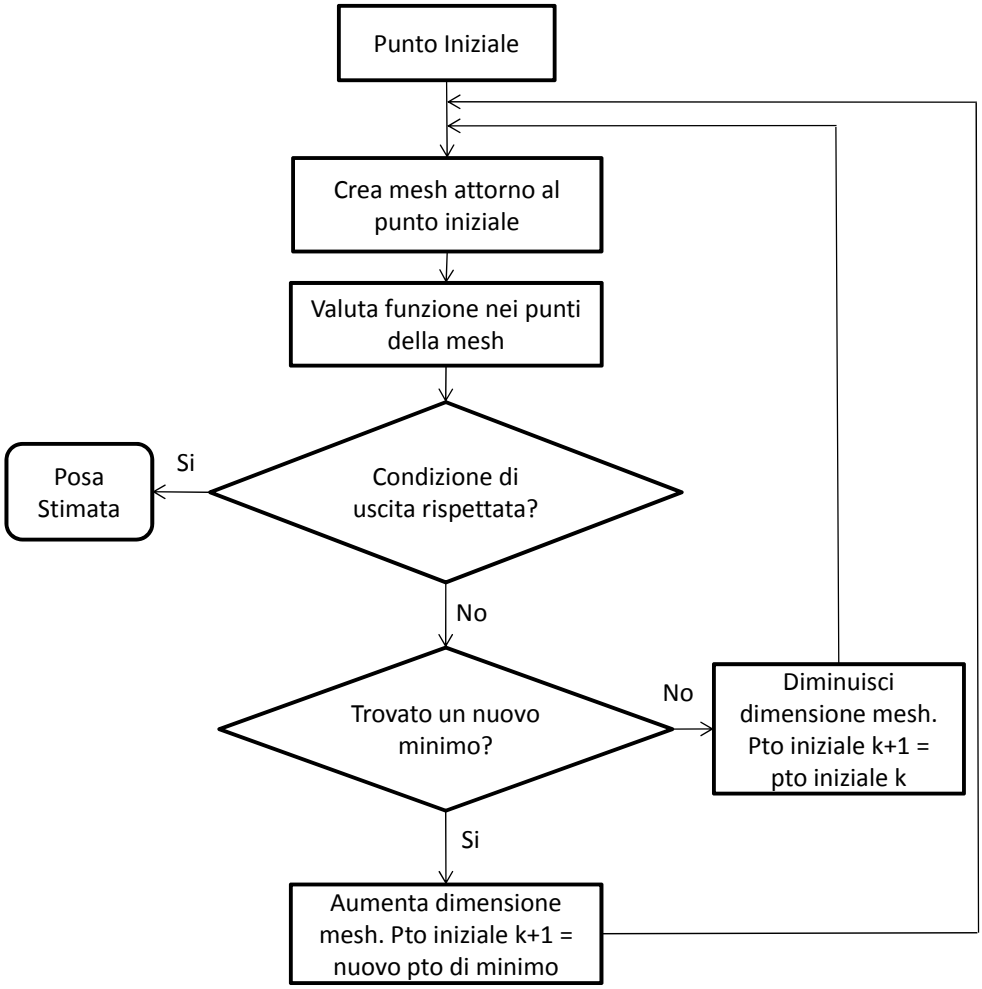


Figura 39: patternsearch flowchart

Per velocizzare la convergenza dell'algoritmo è possibile impostare dei limiti entro cui effettuare la ricerca. Ad esempio se una delle variabili è un angolo, si possono utilizzare come limiti 0 e 2π.

È importante notare come la creazione della mesh per le diverse variabili non tenga conto delle unità di misura, quindi delle dimensioni in gioco. Ad esempio nel caso in trattazione, le tre variabili sono un angolo (assetto) e due lunghezze (posizione in coordinate cartesiane dell'oggetto). Variare di 1 un angolo può voler dire molto o poco, a seconda che si utilizzino gradi o radianti. Analogamente per la lunghezza, se si utilizzano metri o millimetri. Ad esempio, nel caso di gradi e metri, l'algoritmo fará ampie variazioni sulla lunghezza, ma spazierà poco sull'angolo, a meno di un elevato numero di iterazioni. Nel caso in esame le unità di misura scelte sono radianti e metri.

Una soluzione a questo problema consiste nel adimensionalizzare le variabili, ad esempio ponendo:

$$\theta_{adi} = \frac{\theta}{\theta_{\max}}, \qquad X_{adi} = \frac{X}{X_{\max}}, \quad Y_{adi} = \frac{Y}{Y_{\max}}$$

dove:

- θ_{adi} = angolo d'assetto adimensionalizzato
- θ = variabile assetto
- $\theta_{\max}$ = massimo angolo d'assetto possibile
- X_{adi} , Y_{adi} = posizione adimensionalizzata
- X , Y = variabile posizione
- $X_{\max}$, $Y_{\max}$ = massima posizione possibile

Il massimo angolo d'assetto possibile é 2π, mentre per definire la massima posizione possibile si può ricorrere alle dimensioni (note) dell'ambiente in cui l'oggetto si trova o la massima distanza di scansione del laser. Ad esempio, se l'oggetto da cercare si trova in una stanza dalle dimensioni di 10 x 5 m, si possono impostare queste sia come limiti della ricerca, sia come costanti di adimensionalità.

3.2.1.2 Minimizzazione analitica

La soluzione analitica della minimizzazione fornisce:

$$E(a) = \sum_i (E_i(a))^2 = \sum_i (R_\theta P_{B,i} + T - P_{A,i})^2 =$$
$$= \sum_i (\cos\theta x_{B,i} - \sin\theta x_{B,i} + t_x - x_{A,i})^2 + (\sin\theta y_{B,i} - \cos\theta y_{B,i} + t_y - y_{A,i})^2$$

In questo caso, per semplificare le formule, tutti i pesi sono stati considerati unitari.

Definendo con l'apice ' le coordinate della scansione A, mentre senza si fa riferimento alla scansione B, si ricava in forma chiusa:

$$\theta = \arctan\left(\frac{S_{xy'} - S_{yx'}}{S_{xx'} + S_{yy'}}\right)$$

$$t_x = \overline{x}' - (\overline{x}\cos\theta - \overline{y}\sin\theta)$$

$$t_y = \overline{y}' - (\overline{x}\sin\theta + \overline{y}\cos\theta)$$

intendendo con $\overline{x}$, $\overline{y}$, $\overline{x}'$ e $\overline{y}'$ le medie delle rispettive coordinate, e definendo:

$$S_{xx'} = \sum_i (x_i - \overline{x})(x_i' - \overline{x}')$$

$$S_{yy'} = \sum_i (y_i - \overline{y})(y_i' - \overline{y}')$$

$$S_{yx'} = \sum_i (y_i - \overline{y})(x_i' - \overline{x}')$$

$$S_{xy'} = \sum_i (x_i - \overline{x})(y_i' - \overline{y}')$$

3.2.1.3 Confronto metodi di minimizzazione

Il metodo analitico e quello numerico portano agli stessi risultati nel caso in cui il punto iniziale fornito all'algoritmo sia prossimo alla soluzione ottima (qualche grado nella rotazione e qualche decina di centimetri nella traslazione). In questo caso il metodo analitico è più veloce ci circa un fattore 10 rispetto a quello numerico. Si notano differenze nel caso di grandi roto-traslazioni (decine di gradi e qualche metro), in cui il metodo di minimizzazione numerico si dimostra più robusto impiegando oltretutto un numero di iterazioni inferiore rispetto a quello analitico.

3.2.2 Robustezza dell'algoritmo

La definizione di "robusto" sta ad indicare la tendenza o meno di un algoritmo ad ottenere buoni risultati anche in presenza di disturbi.

Nel seguito si affronteranno i principali problemi incontrati, e verranno proposte delle soluzioni.

3.2.2.1 Sezione di coda

Il problema principale nel correlare due scansioni consiste nelle sezioni di coda, sempre presenti nel caso in cui il laser non abbia un campo di vista di 360°. Questo problema é evidenziato soprattutto in caso di grandi rotazioni o traslazioni, ma si verifica comunque sempre in presenza di movimento del robot.

Si tratta di parti di scansione che non sono in comune tra due acquisizioni per un duplice motivo: la presenza di un oggetto potrebbe nascondere parti dell'ambiente in una delle due pose, oppure il cambio di posa non permette ai raggi di scansione di raggiungere quelle zone perché esterne al FOV (campo di vista - Field Of View).

Viene ora affrontato il problema legato all'utilizzo di un sensore LIDAR con campo di vista inferiore a 360°.

In Figura 40 sono mostrate due scansioni dello stesso ambiente acquisite da due pose differenti con un FOV di 180°, e si può notare come nella seconda, in rosso, manchino molte parti rispetto a quella in verde. Queste differenze possono creare problemi al calcolo dell'ICP e portare l'algoritmo a convergere verso soluzioni locali non corrette. Per risolvere il problema, nota la rototraslazione tra le due scansioni a partire dalla stima odometrica (si tratta di una stima affetta da incertezza ma sufficientemente accurata per questa problematica), vengono scartate le parti delle scansioni che non si trovino nel campo di vista comune tra le due pose. Il risultato, è mostrato in Figura 40 come pallini neri: in questo caso è stato sufficiente eliminare elementi dalla scansione 1.

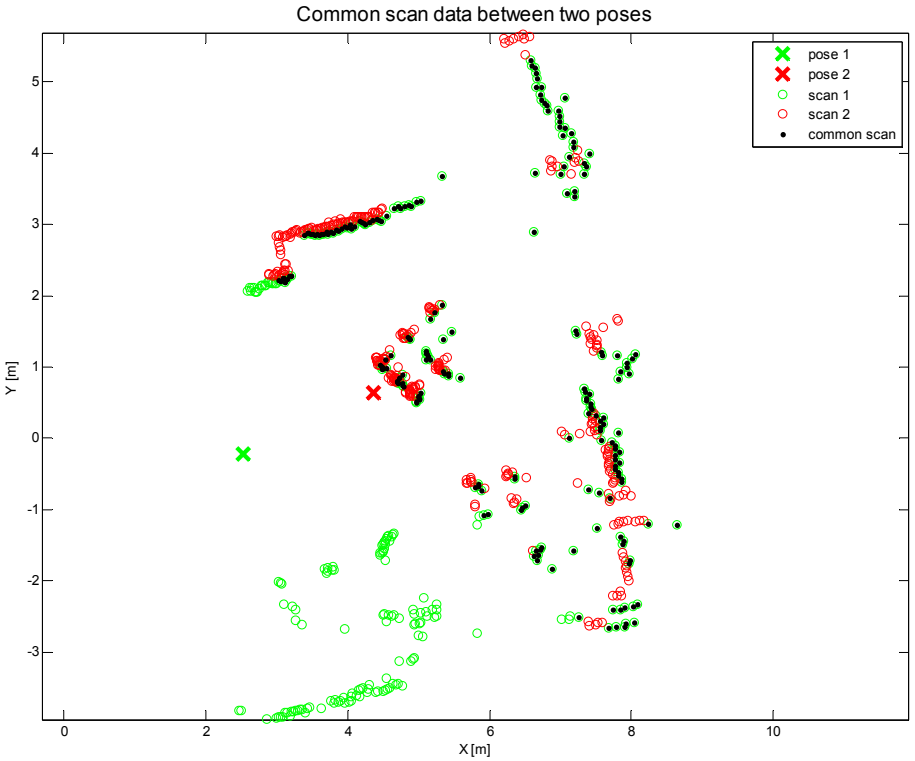


Figura 40: problema delle sezioni di coda tra due scansioni prese da pose differenti

3.2.2.2 Accoppiamenti fra scansioni

Come visto, gli accoppiamenti tra una scansione e l'altra vengono calcolati in modo che ad ogni punto della scansione B sia correlato uno della A, ma non il viceversa, come evidenzia la Figura 41. Questo tipo di associazione puó portare anche a risultati scorretti, come nel seguente esempio: la scansione B é un

rettangolo, mentre la A è una parte di un suo lato minore, ovviamente la soluzione non é unica in quanto la linea puó appartenere ad uno qualsiasi dei lati.

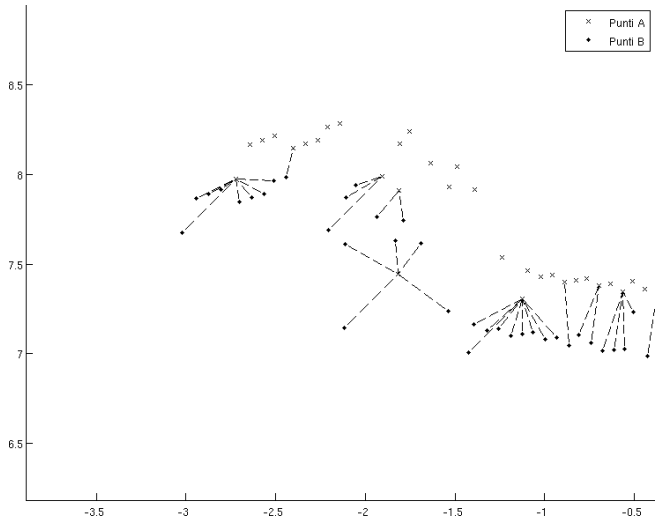


Figura 41: Coppie create dall'algoritmo ICP tra le scansioni A e B

Si supponga di trasformare le linee in molle: in effetti, la funzione minimizza la somma delle distanze, e le forze elastiche sono proporzionali a queste: si sta quindi minimizzando l'energia potenziale elastica di molle appese tra i punti. In questo caso, risulta ovvio attendersi che la linea stia in mezzo al rettangolo, orientata secondo il lato maggiore di questo: indubbiamente questa non é la soluzione cercata, ma é la soluzione matematicamente corretta per quel tipo di associazione (si veda la Figura 42 dx). Ecco dunque che risulta necessario cambiare itinerario, ed inserire una condizione prima di far partire l'algoritmo: la scansione A é quella (tra le due) con il maggior numero di punti (come in Figura 42 sx), in caso di uguaglianza, ovviamente, é indifferente.

Ovviamente si dovrà tener conto di questa scambio nel verso delle trasformazioni, in quanto normalmente è considerata scansione B quella posteriore temporalmente.

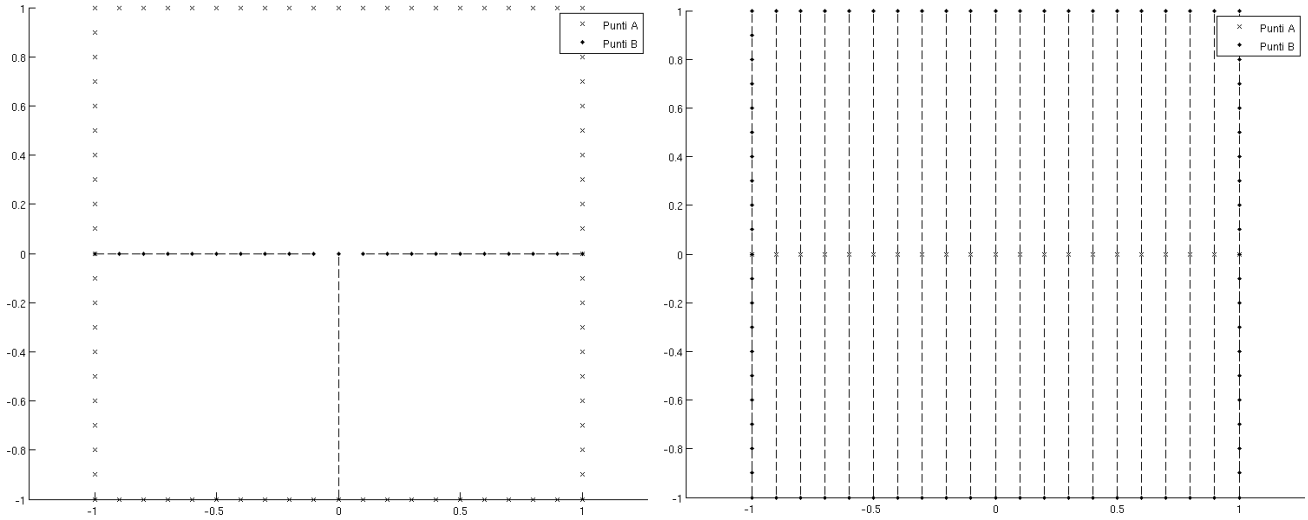


Figura 42: Coppie di punti tra rettangolo e linea: associazione corretta (SX), e associazione non corretta (DX)

3.2.2.3 Kernel

In 3.2.1 è stato definito l'errore di allineamento come:

$$\varepsilon^2(|x|) = \|x\|^2$$

il cui significato è la somma dei quadrati delle distanze tra le coppie di punti. La funzione $\varepsilon(|x|)$ viene definita kernel, ed esistono in letteratura (Fitzgibbon, 2001) esempi di come modificando questa funzione si amplifichi il bacino di convergenza, inteso come una maggiore differenza di posa tra le due scansioni da confrontare (anche decide di gradi sull'assetto). I due kernel (oltre a quello qui utilizzato) maggiormente diffusi sono:

- Huber kernel: $\varepsilon(|x|) = \log(1+\frac{x^2}{\sigma})$
- Lorentzian kernel: $\varepsilon(x) = \begin{cases} x^2 & x < \sigma \\ 2s|x| - \sigma^2 & x \geq \sigma \end{cases}$

In entrambi i casi, σ é un coefficiente da determinare sperimentalmente, mentre x é come sempre la distanza tra le coppie di punti.

Nel caso in questione, grazie all'utilizzo del filtro mediano e dell'eliminazione delle code, non si sono viste differenze sostanziali tra i vari kernel, ma si é osservato che quello che influenza maggiormente i risultati sono i pesi, descritti nel seguito. É da notare, che è possibile adoperare i kernel 2 e 3 solo nel caso di minimizzazione numerica, e non in quella analitica.

3.2.2.4 Funzione costo con differenti pesi

In 3.2.1 é stato descritto come l'algoritmo minimizza la funzione:

$$E_i(a) = \omega_i \varepsilon(P_{A,i} - O(a, P_{B,i}))$$

in cui il coefficiente ω_i rappresenta un peso assegnato ad ogni coppia di punti.

Sono state provate 5 formulazioni diverse di pesi:

1. pesi relativi allo scarto massimo (inversamente proporzionali alla distanza):
$$\omega_i = 1 - \frac{d_i}{d_{\max}}$$
2. pesi relativi ad uno scarto fisso:
$$\omega_i = \begin{cases} 1 & d_i < k d_{\max} \\ 10 & altrimenti \end{cases}$$
3. eliminazione di una percentuale di valori: vengono ordinate le coppie in base alla distanza, e ω_i = 1 se la i-esima coppia fa parte del n % delle distanze minime, altrimenti ω_i = 10;
4. tutti i dati con lo stesso peso: $\omega_i = 1$
5. pesi relativi allo scarto massimo (direttamente proporzionali alla distanza):
$$\omega_i = \frac{d_i}{d_{\max}}$$
definendo con d_i la distanza tra i punti della coppia in esame, $d_{\max}$ é la distanza massima tra tutte le coppie di punti.

I coefficienti *k* ed *n* vanno determinati sperimentalmente, qui k=0.7 ed n=70%. Nei pesi 2 e 3, in caso si voglia ridurre l'influenza di una coppia non si assegna peso nullo come si fa di solito nelle medie pesate, bensí un peso elevato: diversamente, l'algoritmo divergerebbe verso zone in cui gli accoppiamenti danno distanze elevate dove molti pesi valgono 0, e quindi effettivamente la funzione raggiungerebbe un minimo, ma che non sarebbe quello cercato.

Analisi condotte su piú ambienti hanno mostrato come in generale il peso di tipo 5 é quello che diverge maggiormente dalla soluzione, o che comporta sarti i maggiori.

L'utilizzo del peso 4 (tutti unitari) permette di avere una risoluzione analitica di facile implementazione, come quella riportata in 3.2.1.2.

3.3 Analisi di incertezza

É stata condotta una analisi per stimare l'incertezza dell'algoritmo. La taratura é stata eseguita ponendo il laser su una guida graduata, nella quale era possibile una traslazione ed una rotazione. La prova é stata condotta acquisendo una scansione dell'ambiente da 2 pose diverse del LIDAR come riassunto in Tabella 3, seguendo la simbologia di Figura 43:

Tabella 3: valutazione incertezza algoritmo ICP, posa laser

	θ (deg)	d [cm]
Posa 0	-10	0
Posa 1	20	10

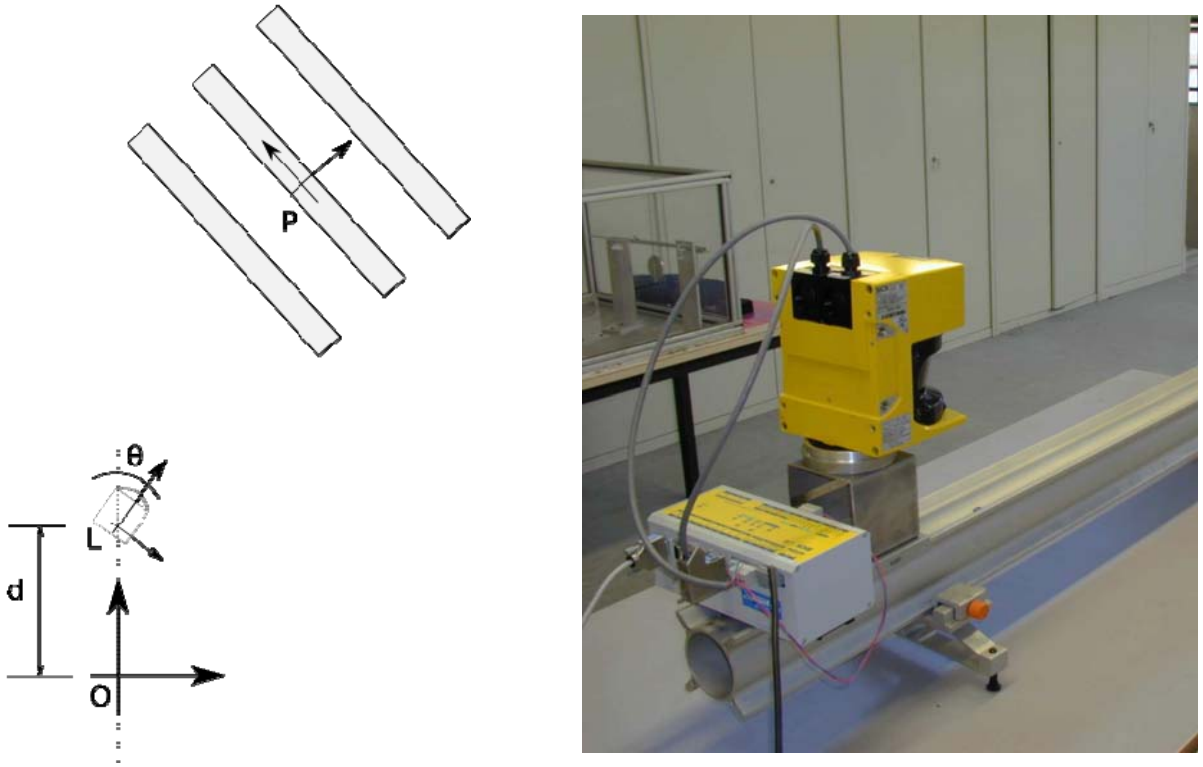


Figura 43: Rappresentazione apparato sperimentale per valutazione incertezza (sx), e apparato sperimentale (dx)

Il sistema di riferimento utilizzato, per semplicità, é quello solidale con la prima scansione, di conseguenza il risultato atteso dall'algoritmo, ossia la differenza di posa tra le due acquisizioni, é il seguente:

- $\Delta\theta = 30° \deg$
- $\Delta X = 10\sin(10°) = 17.4mm$
- $\Delta Y = 10\cos(10°) = 98.5mm$

Per ogni posa sono state eseguite almeno 50 scansioni, ed in tutto sono stati acquisiti 15 ambienti diversi, per ottenere un'indicazione sia sulla ripetibilitá che sulla riproducibilitá. Inoltre, sono stati valutati tutti i parametri di influenza per l'algoritmo, e cioé il tipo di minimizzazione, numerica o analitica, e nel primo caso i 5 tipi di peso ed i 3 tipi di kernel, come spiegato in precedenza.

3.3.1 Risultati con minimizzazione numerica

Nel seguito, a titolo di esempio vengono mostrati alcuni grafici ottenuti con l'algoritmo di minimizzazione numerica Patternsearch. Il primo dei grafici presenta i risultati dividendoli per ambiente sull'asse delle ascisse, per ogni ambiente sono mostrati i risultati delle 50 prove, nelle ordinate viene mostrata la differenza con il valore di riferimento relativa ad ogni prova e il valore mostrato in linea continua è la media di tali valori. Il secondo è l'istogramma globale dei risultati, e raggruppa tutti gli ambienti. I risultati saranno commentati nel seguito. Ogni coppia di grafici rappresenta una coppia di peso-kernel, come spiegato nelle didascalie.

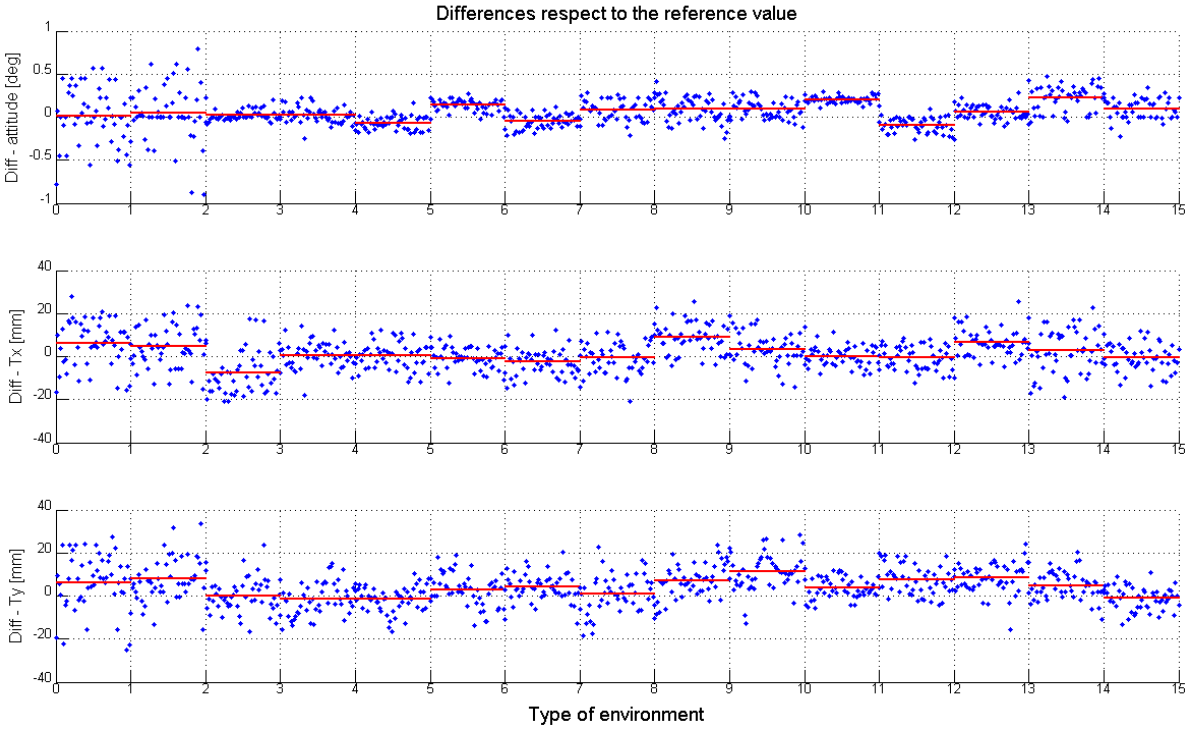


Figura 44: differenze rispetto al valore di riferimento plottato per ambiente [peso 1, kernel 1]

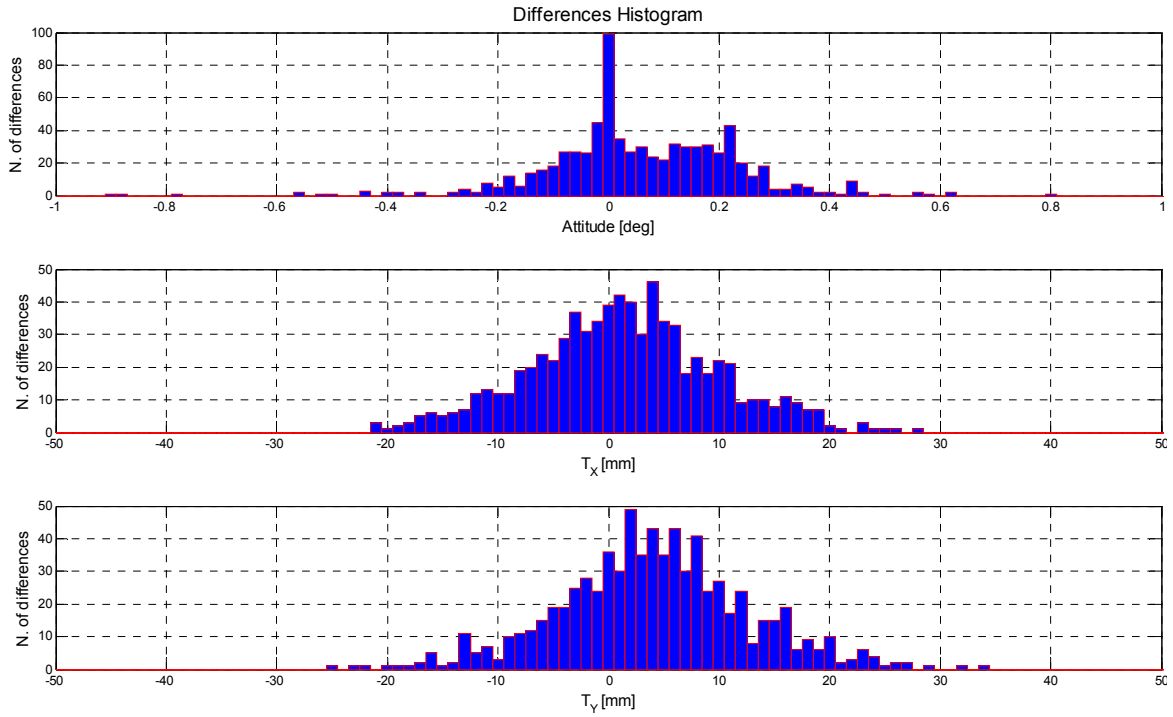


Figura 45: istogramma globale risultati [peso 1, kernel 1]

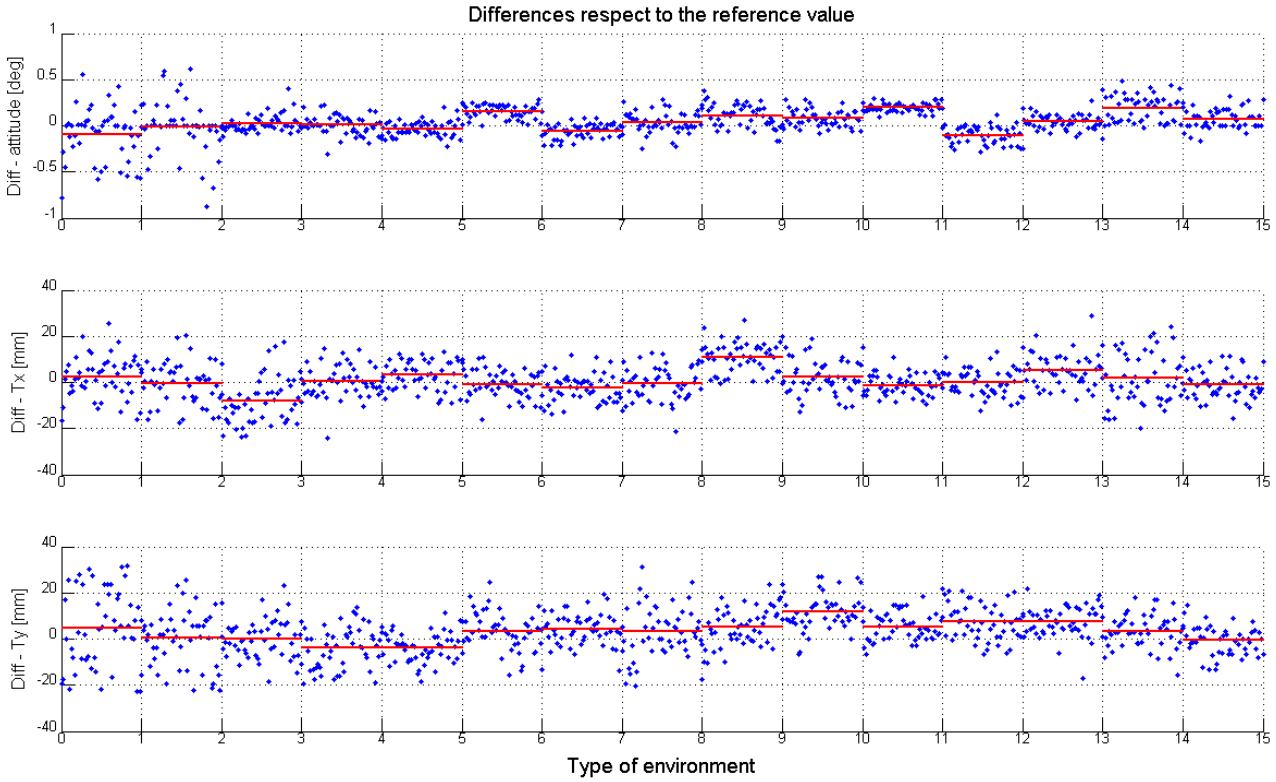


Figura 46: differenze rispetto al valore di riferimento plottato per ambiente [peso 1, kernel 2]

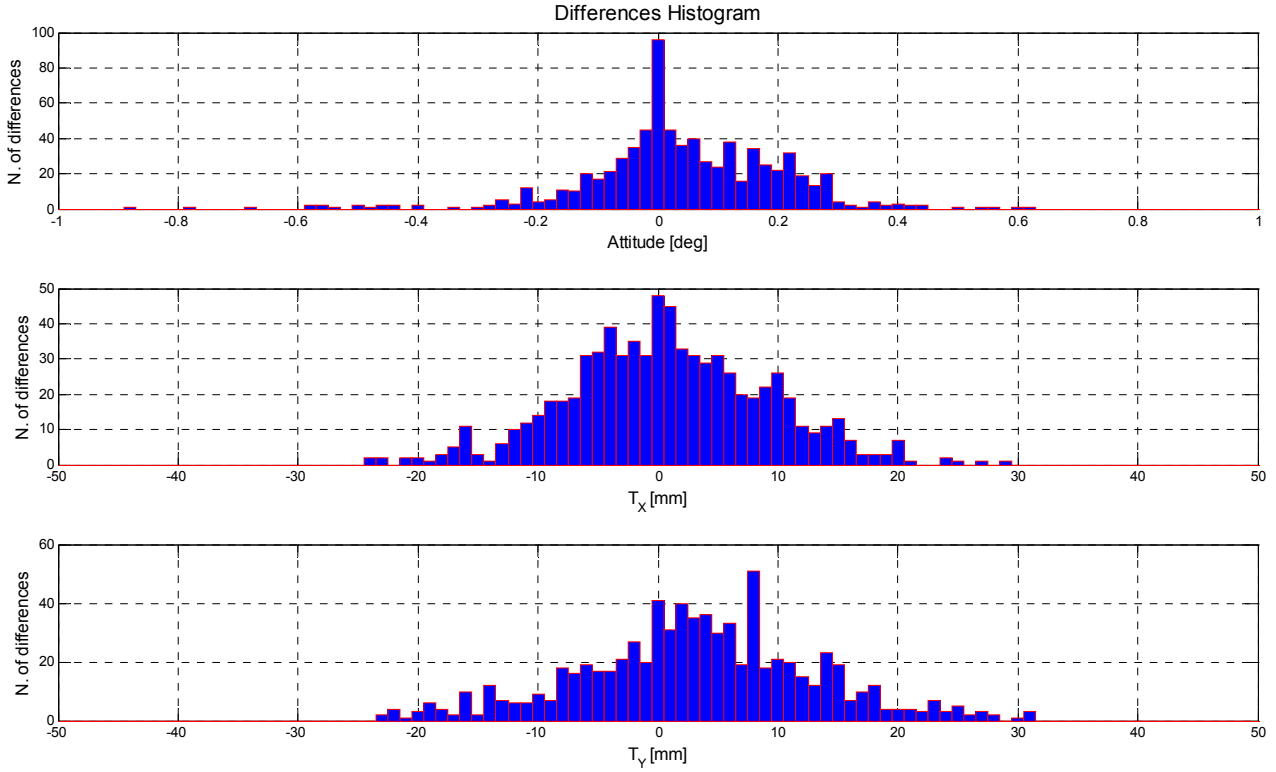


Figura 47: istogramma globale risultati [peso 1, kernel 2]

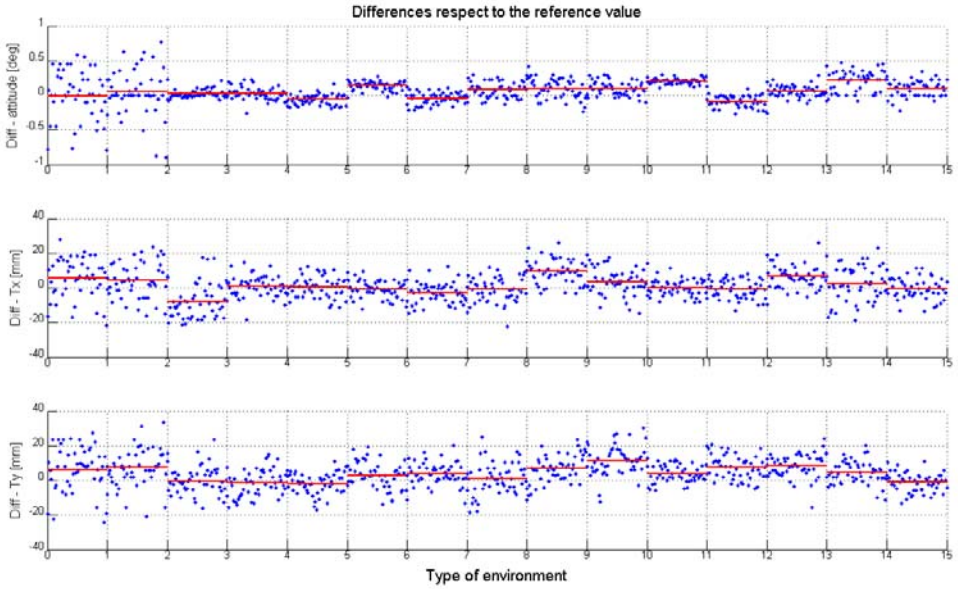


Figura 48: differenze rispetto al valore di riferimento plottato per ambiente [peso 1, kernel 3]

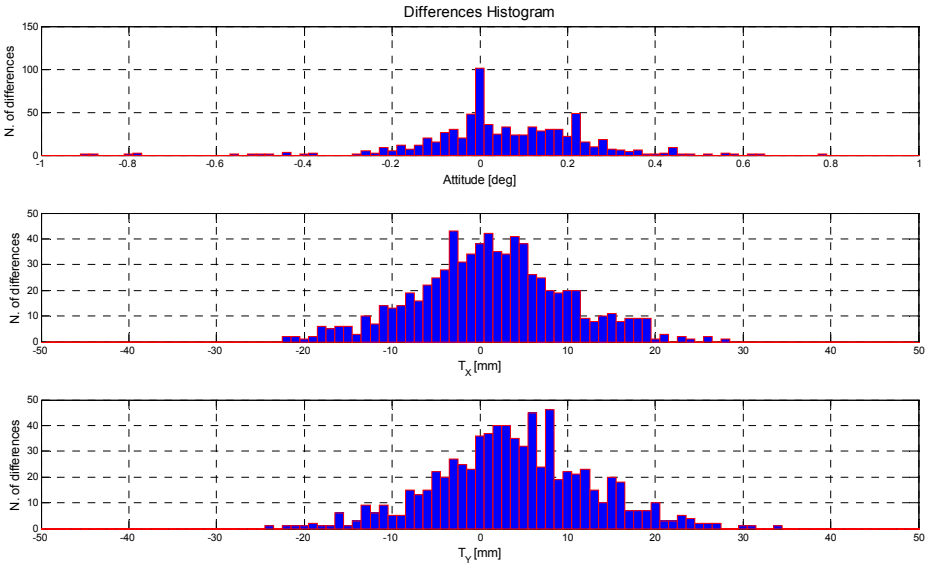


Figura 49: istogramma globale risultati [peso 1, kernel 3]

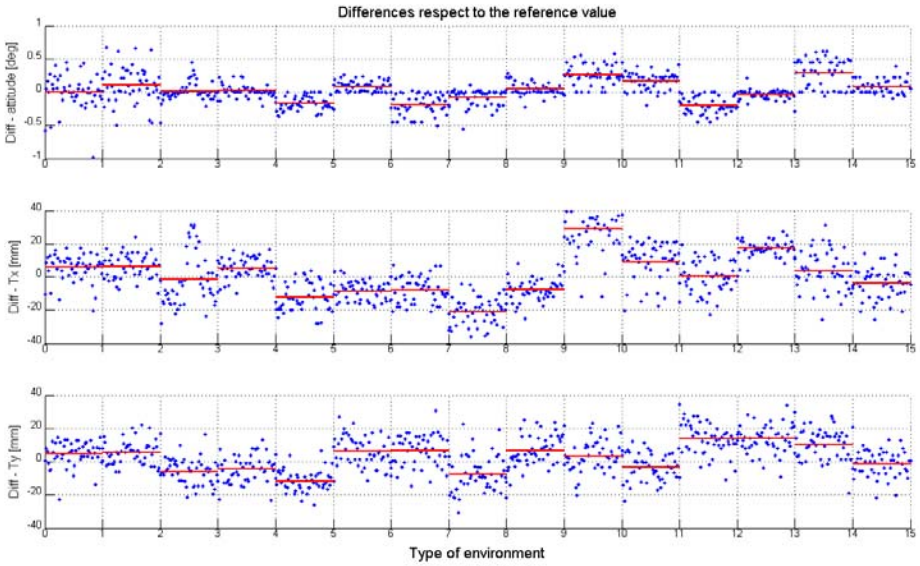


Figura 50: differenze rispetto al valore di riferimento plottato per ambiente [peso 2, kernel 1]

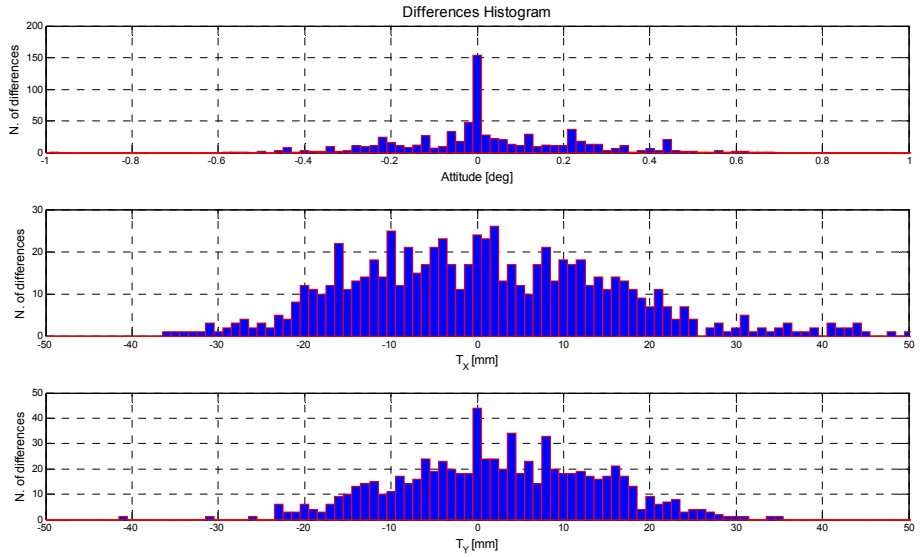


Figura 51: istogramma globale risultati [peso 2, kernel 1]

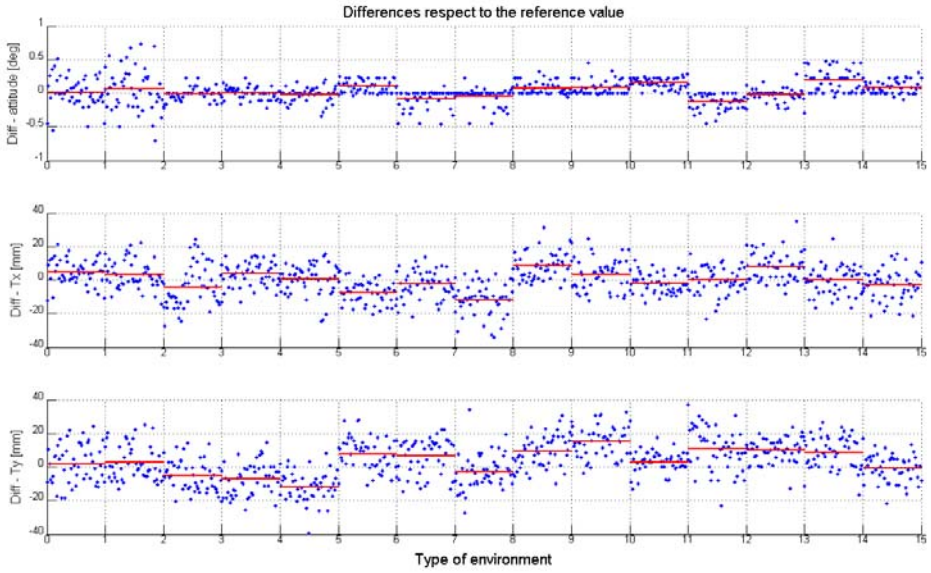


Figura 52: differenze rispetto al valore di riferimento plottato per ambiente [peso 2, kernel 2]

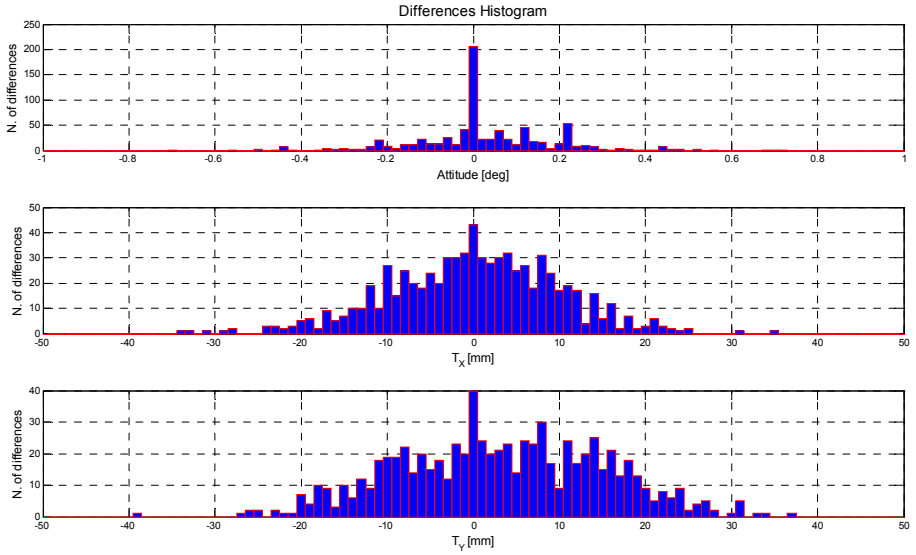


Figura 53: istogramma globale risultati [peso 2, kernel 2]

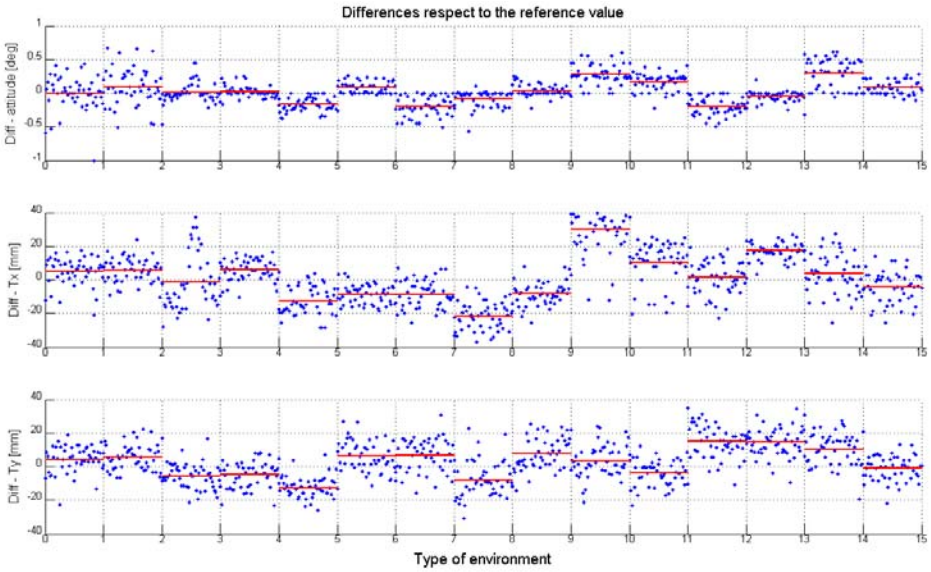


Figura 54: differenze rispetto al valore di riferimento plottato per ambiente [peso 2, kernel 3]

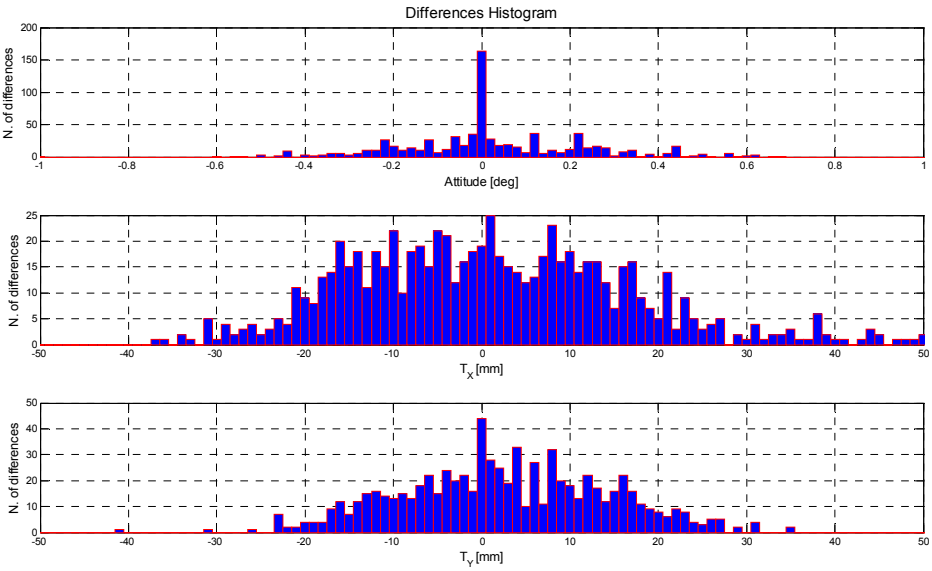


Figura 55: istogramma globale risultati [peso 2, kernel 3]

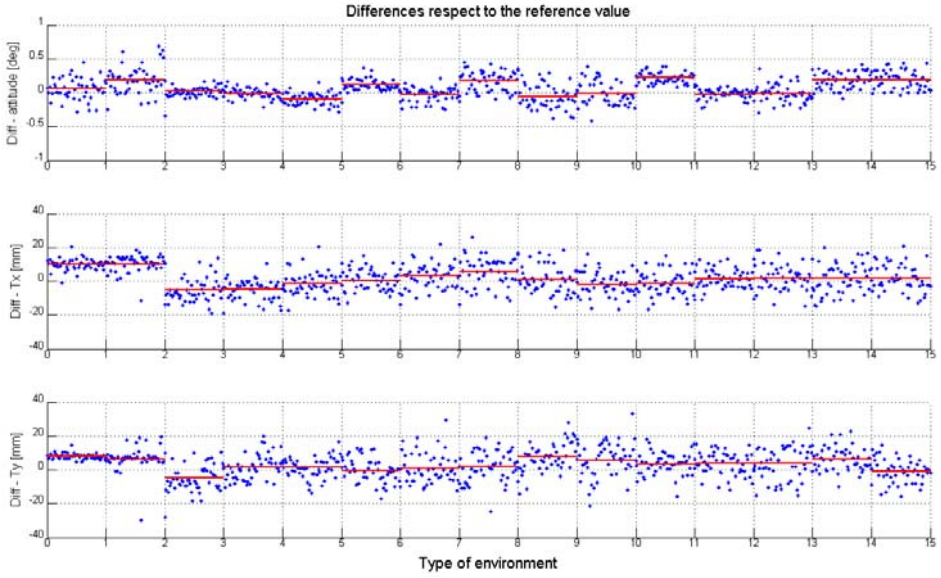


Figura 56: differenze rispetto al valore di riferimento plottato per ambiente [peso 3, kernel 1]

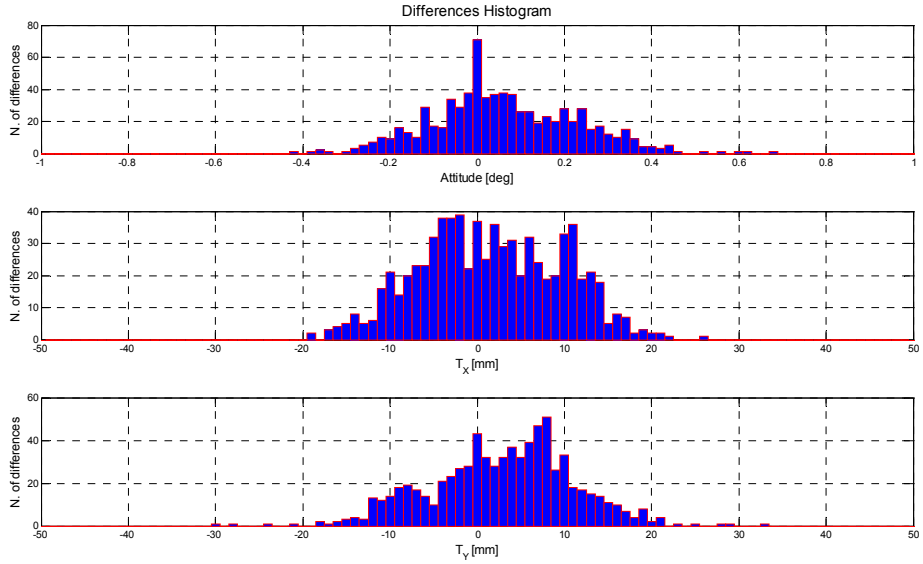


Figura 57: istogramma globale risultati [peso 3, kernel 1]

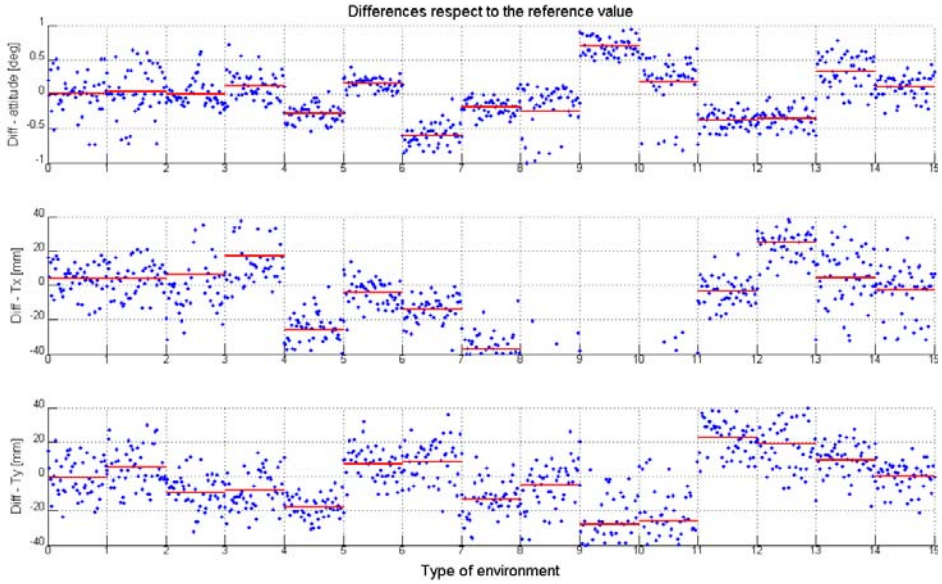


Figura 58: differenze rispetto al valore di riferimento plottato per ambiente [peso 5, kernel 1]

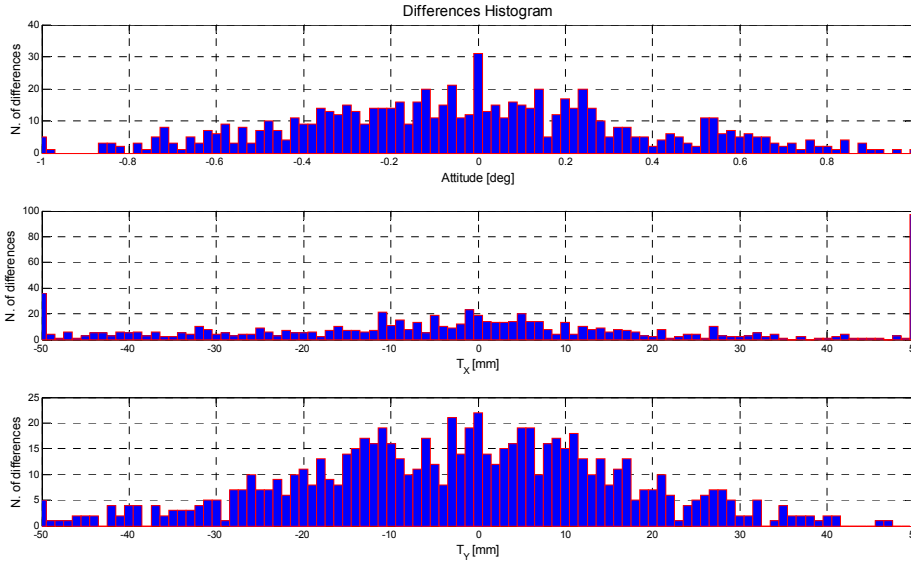


Figura 59: istogramma globale risultati [peso 5, kernel 1]

3.3.2 Risultati con minimizzazione analitica

Nel seguito, a titolo di esempio vengono mostrati alcuni grafici ottenuti con l'algoritmo di minimizzazione analitica. Il primo dei grafici presenta i risultati dividendoli per ambiente sull'asse delle ascisse, per ogni ambiente sono mostrati i risultati delle 50 prove, nelle ordinate viene mostrata la differenza con il valore di riferimento relativa ad ogni prova e il valore mostrato in linea continua è la media di tali valori. Il secondo è l'istogramma globale dei risultati, e raggruppa tutti gli ambienti. I risultati saranno commentati nel seguito. Ogni coppia di grafici rappresenta una coppia di peso-kernel, come spiegato nelle didascalie.

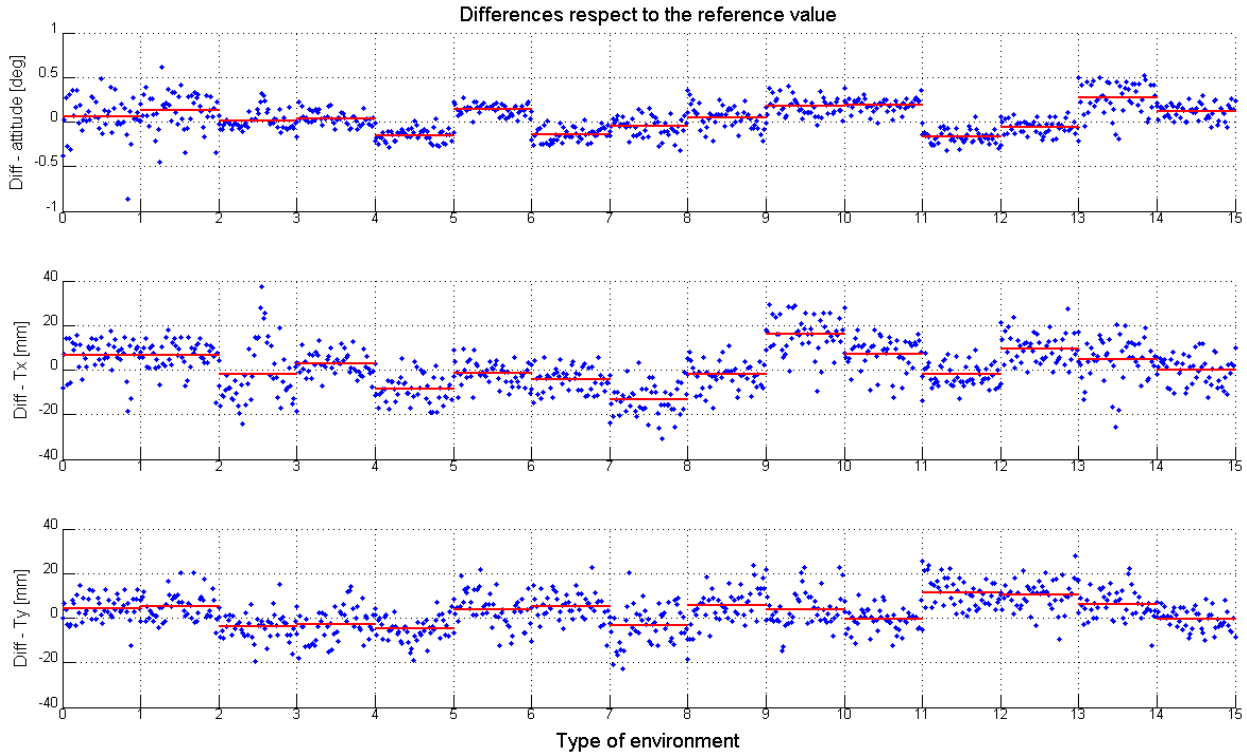


Figura 60: differenze rispetto al valore di riferimento plottato per ambiente [peso 1, kernel 1]

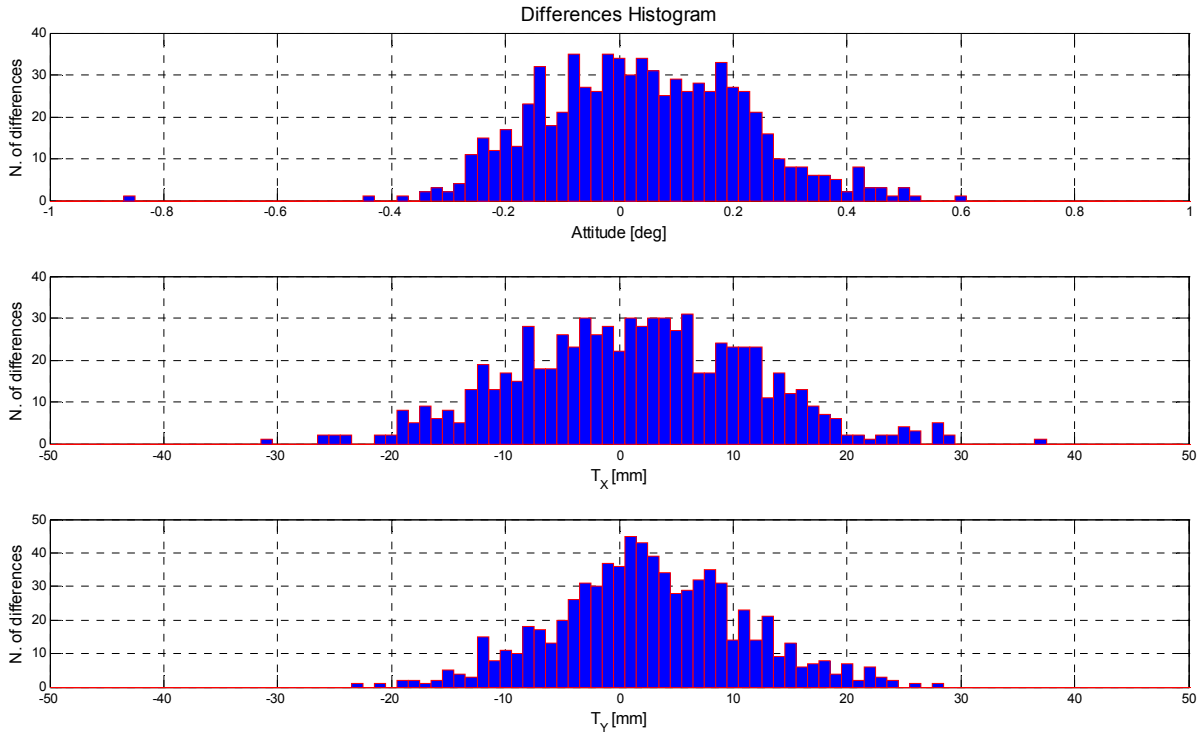


Figura 61: istogramma globale risultati [peso 1, kernel 1]

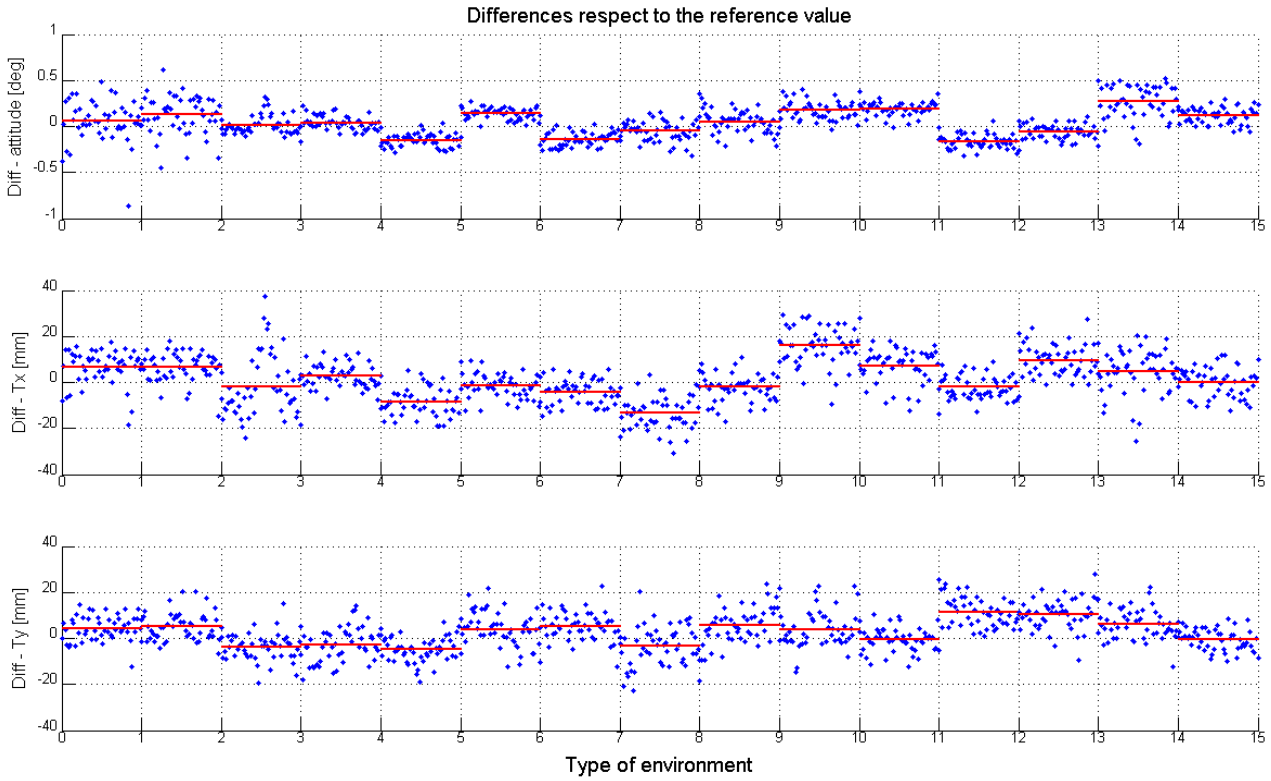


Figura 62: differenze rispetto al valore di riferimento plottato per ambiente [peso 4, kernel 1]

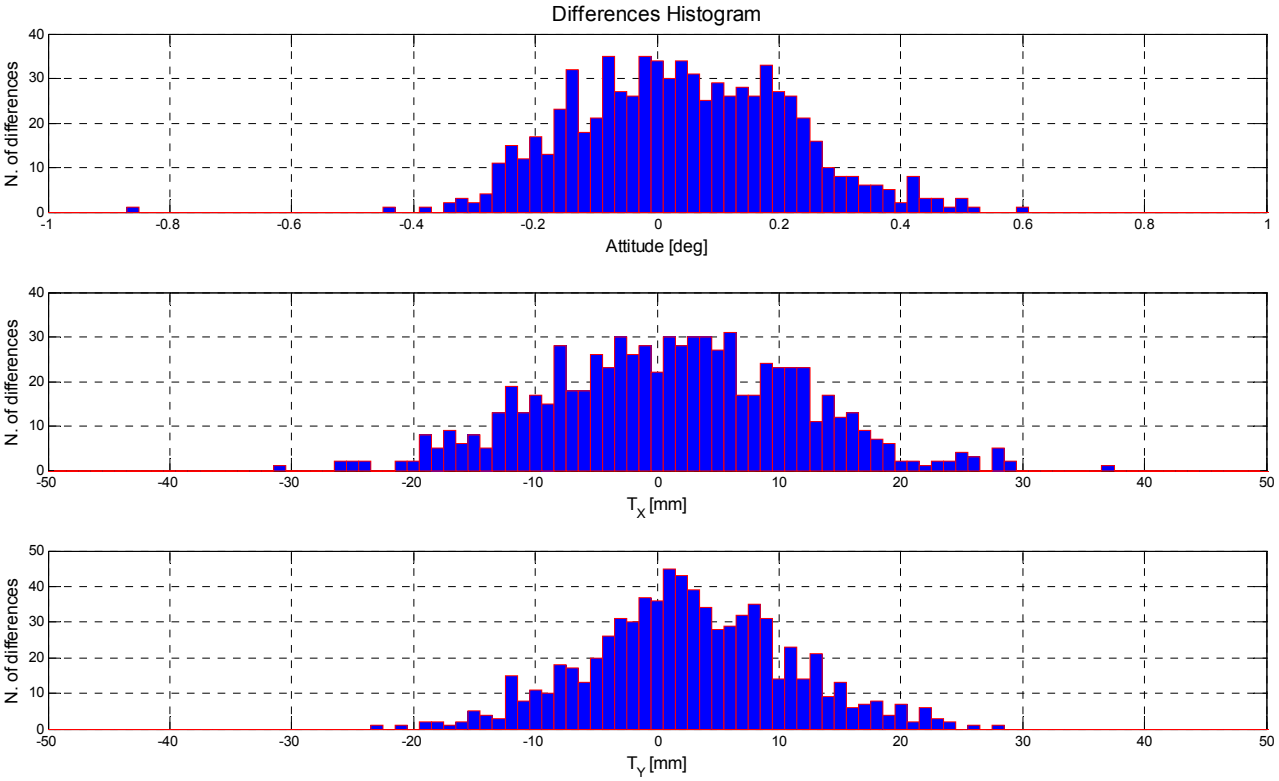


Figura 63: istogramma globale risultati [peso 4, kernel 1]

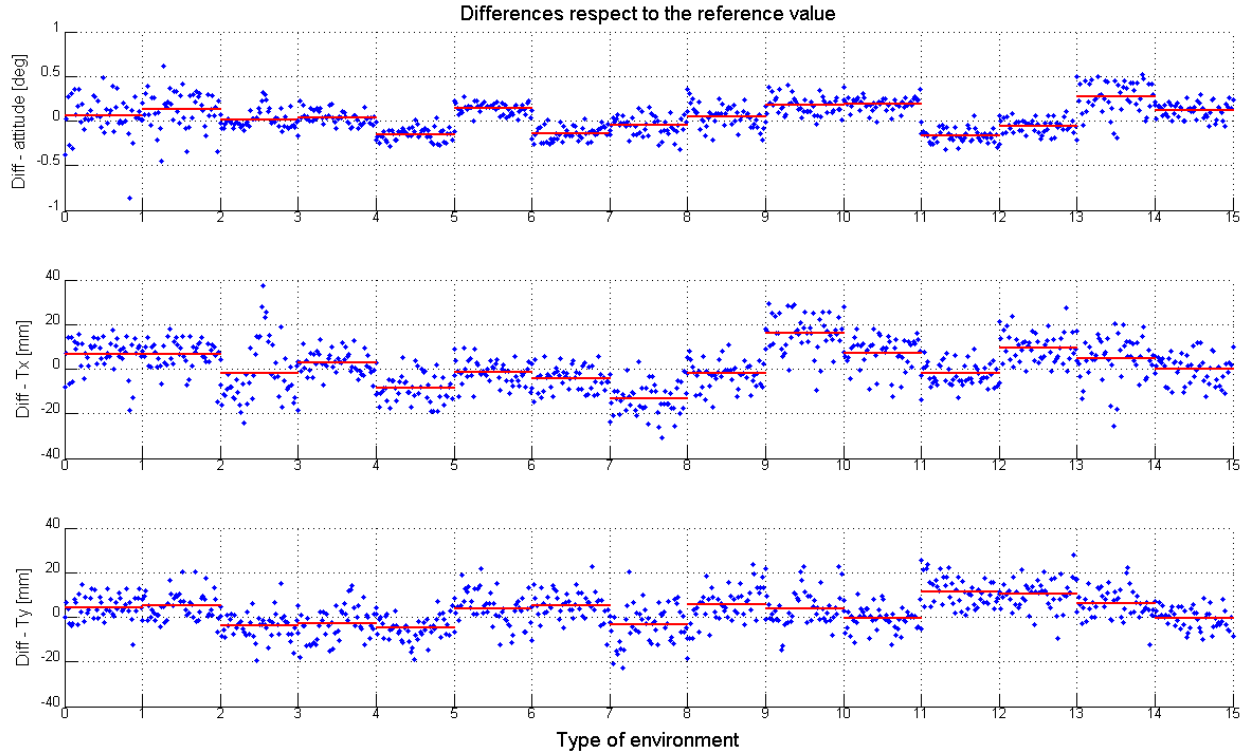


Figura 64: differenze rispetto al valore di riferimento plottato per ambiente [peso 5, kernel 1]

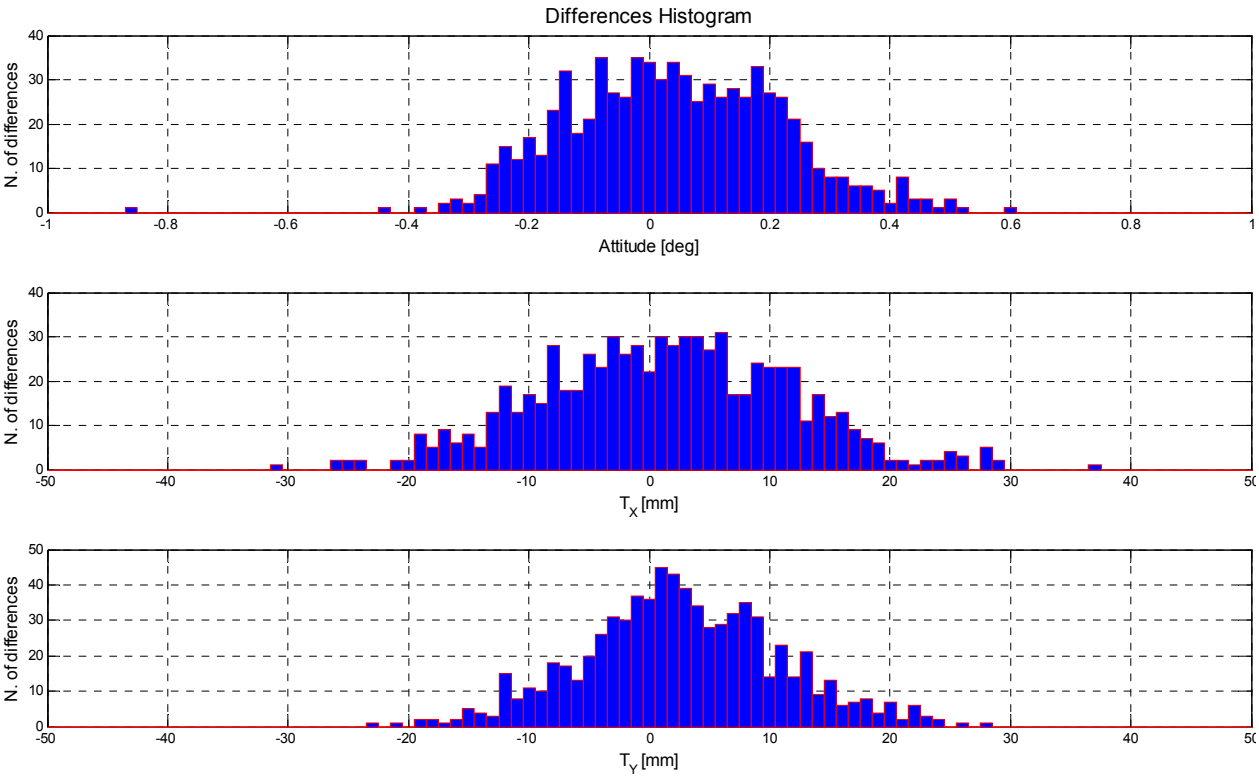


Figura 65: istogramma globale risultati [peso 5, kernel 1]

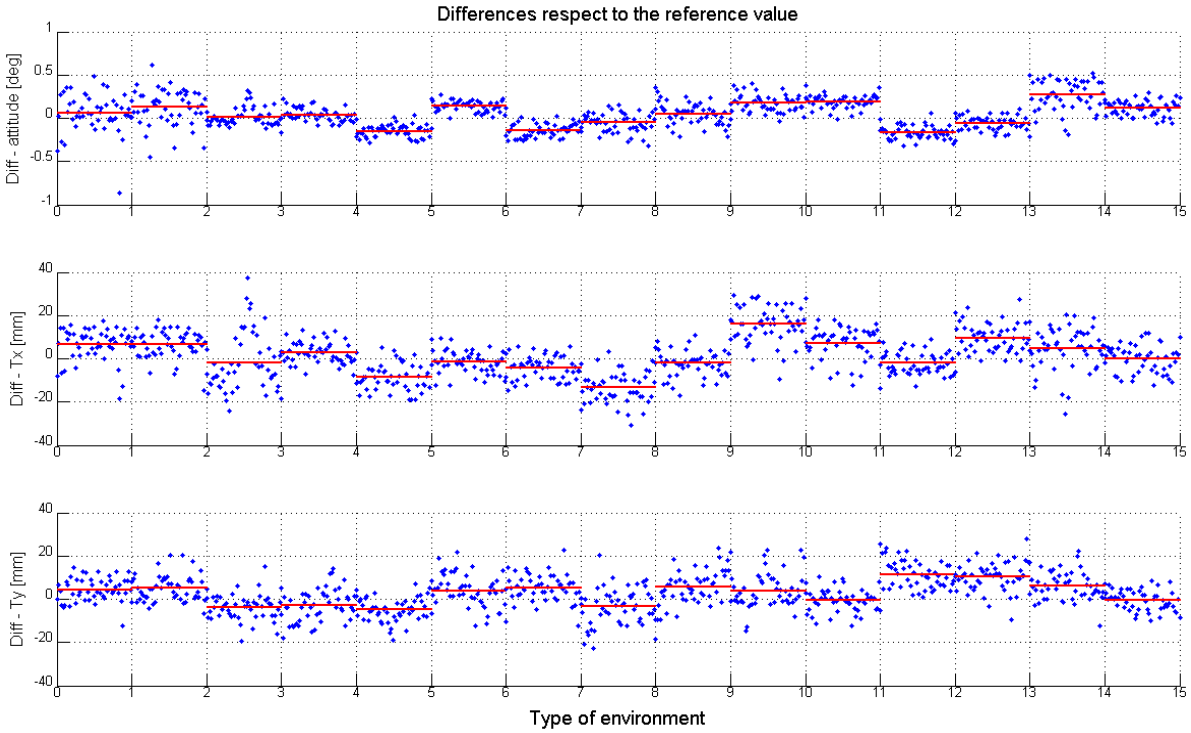


Figura 66: differenze rispetto al valore di riferimento plottato per ambiente [peso 5, kernel 3]

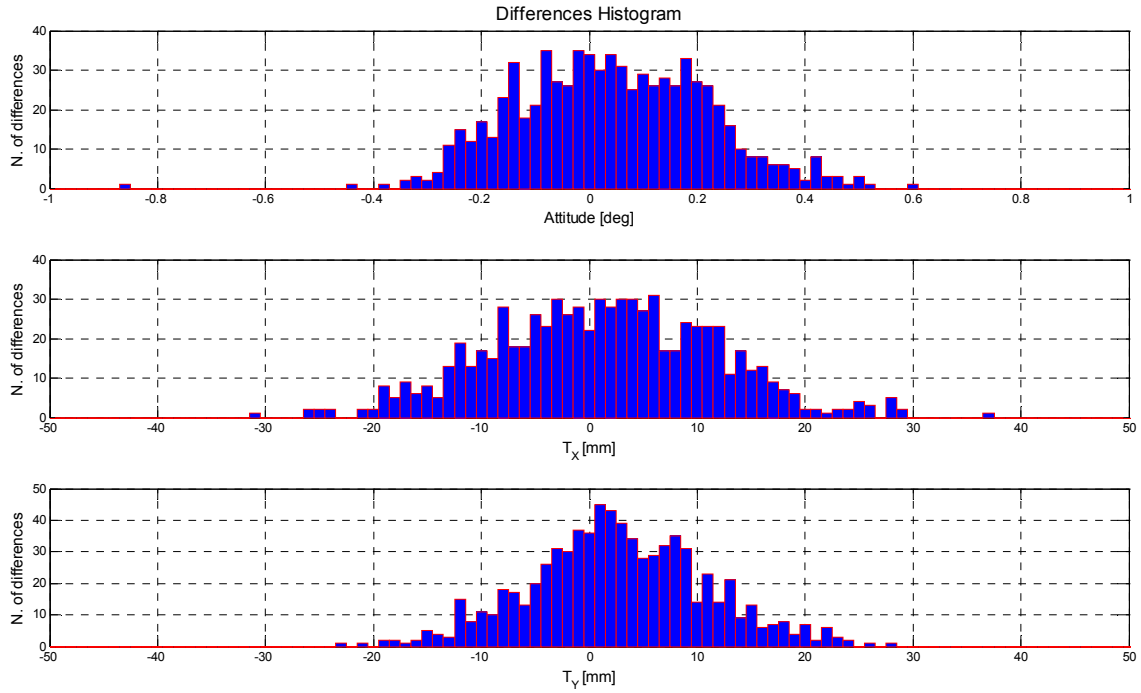


Figura 67: istogramma globale risultati [peso 5, kernel 3]

3.3.3 Commento risultati

Come si può notare, variare kernel non porta a differenze apprezzabili, nel senso che tutti e tre danno scarti (a paritá di peso utilizzato) comparabili. Lo stesso discorso non vale invece per i pesi, dove si nota come il peso 5 porta ad una notevole dispersione degli scarti. Il peso 4 (tutti unitari) permette una implementazione analitica, ed infatti si nota come i risultati numerici con kernel 1 (ma anche con gli altri 2) e questo tipo di peso siano molto simili.

I migliori pesi, osservando i risultati, sembrano essere il primo ed il secondo, ma quest'ultimo richiede un tuning particolare dei parametri, come il valore oltre il quale una coppia ha 0 come peso.

Capitolo 4

Ricerca di oggetti mediante Lidar 2D

In questo capitolo verrà descritto un algoritmo di ricerca di oggetti di forma qualsiasi purché descrivibile tramite segmenti, basato sul matching di scansioni del Lidar.

4.1 Algoritmo di ricerca di oggetti

La localizzazione di un oggetto all'interno di una scansione 2D dell'ambiente ha una duplice utilità. Innanzitutto la ricerca dell'oggetto stesso può essere molto utile in varie applicazioni, come ad esempio la movimentazione dei pallet nei magazzini che spesso vengono posizionati senza accortezze particolari. In secondo luogo è possibile sfruttare l'informazione per localizzare le scansioni ottenute rispetto ad una scansione di partenza: dalle differenze di posa dell'oggetto trovate tra 2 scansioni, sapendo che l'oggetto non si é mosso, si ricava la posa della seconda acquisizione rispetto alla prima e quindi del robot rispetto alla posa iniziale.

L'algoritmo sviluppato si basa su un principio di minimizzazione energetica, in cui viene costruita una funzione di correlazione tra i segmenti dell'oggetto teorico in una specifica posa e i punti scansionati: minimizzando questa funzione, si ottiene il risultato ricercato. È un metodo iterativo, e dopo ogni ricerca viene fatto un controllo sul risultato, simulando una scansione dell'oggetto nella posa trovata e confrontando questa con le acquisizioni reali.

4.1.1 Descrizione algoritmo

Lo scopo dell'algoritmo é verificare la presenza di un oggetto all'interno di una scansione 2D, e trovarne la posa (posizione ed assetto). Il metodo seguito si basa su principi simili a quelli descritti nel capitolo precedente per confrontare 2 scansioni e determinarne la differenza di posa (Biber, 2003). Approcci simili sono stati sviluppati di recente anche da (Lecking, Wulf, Wagner, 2006).

Gli input dell'algoritmo sono una scansione 2D dell'ambiente, il modello dell'oggetto da trovare ed i parametri del laser impiegato (FOV, risoluzione angolare e range massimo). L'oggetto da ricercare è descritto da un modello costituito da segmenti, ogni forma così rappresentabile può essere cercata. I vari segmenti vengono descritti in una matrice in cui per ogni riga: le prime due colonne rappresentano i coefficienti θ e ρ [rad, m] della linea che rappresenta i segmenti secondo la relazione $\cos(\theta)X + \sin(\theta)Y = \rho$, le colonne 3 ÷ 6 rappresentano le coordinate cartesiane (X1Y1, X2Y2) degli estremi del segmento, espressi in metri. A titolo di esempio, il rettangolo di Figura 68:

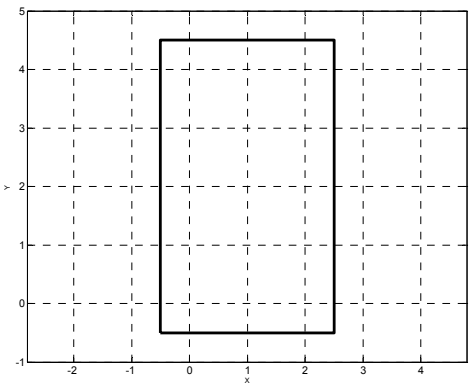


Figura 68: esempio di rettangolo

viene descritto dalla seguente matrice:

0	2.5	2.5	-0.5	2.5	4.5
1.5708	4.5	2.	4.5	-0.5	4.5
3.1416	0.5	-0.5	4.5	-0.5	-0.5
4.7124	0.5	-0.5	-0.5	2.5	-0.5

In questo lavoro l'oggetto cercato è un euro pallet, rappresentato in Figura 69, in quanto si tratta di un caso di rilevante interesse in ambito industriale. Il modello rappresenta i 9 blocchetti di supporto, le cui dimensioni sono state tratte da http://it.wikipedia.org/wiki/Pallet e il risultato è raffigurato in Figura 70.

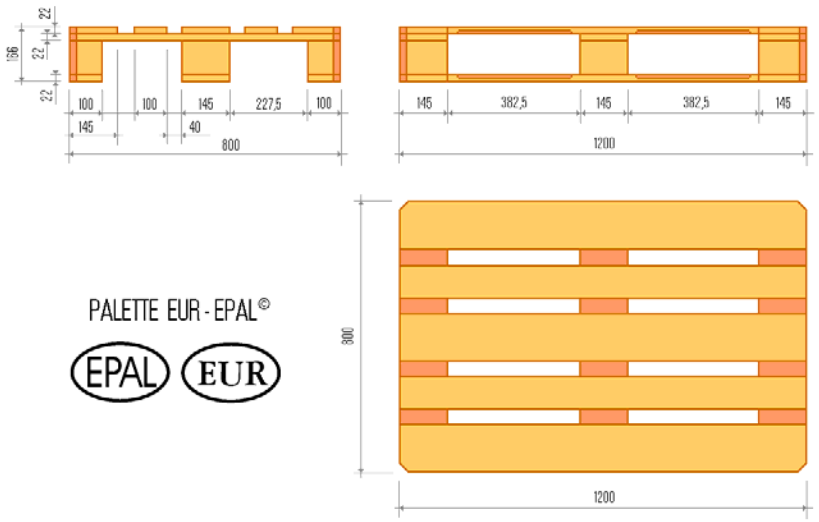


Figura 69: modello europeo di pallet

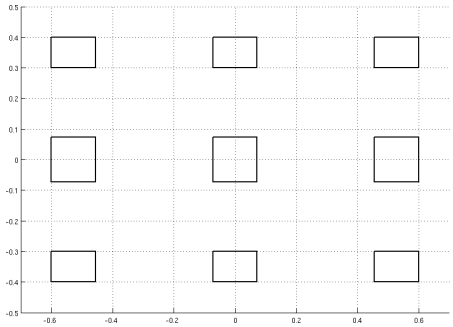


Figura 70: Linee del pallet usate come modello, come si vede vengono modellati i piedini di supporto

In pseudocodice, l'algoritmo è composto dai seguenti passi (schematizzati come flowchart in Figura 71):

Algoritmo di ricerca oggetti → Dato un modello X e un oggetto P:

1. A partire da una trasformazione iniziale: p_0
2. Minimizzazione di una funzione costo→ p_{opt}
3. Rototraslazione del modello con i parametri ottimizzati
4. Simulazione della scansione LIDAR con la posa calcolata del modello
5. Matching tra la scansione reale e quella simulata e calcolo della distanza (simile all'ICP)
6. Se una certa percentuale % delle distanze è sotto a una determinata soglia: stop, altrimenti $p_0 = p_{opt}$ e ritorna al passo 1

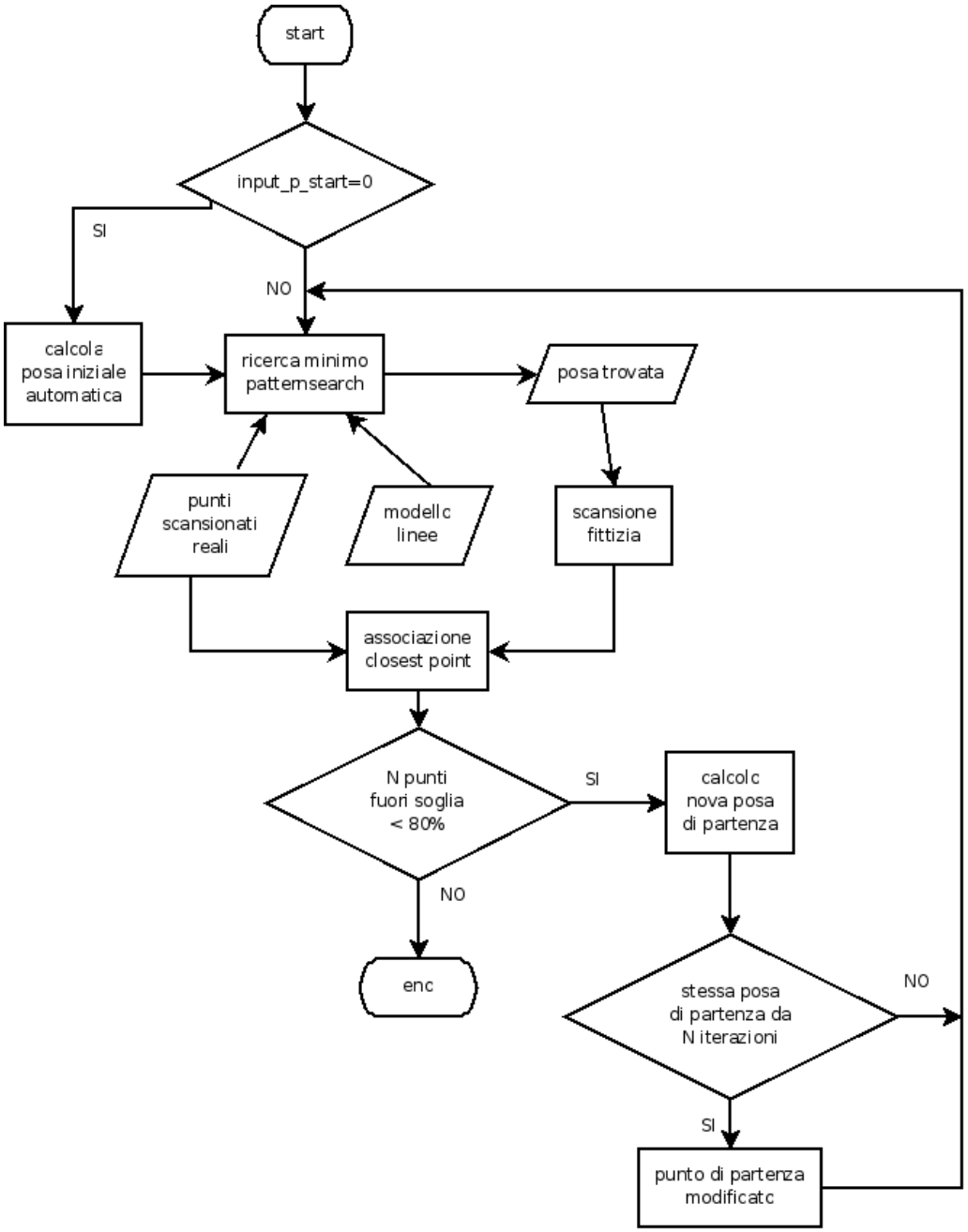


Figura 71: Flowchart algoritmo di ricerca pallet

L'algoritmo comincia la ricerca a partire da una posa di partenza, e viene poi cercato il minimo di una funzione costo (analizzata in 4.1.2). Sono state provate diverse strade per la minimizzazione: inizialmente si è utilizzato l'algoritmo patternsearch.m (si veda 3.2.1.1), e poi sono stati implementati differenti algoritmi genetici, analizzati in 4.1.3. A partire dalla posa ottimizzata così calcolata viene eseguita una scansione fittizia del modello e si associano i punti così ottenuti con quelli reali in base alla distanza (ogni punti del primo gruppo viene associato al più vicino del secondo, con un passaggio simile a quanto si fa nell'ICP, come mostrato in Figura 72). Da queste associazioni si ricavano le distanze tra le varie coppie, e se una percentuale sufficientemente alta (80%) di questi valori é sotto una soglia (5 cm usando scansioni reali, 3 cm simulandole) l'algoritmo ha terminato, e la posa corretta é stata trovata. Il motivo della differenza di soglia tra scansioni reali e simulate risiede nel fatto che le simulazioni considerano un modello di laser semplificato (si veda 3.1.1 e 3.1.2) che non comprende per esempio l'angolo di incidenza tra il raggio e la superficie o le caratteristiche fisiche (colore, coefficiente di riflessione/diffrazione) della stessa; l'incertezzae che si ritrova, dunque, é minore nelle scansioni simulate.

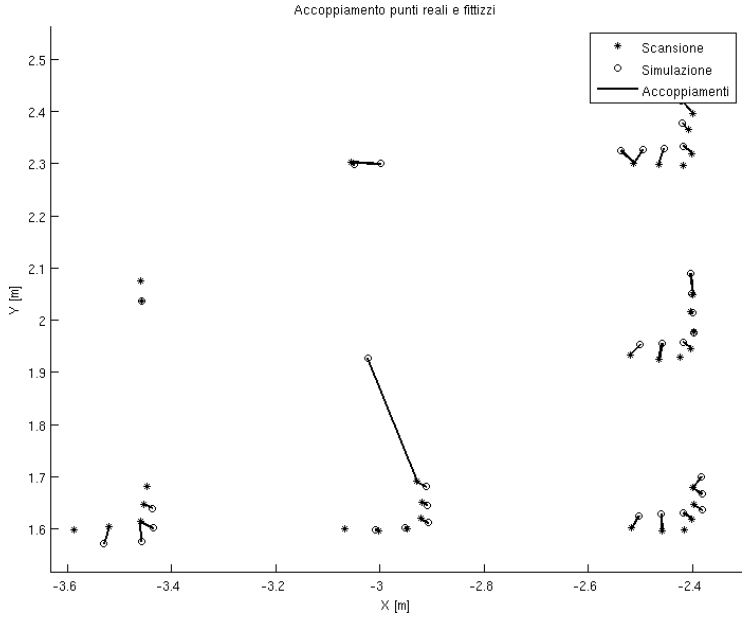


Figura 72: Accoppiamenti tra punti fittizi e reali

Sono state provati altri metodi di controllo di convergenza, tra i quali le distanze tra i punti fittizi e i punti reali correlati; l'esperienza ha mostrato che questo portava spesso a dei falsi positivi, mentre finora il metodo sopra spiegato non ha portato a errori, sia in termini di falsi positivi che di falsi negativi. Nel caso in cui il check delle distanze non dia esito positivo, si procede al calcolo di una nuova posa iniziale, nel seguente modo: si calcolano i baricentri dei punti della scansione fittizia e di quelli reali collegati a questa, e si trasla la posizione di partenza secondo il vettore che congiunge il baricentro dei punti fittizi con quello dei punti reali.

Esiste inoltre un problema di tipo numerico: può accadere che i risultati portino sempre alla stessa soluzione, anche se questa non é quella cercata (e lo si capisce dal controllo descritto sopra sui baricentri); in questo caso con alta probabilità ci si trova in presenza di minimi locali. Per ovviare a questo fatto, la traslazione della posizione di partenza viene effettuata lungo il versore che congiunge i baricentri (come prima) ma il modulo dello spostamento é pari ad un valore scelto (10 cm) moltiplicato N, dove N rappresenta il numero di volte che la soluzione é stata ripetuta. Si é visto che in questo modo é possibile aiutare l'algoritmo di minimizzazione ad oltrepassare i minimi locali, soprattutto nel caso in cui la distanza tra i baricentri é minima (qualche mm).

4.1.2 Analisi della funzione costo

La funzione da minimizzare restituisce uno scalare a partire dal modello di oggetto da cercare e dai punti della scansione. Il modello deve essere descritto tramite segmenti, come già spiegato in 4.1.1, la scansione invece é definita dalle coordinate cartesiane dei punti scansionati. La funzione costo ha la forma:

$$C(\phi_m, x_m, y_m) = -\sum_i A e^{Bd_i}$$

dove A e B sono dei coefficienti, mentre d_i rappresentano le distanze tra il punto reale in esame e la sua proiezione sulla linea più vicina. La funzione costo viene quindi valutata sommando i contributi di tutti i punti della scansione.

In particolare, la funzione costo viene valutata secondo il seguente metodo: ad ogni punto della scansione ne viene associato un altro, ottenuto come proiezione di questo sui segmenti del modello (si veda la Figura 73 A). Se la proiezione cade fuori degli estremi del segmento, verrà utilizzato l'estremo più vicino (come mostrato in Figura 73 B). Per ognuno dei punti della scansione, quindi, ci saranno tanti punti quante sono le linee del modello, e di questi viene scelto quello che ha distanza minore con il punto in analisi (si veda la Figura 73 C). Dei punti così associati viene quindi calcolata la distanza e quindi la funzione costo.

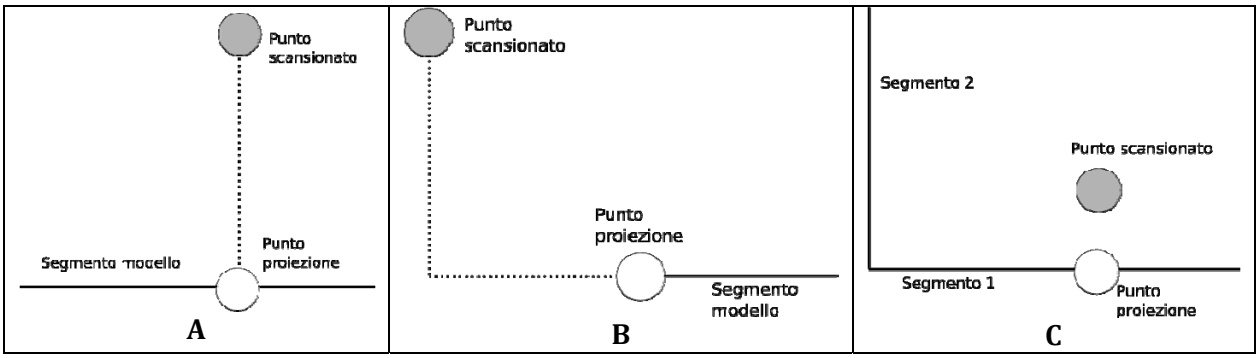


Figura 73: A=Proiezione di un punto su un segmento. B= Proiezione di un punto su un segmento: se questa cade fuori dagli estremi del segmento, verrà utilizzato l'estremo più vicino. C= Proiezione di un punto su un segmento: caso con piú linee.

La funzione costo é simile ad una gaussiana monodimensionale, e lo sarebbe a tutti gli effetti se si elevasse al quadrato il termine *dist* e si ponesse:

$$A = \frac{1}{\sqrt{2\pi}\sigma} \qquad \text{e} \qquad B = \frac{0.5}{\sigma^2}$$

dove σ rappresenta un ipotetico scarto quadratico. Si é visto che non utilizzando il quadrato nel termine esponenziale, l'operazione di ricerca di minimo é più rapida (1 ordine di grandezza sul numero di iterazioni). A titolo di esempio, si può notare l'andamento del valore della funzione costo (sull'asse Z) nelle figure Figura 74, Figura 75, Figura 76, Figura 77. Sono state ottenute ricercando un pallet (modello in Figura 70) in una scansione simulata, ponendo il modello in un angolo di assetto fisso e traslandolo (assi X e Y). Come si può notare (soprattutto dalla vista laterale) in entrambe le funzioni sono presenti un elevato numero di minimi locali, ma la distinzione dei picchi é più marcata utilizzando la funzione senza il termine esponenziale al quadrato.

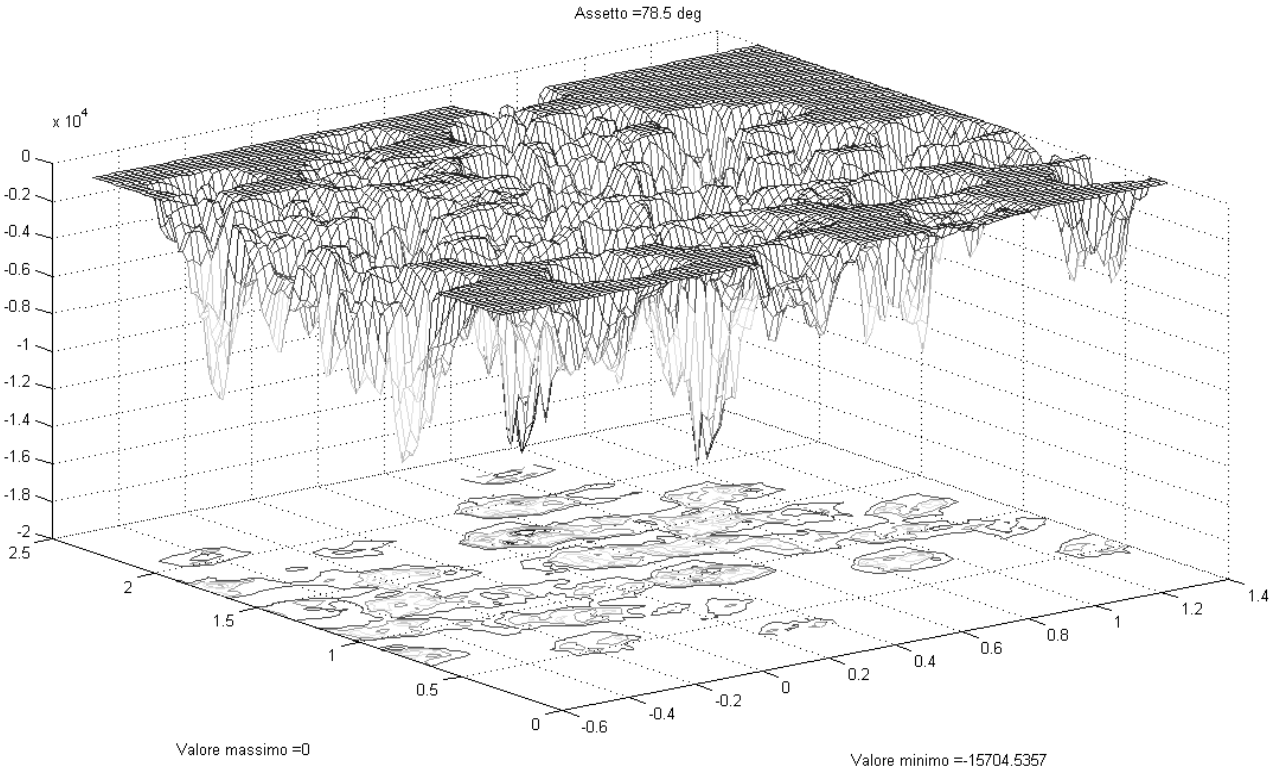


Figura 74: Funzione costo con il termine esponenziale al quadrato

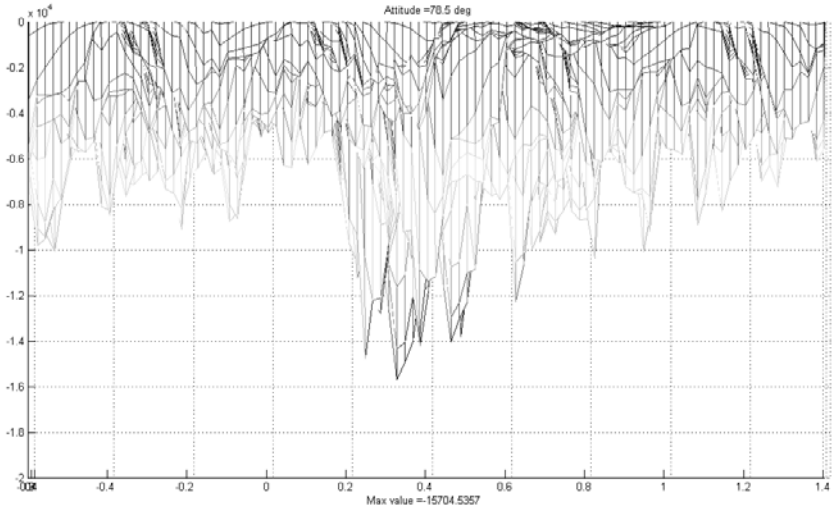


Figura 75: Funzione costo con il termine esponenziale al quadrato: vista di lato

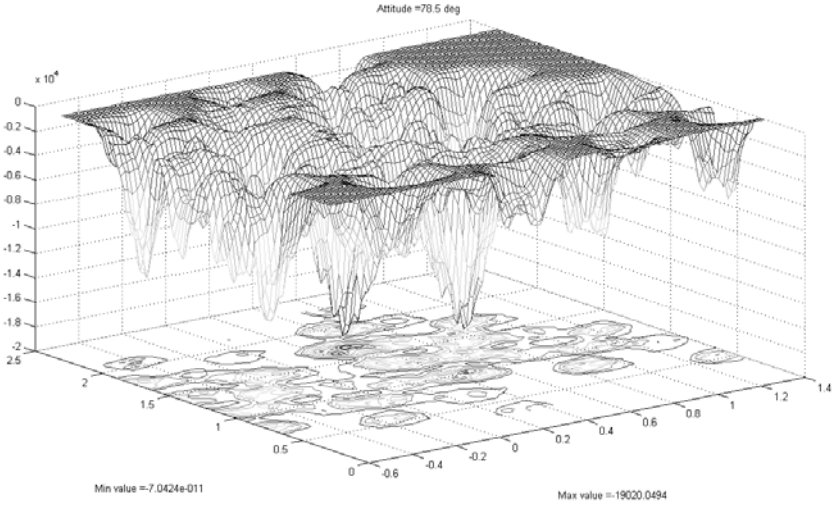


Figura 76: Funzione costo senza il termine esponenziale al quadrato

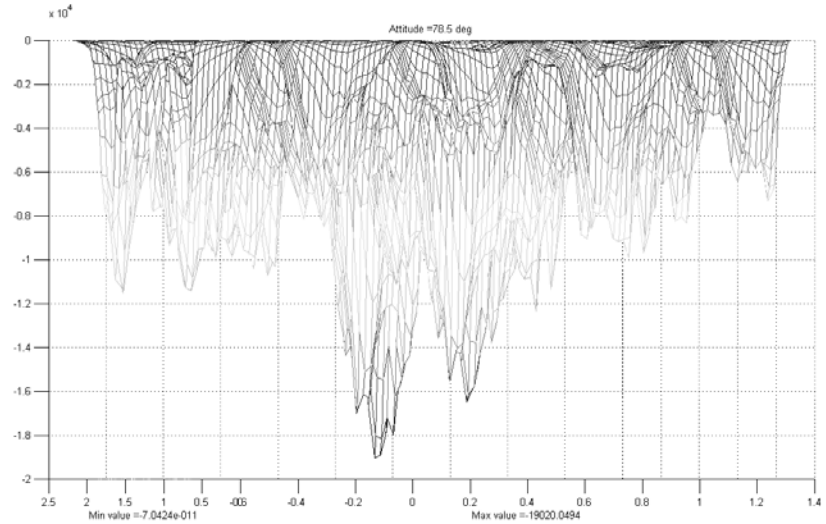


Figura 77: Funzione costo senza il termine esponenziale al quadrato: vista di lato

Le figure Figura 78 e Figura 79 rappresentano il valore della funzione costo per ogni punto dell'area (tra -4 e +4, con step di 0.05, sia in X che in Y), il modello utilizzato é un segmento con estremi di coordinate [-2 ; -2] e [2 ; 2]. Si può notare la presenza di una cuspide in corrispondenza del segmento nel caso di argomento dell'esponenziale non al quadrato.

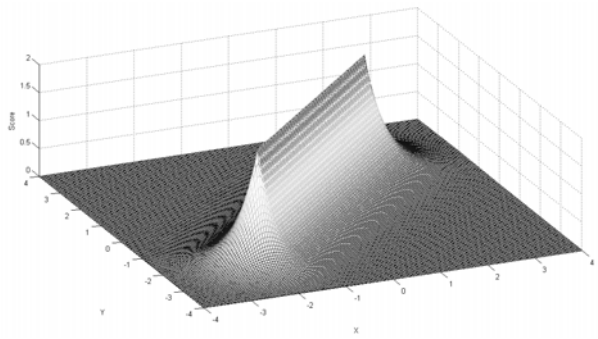


Figura 78: Funzione costo puntuale: argomento dell'esponenziale non al quadrato

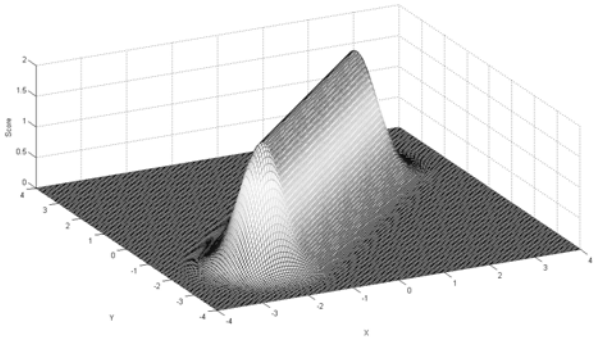


Figura 79: Funzione costo puntuale: argomento dell'esponenziale al quadrato

Altri tentativi di migliorare la convergenza e la velocità dell'algoritmo sono stati fatti cambiando la funzione costo: invece di utilizzare per ogni punto il valore della funzione costo data dalla proiezione più vicina, si é provato con la somma di tutti i segmenti. Le figure Figura 80 e Figura 81 rappresentano il valore della funzione costo per ogni punto dell'area (tra -4 e +4, con step di 0.05, sia in X che in Y), il modello utilizzato é un quadrato centrato nell'origine di lato 2. Si può notare che in presenza di spigoli, il valore é molto maggiore rispetto al centro della linea. Si é visto che questo influisce negativamente sulla convergenza, portando spesso a dei falsi positivi.

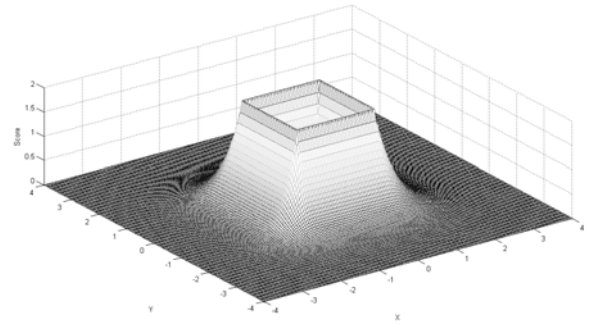


Figura 80: Funzione costo puntuale: esponenziale non al quadrato

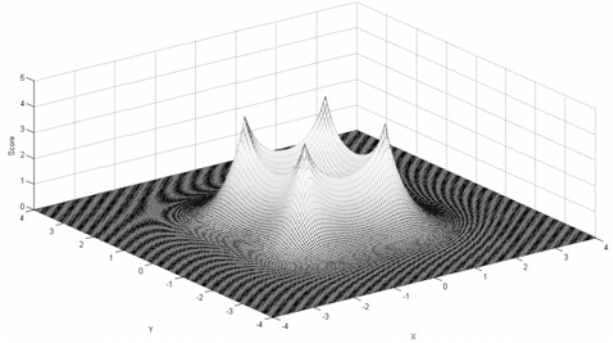


Figura 81: Funzione costo puntuale: esponenziale al quadrato

I valori dei coefficienti A e B servono a velocizzare la ricerca di minimo o a renderla più robusta. Le immagini di Figura 82, Figura 83 e Figura 84 mostrano l'influenza di questi parametri.

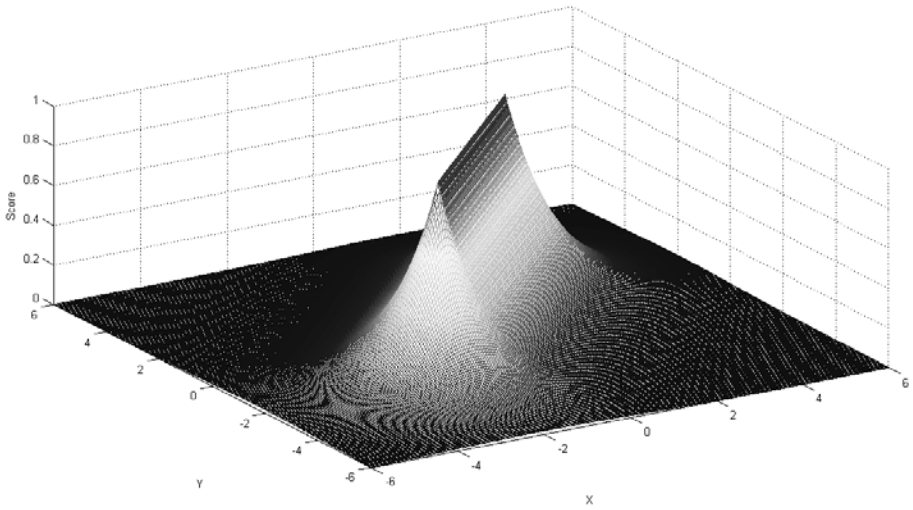


Figura 82: Funzione costo puntuale, sensibilità ai parametri A e B: A = 1 e B = 1

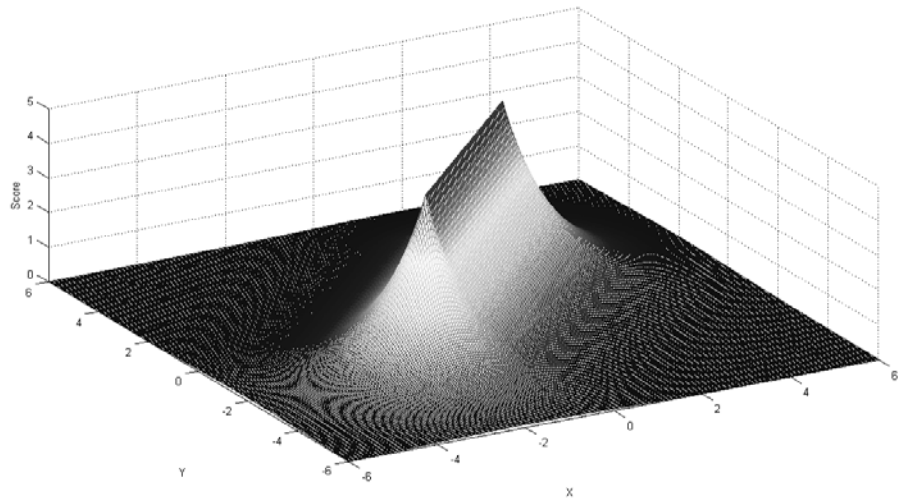


Figura 83: Funzione costo puntuale, sensibilità ai parametri A e B: A = 5 e B = 1

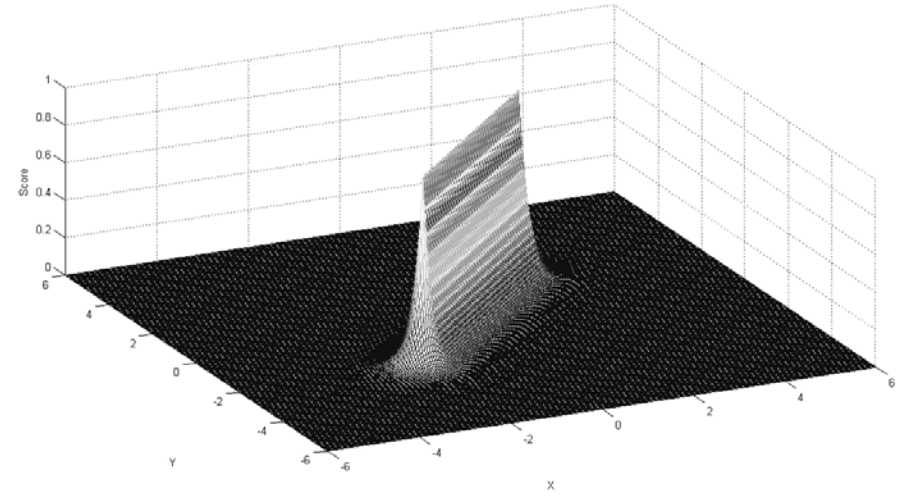


Figura 84: Funzione costo puntuale, sensibilità ai parametri A e B: A = 1 e B = 5

Come ci si aspettava, conoscendo l'andamento di una gaussiana classica, aumentando il coefficiente A la forma della campana non cambia, ma ne aumenta il valore (moltiplicato appunto di un fattore A). Si é visto che questo parametro ha poca influenza, se non per un riscontro visivo nei grafici tipo quello di Figura 74, dove avere dei picchi che si discostano maggiormente può aiutare a capire le differenze tra i vari minimi. Il vero parametro che influenza la ricerca é il coefficiente B, che come si può notare rende la campana più o meno marcata. Aumentandone il valore si ottiene una convergenza più rapida, a patto che il punto iniziale della ricerca si sufficientemente vicino. Al contrario, un valore basso di B permette di avere punti di partenza più “lontani”, a patto di un aumento del numero di iterazioni necessarie a raggiungere la soluzione. Volutamente non si sono dati valori da utilizzare per questi coefficienti, in quanto si é visto che questi dipendono fortemente dal modello (forma, numero di linee...) e dall'accuratezza del sistema di acquisizione. Negli esempi qui esposti, i coefficienti avevano valore A = 40 e B = 50.

4.1.3 Minimizzazione numerica: algoritmi genetici

Gli algoritmi genetici (GA) sono metodi iterativi stocastici che possono essere usati per risolvere problemi di ricerca e ottimizzazione.

Si basano sull’imitazione dei processi genetici degli organismi biologici. Dopo molte generazioni, le popolazioni si evolvono secondo i principi della selezione naturale e della sopravvivenza del migliore, come teorizzato per la prima volta da Charles Darwin nella sua opera "L'origine delle specie". Imitando questi processi, se codificati opportunamente, gli algoritmi genetici sono in grado di evolvere soluzioni per problemi del mondo reale.

I principi di base dei GA sono stati definiti per la prima volta da Holland: essi simulano quei processi che nelle popolazioni naturali sono essenziali per l'evoluzione. Quali processi esattamente siano essenziali o quali giochino un ruolo trascurabile (o nessuno) per l'evoluzione è un problema della ricerca, ma i principi di base sono chiari. In Natura, gli individui di una popolazioni competono uno con l'altro per risorse come cibo, acqua e territorio. Inoltre membri della stessa specie spesso competono per attrarre un compagno. Quegli individui che hanno più successo nella sopravvivenza e nella riproduzione avranno un numero relativamente grande di discendenti. Gli individui che si mostreranno essere meno adatti produrranno poca o forse nessuna prole. Questo significa che i geni degli individui più adattati (fit individuals) saranno trasmessi a un crescente numero di individui in ciascuna delle generazioni successive.

La combinazione delle buone caratteristiche di diversi antenati possono a volte produrre una discendenza molto adattata (superfit), la cui qualità è superiore a quella di ciascun genitore. In questo modo le specie si evolvono e diventano sempre più adattate al loro ambiente.

I GA hanno una diretta analogia con il comportamento della natura. Lavorano con una popolazione di individui, ciascuno dei quali rappresenta una possibile soluzione del problema posto. A ogni individuo è associato un punteggio di adattamento "fitness score" a seconda di quanto sia buona la soluzione al problema. In natura è equivalente a stabilire quanto un individuo riesce a competere per le risorse. Gli individui migliori hanno la possibilità di riprodursi incrociandosi con altri individui della popolazione. Questo produce nuovi individui discendenti che condividono alcune caratteristiche di ciascun genitore. Gli individui meno adattati hanno meno probabilità di riprodursi e quindi si estinguono. Una intera nuova popolazione di possibili soluzioni è così prodotta dalla selezione degli individui migliori della generazione corrente che accoppiandosi tra loro producono un nuovo insieme di individui. Questa nuova generazione contiene una proporzione più alta delle caratteristiche possedute dagli individui buoni della precedente generazione. In questo modo dopo molte generazioni le buone caratteristiche vengono propagate a tutta la popolazione, essendo mischiate e scambiate con altre buone caratteristiche.

Favorendo l'accoppiamento tra gli individui più adatti, vengono esplorate la aree più promettenti dello spazio di ricerca. Se il GA è stato costruito bene, la popolazione converge a una soluzione ottima del problema.

La potenza degli Algoritmi Genetici viene dal fatto che hanno una tecnica robusta e possono essere usati con successo in molti campi, e in problemi che altri metodi difficilmente riescono a risolvere. I GA non garantiscono di trovare una soluzione ottima per un problema, ma generalmente ne trovano una sufficientemente buona in tempi sufficientemente rapidi. Dove esistono tecniche specializzate per risolvere particolari problemi, queste hanno spesso prestazioni migliori dei GA sia in termini di accuratezza che di velocità . Il terreno migliore dei GA sono dunque le aree dove non esistono tecniche specializzate. Dove esistono tecniche che funzionano bene, si possono avere miglioramenti "ibridizzandole" con i GA.

4.1.3.1 Principi di base

Gli algoritmi genetici si basano su una *popolazione* composta da individui, rappresentanti le possibili soluzioni del problema. Ogni *individuo* è composto da *cromosomi* (genotipo, in analogia con la genetica) rappresentati le variabili del problema, ed il suo fitness (fenotipo) viene attribuito inserendo quei valori nella funzione da massimizzare, definita quindi *funzione fitness*.

È un algoritmo iterativo, e ad ogni iterazione vengono svolti i seguenti passi:

- Calcolo fitness della popolazione
- Riproduzione: selezione – riproduzione – mutazione
- Formazione di una nuova popolazione

Il calcolo del fitness consiste semplicemente nell'inserire le variabili che descrivono ogni individuo all'interno della funzione da ottimizzare. In seguito, per creare la nuova popolazione, vengono selezionati ogni volta 2 individui, che tramite la riproduzione per crossover generano la discendenza. Finito il processo, in maniera probabilistica vengono create delle mutazioni sulla nuova popolazione appena formata.

Solitamente, viene richiesto che la funzione fitness restituisca sempre un valore positivo; nella seguente trattazione viene esclusa questa limitazione.

4.1.3.1.1 Da variabili reali a codice binario/Gray

Ogni individuo viene descritto tramite dei cromosomi, che non sono altro che la rappresentazione in binario di ogni singola variabile.

In (Alden, 1991) viene descritto come sia possibile descrivere una variabile reale in questa codifica, passando quindi dal campo dei reali ad uno discreto.

Dati degli estremi A e B, è possibile discretizzare l'intervallo tra di essi utilizzando n bit, tramite la seguente formula:

$$x = A + k\frac{B-A}{2^n}$$

dove k è un intero compreso tra 0 (incluso) e 2^n (escluso).

Ovviamente, maggiore sarà la precisione richiesta, e quindi maggiore il valore di n, tanti più bit saranno necessari per descrivere un numero, pesando però sulle prestazioni dell'algoritmo.

Ad esempio, per descrivere l'intervallo [0 100] con una risoluzione di almeno 0.1, il cromosoma sarebbe lungo 10 bit, e la risoluzione effettiva pari a $\frac{100-0}{2^{10}} = 0.0977$ $\frac{100-0}{2^{10}} = 0,0$. k sarà compreso tra 0 e 1024, e ad esempio per un valore di k pari 5, si avrà x pari a:

$$x = 0 + 5\frac{100-0}{2^{10}} = 0.49$$

E la codifica in binario:

$$5^{dec} = 0000000101^{bin}$$

In questo modo è possibile minimizzare l'occupazione di memoria, in quanto non serve salvare per ogni cromosoma l'intera sequenza di bit, ma solamente l'intero positivo k.

Dal codice binario, successivamente, si può passare al codice Gray tramite la seguente formula:

$$g_i = \begin{cases} b_1 & \text{per } i = 1 \\ b_{i-1} +_2 b_i & i > 1 \end{cases}$$

I termini b_i sono i valori di ogni singolo bit del codice binario, mentre g_i il corrispettivo bit in Gray code. L'operatore $+_2$ rappresenta la somma in modulo 2.

Come si può osservare, la codifica precedente rende necessaria la conoscenza di *boundaries* nella ricerca.

4.1.3.1.2 Selezione

Una volta noto il valore di fitness (f_i) per tutti gli elementi, , se necessario il vettore viene traslato in modo che tutti gli elementi siano positivi, sommando a tutti i valori il minore di essi. In questo modo, però, il termine minore avrà valore di fitness nullo e quindi, come si vedrà tra poco, probabilità di discendenza nulle. Per ovviare a questo problema, si somma 1 al vettore, prima ottenuto.

In seguito, si normalizza il risultato in modo da ottenere somma unitaria, dividendo tutti i valori per la loro somma.

$$p_i = \frac{f_i}{\sum f_i}$$

Questo nuovo vettore p rappresenta la probabilità che un numero venga estratto, nel caso di selezione a *roulette* (tra breve descritta).

Il passo successivo è quello di calcolare la probabilità cumulata.

$$q_i = \frac{p_i}{\sum_{j=1}^{i} p_j}$$

In Tabella 4 vengono mostrati dei valori di esempio.

Tabella 4: esempio di calcolo fitness cumulato

	Fitness		
Individuo	Calcolato	Normalizzato	Cumulato
1	4	0.10	0.10
2	9	0.23	0.33
3	20	0.51	0.85
4	1	0.03	0.87
5	5	0.13	1.00

Una volta calcolati i precedenti valori, si procede alla selezione, che può essere fatta in diversi modi:

Roulette: si estrae un numero R random tra 0 e 1, e si seleziona l'individuo in modo che sia il numero 1 se $R<q_1$ (dove q_i è la frequenza cumulata dell'i-esimo individuo), altrimenti si sceglie l'individuo con indice i se $q_{i-1}<R<q_i$. Nell'esempio precedente, se R=0.86, verrà scelto il 4° individuo. In questo modo, maggiore è il valore di p_i e maggiore sarà la probabilità di essere scelti.

Tournament: si effettua N volte la selezione a roulette, e tra gli individui estratti viene preso il migliore. Una variante è quella di effettuare di nuovo una roulette tra il gruppo ristretto, con la necessità però di dover ricalcolare il fitness cumulato.

Top Percent: si sceglie l'individuo facendo una roulette tra i migliori N.

Best: viene scelto l'individuo con fitness maggiore; in caso di più individui con lo stesso fitness, tra questi la selezione è casuale.

Random

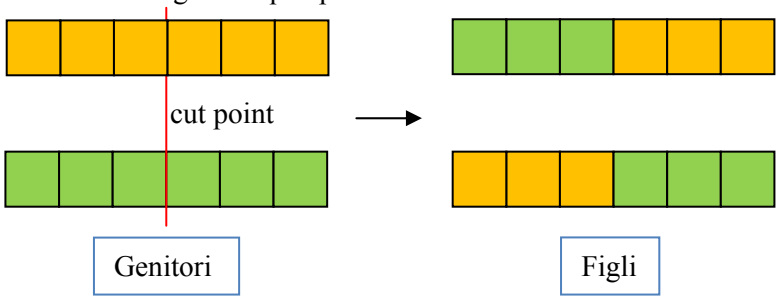
Si sceglie a caso tra tutti gli individui.

4.1.3.1.3 Crossover

Per creare la generazione successiva, si scelgono di volta in volta 2 individui con i metodi sopra descritti. L'operazione di crossover consiste nella creazione di 1 o 2 individui a partire da 2 genitori, "assemblandoli" con parti di entrambi questi ultimi.

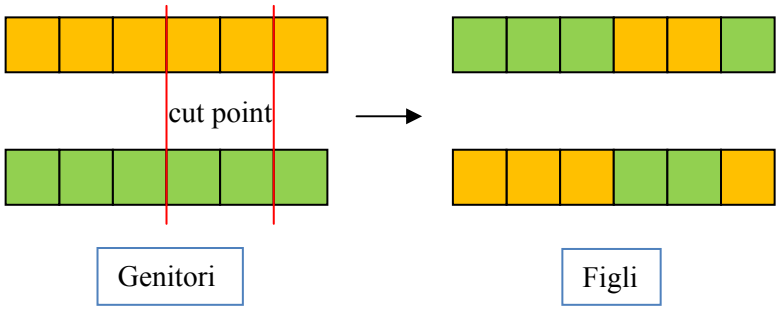
Esistono diverse varianti di questa operazione, e nel seguito verranno illustrate le più classiche:

One Point: viene scelto un punto all'interno dei cromosomi (che ovviamente devono avere la stessa lunghezza), e vengono scambiati i geni in quel punto.



Il punto può essere preventivamente deciso (ad esempio, a metà) oppure totalmente random.

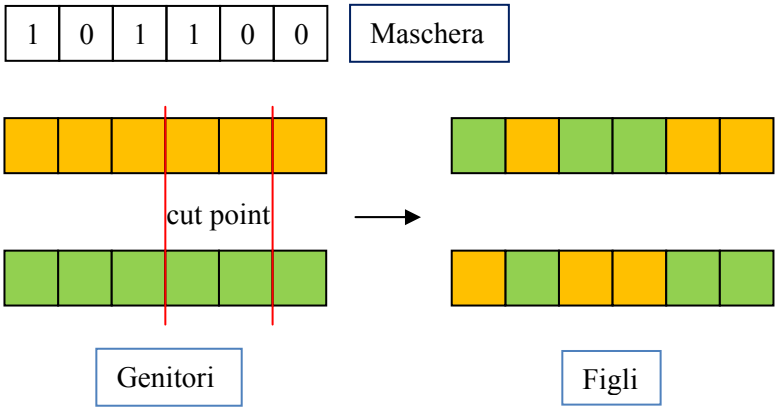
Two Point: come il precedente, ma i punti di taglio sono due al posto di uno solo.



Uniform: questa tecnica è completamente differente dal one-point crossover. Ciascun gene nei figli è creato tramite una copia del corrispondente gene da uno dei due genitori, scelto in accordo a una "maschera di crossover" creata in maniera casuale.

Dove c'è un 1 nella maschera, il gene è copiato dal primo genitore, e dove c'è uno zero, il gene è copiato dal secondo genitore.

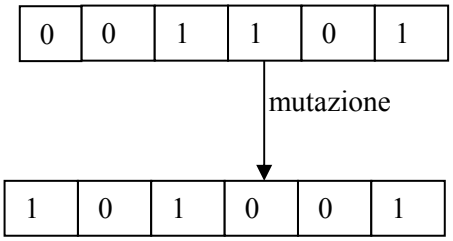
Il processo è ripetuto con i genitori scambiati per produrre un (eventuale) secondo figlio. Una nuova maschera crossover è generata casualmente per ciascuna coppia di genitori. Il figlio, quindi, contiene una mistura di geni provenienti da ciascun genitore.



4.1.3.1.4 Mutazione

Per ogni individuo, viene estratto un numero random tra 0 e 1, e se questo è maggiore di un valore prestabilito, definito "probabilità di mutazione" (ad esempio 0.05), allora avviene la mutazione.

Viene scelto un cromosoma a caso tra le variabili a disposizione, ed in questo viene estratto sempre a random un gene, che viene modificato. Come spiegato precedentemente, un cromosoma è una sequenza di bit (espressione binaria del valore della variabile), e ognuno di questi rappresenta un gene che quindi può valere 0 o 1.



4.1.3.1.5 Mu + Mu

La tecnica "Mu + Mu" (μ+μ) è utile per cercare di non peggiorare l'evoluzione, in modo che la media del fitness non cali. Consiste nell'utilizzare le ultime 2 popolazioni per la riproduzione: per ottenere N individui, si utilizzano 2N genitori, il doppio rispetto al metodo classico. Richiede una quantità doppia di memoria, ma in genere le popolazioni non superano i 50/100 individui e quindi lo spazio necessario per l'immagazzinamento dati è relativamente esiguo, soprattutto nel caso si usi la rappresentazione decimale dei cromosomi invece che quella binaria grazie ala quale ogni individuo è descritto da tanti interi quante sono le variabili in gioco. Il numero di calcoli richiesti sono minimi, considerando che la spesa maggiore è quella per il computo del fitness, e applicando questa tecnica o meno non cambia questo valore, ossia il numero di funzioni da valutare rimane sempre lo stesso.

Una variante del metodo viene definita "Mu + Lambda" (μ+λ), nella quale il numero di genitori provenienti dalla 2° generazione (quella più vecchia) non è pari a quello della 1°.

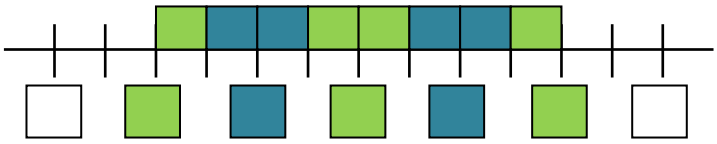
4.1.3.1.6 Numero random intero

Per creare un numero random intero, si ricorre alla funzione *rand* (presente in Matlab e sotto altri nomi in tutti i linguaggi di programmazione), che fornisce in output un numero random compreso tra 0 e 1. Volendo generare un numero intero ad esempio tra 3 e 7, si potrebbe moltiplicare il risultato di *rand* per la grandezza dell'intervallo, ed arrotondare (tramite la funzione *round*) questo valore senza cifre decimali, e cioè seguire la seguente formula:

$$x_r = round[rand * (x_{\max} - x_{\min})]$$

dove x_r è il valore arrotondato, mentre x_{max} e x_{min} sono gli estremi (nell'esempio x_{max}=7 e x_{min}=3).

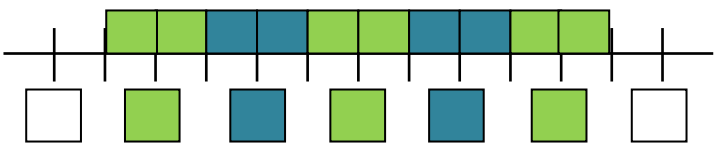
In questo modo, però, i valori estremi avrebbero un probabilità di venire estratti dimezzata rispetto agli altri. La seguente figura mostra questo fatto:



Per ovviare a questo inconveniente, è sufficiente modificare la formula precedente nel seguente modo:

$$x_r = round\{[rand * (x_{\max} - x_{\min} + 1)] - 0.5\}$$

Così, la finestra viene prima aumentata di 1 unità, ed in seguito traslata verso sinistra di 0.5: in pratica, è come se venisse espansa sia a destra che a sinistra di 0.5, come evidenzia la seguente figura:



4.1.3.2 Funzioni test

In (Knoll, et al, 2002), (Sastry, Goldberg, Kendall) vengono mostrate delle funzioni test su cui effettuare le prove degli algoritmi sviluppati. In particolare, alcune hanno la peculiarità di avere molti massimi locali, il che le rende adatte alla ricerca di un metodo robusto che riesca ad evitarli.

Nel seguito verranno descritte analiticamente e rappresentate mediante grafici, nel caso di 2 variabili in input.

4.1.3.2.1 DCS function

Questa funzione, si trova in (Knoll, et al, 2002), è simile ad una calotta sferica corrugata, infatti l'acronimo del suo nome sta per *Deflected Corrugated Spring*; ha un numero *m* arbitrario di variabili, e la sua espressione matematica è la seguente:

$$f(x) = \frac{1}{10}\sum_{i=1}^{m}(x_i - c_i)^2 - \cos\left[\left[\sum_{i=1}^{m}(x_i - c_i)^2\right]^{0.5}\right]$$

Nella sua formulazione, i coefficienti c_i sono i valori del punto di minimo x_{min}, ed inoltre $f(x_{min}) = -1$. In particolare, un punto di minimo locale che è molto "attraente" per la maggior parte degli ottimizzatori è quello per cui $f(x) = -0.84334$. Il coefficiente k nel coseno varia la "rugosità" della superficie: maggiore è il suo valore e più la risultante sarà corrugata (maggiore è la frequenza all'interno del termine coseno).

In Figura 85 viene plottata la funzione (invertita per cercarne il massimo invece che il minimo) nel caso di 2 variabili, e con c_1=1 e c_2=2.

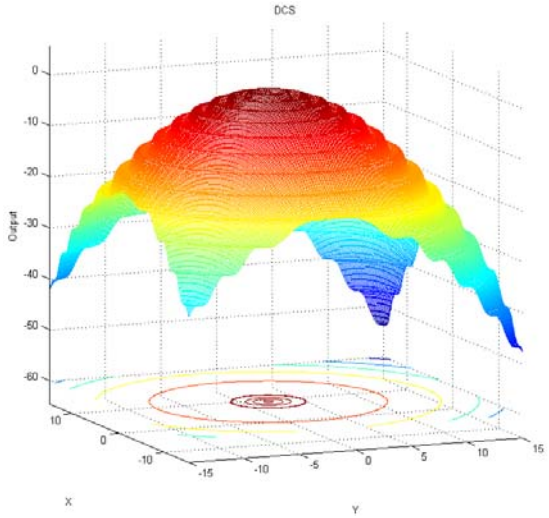


Figura 85: DCS function con m=2

4.1.3.2.2 Shekel function

Anche questa è una funzione con un numero *m* arbitrario di variabili. È descritta in (4) e la sua formulazione è la seguente:

$$f(x) = \sum_{i=1}^{n}\frac{1}{c_i + \sum_{j=1}^{m}(x_i - a_{ji})^2}$$

I termini c_i appartenenti al vettore *C* e i termini a_i appartenenti alla matrice *A* servono a definire la posizione e l'ampiezza degli *n* picchi, definibili dall'utente. Di conseguenza, *C* ha dimensione pari al numero scelto di massimi, mentre *A* ha dimensione *m x n*. In bibliografia, sono sempre stati riscontrati valori di c_i compresi tra 0 e 1, mentre per a_i i valori sono compresi tra -10 e 10.

La tabella seguente mostra i valori utilizzati in Figura 86 per *C* ed *A*.

Tabella 5: valori esempio in Shekel Function per C ed A

C									
0.4387	0.3816	0.7655	0.7952	0.1869	0.4898	0.4456	0.6463	0.7094	0.7547
A									
8.1472	9.0579	1.2699	9.1338	6.3236	0.9754	2.785	5.4688	9.5751	9.6489
1.5761	9.7059	9.5717	4.8538	8.0028	1.4189	4.2176	9.1574	7.9221	9.5949

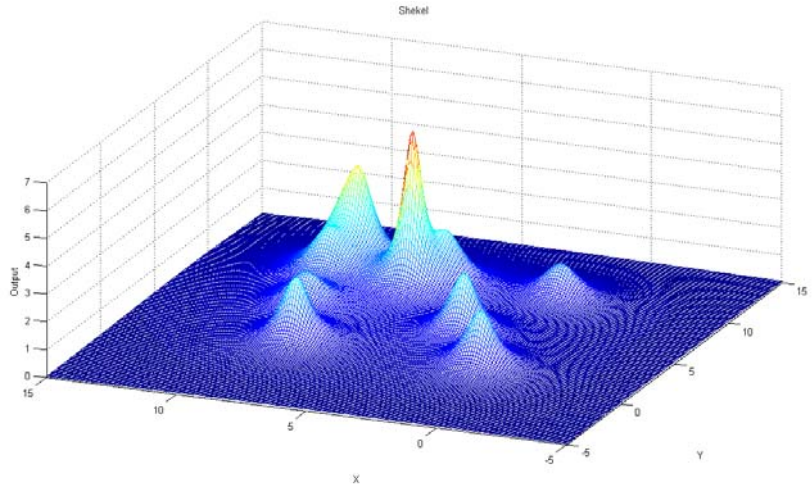


Figura 86: Shekel function con m=2

4.1.4 Robustezza algoritmo

Verranno ora analizzate alcune tecniche di filtraggio dei dati grezzi che sono state provate con alterne fortune per aumentare la robustezza dell'algoritmo di ricerca di oggetti. Il primo passo consiste nel filtrare i dati grezzi (raw data) delle scansioni, per mezzo di un filtro mediano. In seguito si procede con l'estrazione di linee dall'immagine e all'eliminazione di parte dei dati in base a determinati criteri.

4.1.4.1 Media delle scansioni

Come già messo in evidenza nel paragrafo 3.1.2 diverse scansioni vengono raccolte dalla stessa posa e mediate per diminuire il rumore sui dati.

4.1.4.2 Filtro Mediano

Il filtro mediano é un filtro non lineare, utilizzato principalmente nel trattamento di immagini 2D, per eliminare i cosiddetti outliers, ossia punti solitari dovuti ad errori nelle scansioni (ad esempio a causa di riflessioni dei raggi su superfici troppo lucide oppure per i problemi descritti nel paragrafo 3.1.2). Rispetto ad altre tipologie di filtri (passa-basso o passa-banda), permette di ottenere effetti di smoothing o blurring dell'immagine minimi.

Nel caso delle scansioni del LIDAR, il filtro individua una "finestra" centrata attorno a ogni punto,, ossia dato il valore i-esimo si considerano i valori dal i-1-esimo al i+1-esimo nel caso di finestra a 3 valori, oppure dal i-2-esimo al i+2-esimo per una finestra a 5 valori, e così via a seconda della dimensione della finestra che si vuole considerare. Maggiore la dimensione della finestra, e maggiore sarà l'effetto di smoothing ottenuto, quindi le finestre consigliate sono di dimensione 3, 5 o al massimo 7.

Il passo successivo é ordinare i punti del gruppo secondo la distanza, e di questi scegliere il dato mediano. Questo valore andrà a sostituire quello del punto su cui è centrata la finestra.

Come esempio, prendiamo la seguente serie di dati:

5 8 42 3 6 9 8

Come si può notare, il 3o elemento può facilmente essere definito un outliers, in quanto si discosta di molto dalla media dei valori. Il filtro mediano applicato a quel valore con dimensione della finestra pari a 3 si comporta nel seguente modo:
Finestra: 8 42 3
Ordinamento: 3 8 42
Dato scelto: 8
Il risultato finale sarà:

5 8 8 6 6 8 8

In Figura 87 si possono osservare i dati iniziali e finali, dopo l'applicazione del filtro come descritto.

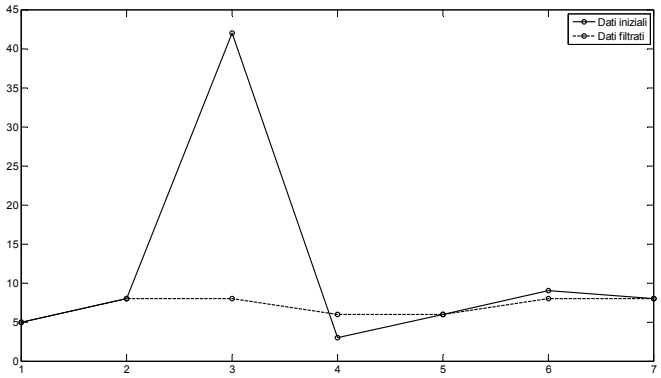


Figura 87: esempio filtro mediano

In Figura 88, Figura 89 e Figura 90 si può osservare il processo di filtraggio applicato a delle scansioni reali, con una finestra di dimensione 7. Si può notare subito come il punto che dista 50 m dall'origine, visibile nella prima immagine (probabilmente causato da una superficie non abbastanza opaca o da una finestra) scompaia nella terza immagine, insieme a molti altri punti.

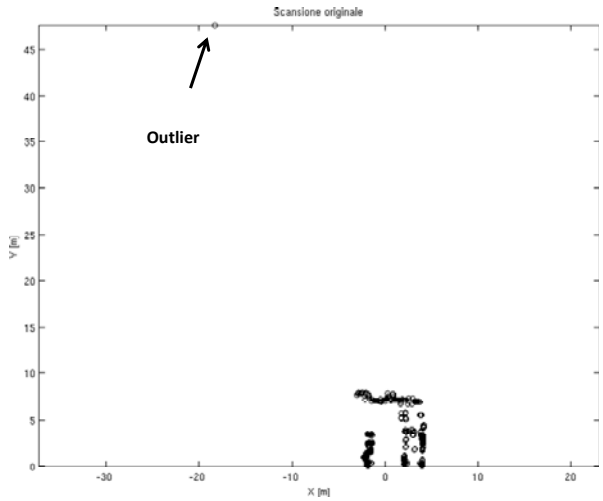


Figura 88: Filtro mediano con finestra di dimensione 7: scansione di base - da notare l'outlier in alto a sinistra

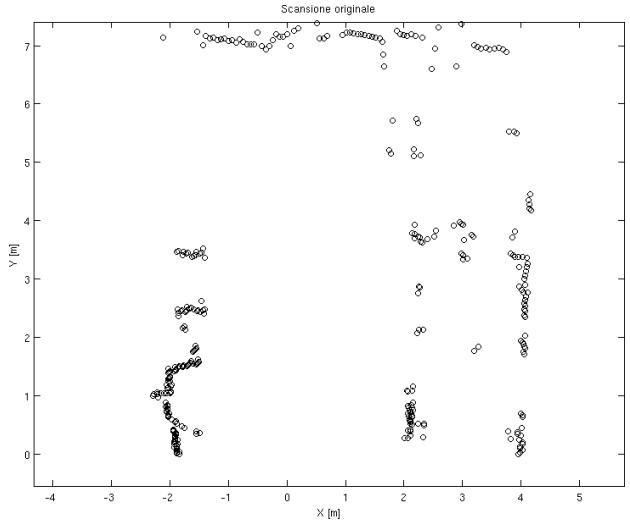


Figura 89: Filtro mediano con finestra di dimensione 7: scansione base - zoom ambiente

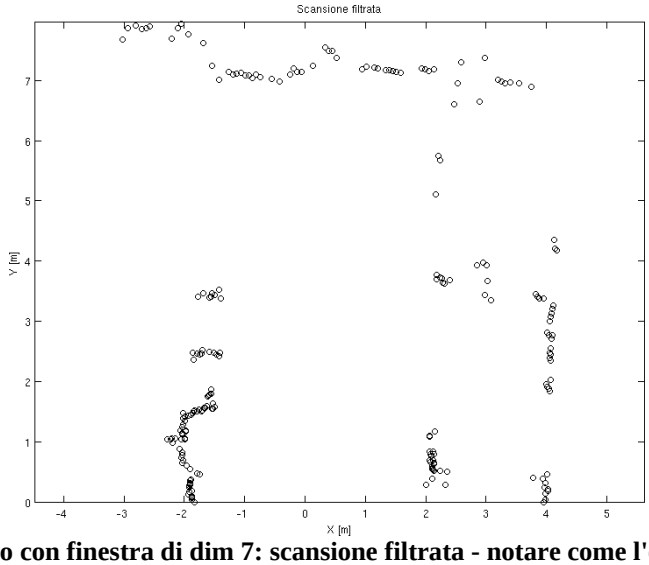


Figura 90: Filtro mediano con finestra di dim 7: scansione filtrata - notare come l'outlier iniziale sia sparito

4.1.4.3 Cluster

Il processo di clusterizzazione consiste nel suddividere in gruppi i punti delle scansioni. In questo caso si è proceduto a clusterizzare i dati in base alla differenza tra due valori di range consecutivi. Viene creato un vettore delle differenze, in cui l'elemento i-esimo rappresenta il modulo della differenza tra il valore i+1 e l'elemento i, distanze, del vettore contenente la singola scansione.

Quando un elemento del vettore differenze supera una certa soglia viene creato un nuovo cluster: il gruppo precedente termina con l'elemento i e quello nuovo inizia con l'elemento i+1-esimo.

Con questo metodo, dunque, si cerca di individuare gruppi di punti che rappresentino oggetti singoli.

4.1.4.4 Line extraction

L'estrazione di linee é il processo attraverso il quale dai punti in coordinate cartesiane si passa ad una visione geometrica dell'ambiente, descrivendolo quindi tramite linee e segmenti. Ogni cluster individuato come spiegato in precedenza, viene processato per estrarre un segmento.

Si é seguito un approccio simile a split and merge descritto in (Nguyen, et al): dato un gruppo di punti si calcola la linea che passa per il primo e l'ultimo, si individua il punto più distante da questa, e se la distanza é maggiore di una soglia (es. 5 cm), il cluster iniziale viene diviso in 2 gruppi e si ripete il procedimento per entrambi. In Figura 91 viene schematizzato il procedimento.

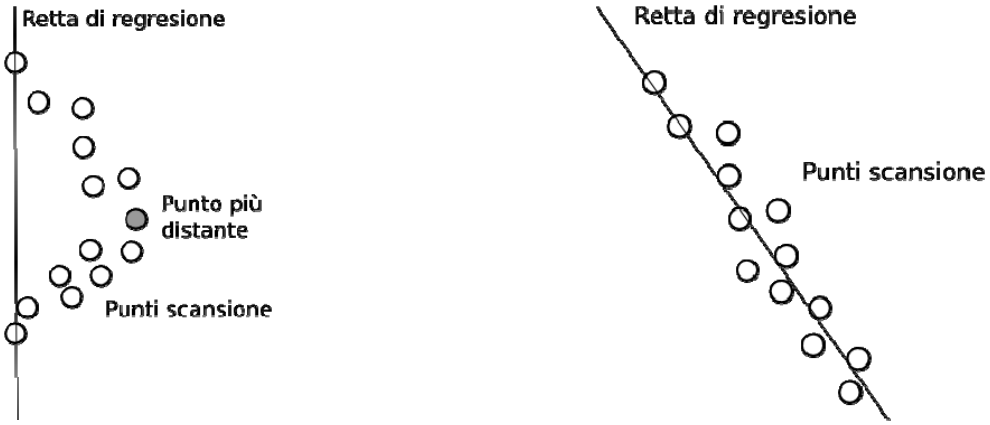
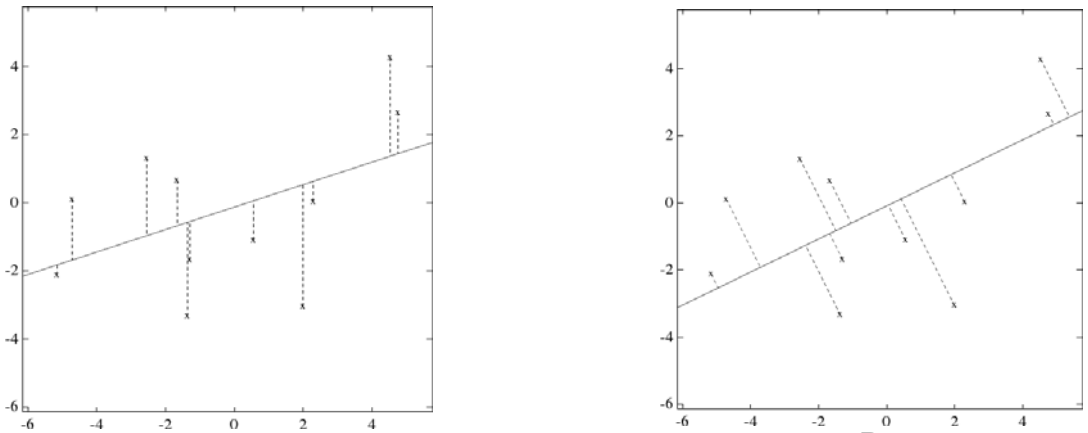


Figura 91: Split and merge. SX:la linea non fitta bene il cluster, che deve quindi essere diviso. DX: la linea trovata fitta il cluster

Una volta definiti i gruppi, per ognuno si ricalcola la linea, minimizzando la somma delle distanze dei punti da questa. Per un migliore fitting dei punti, viene utilizzato l'algoritmo Total Least Squares come descritto in (de Groen. 1996) e (Krystek, Anton, 2007). Il Total Least Squares minimizza la lunghezza del segmento perpendicolare alla retta che passa per il punto come mostrato in Figura 92 B, piuttosto che la distanza lungo Y come nella minimizzazione classica dei minimi quadrati (Least squares) come in Figura 92 A.



A **B**

Figura 92: Fitting lineare. A = metodo least square: minimizzazione della distanza lungo Y. B = metodo total least square: minimizzazione della distanza perpendicolare

Una volta calcolata la retta che fitta i punti, gli estremi del segmento sono calcolati come la proiezione dei punti estremi del gruppo, come schematizzato in Figura 93.

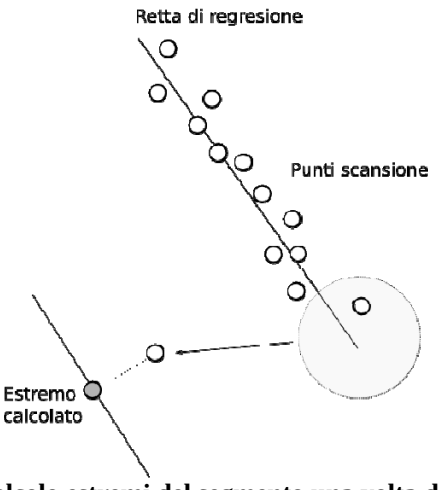


Figura 93: Calcolo estremi del segmento una volta definita una linea

Oltre a questo metodo, sono stati provati altri algoritmi per la line extraction, quali la trasformata di Hough (vedi (Duda, Hart. 1972)), ma hanno fornito risultati peggiori sia qualitativamente che in termini computazionali.

4.1.4.5 *Analisi risultati pre-processing*

Il filtro mediano permette di eliminare outliers che possono dare problemi nel matching tra le scansioni, creando dei minimi locali che rallentano la convergenza della minimizzazione. Questi punti isolati hanno origine soprattutto quando i raggi di scansione del LIDAR incontrano spigoli che si aprono nel vuoto, come descritto in 3.1.2, o a causa di finestre e superfici poco opache.

A tutte le scansioni, prima di essere processate, può essere applicato questo filtro, con finestra di dimensione 3. Si é visto, però, che non é opportuno utilizzarlo nella ricerca di oggetti, in quanto potrebbe essere eliminati punti isolati che però fanno parte dell'oggetto. Di conseguenza il suo impiego é limitato all'algoritmo ICP descritto nel capitolo 3, con una finestra di dimensione 3.

L'estrazione di rette viene utilizzata per eliminare dalle scansioni quei punti che sicuramente non appartengono al modello cercato: si supponga ad esempio di trovare, dato un gruppo di punti, una retta di lunghezza stimata 2 m, se nel modello il segmento più lungo é di 20 cm, di sicuro i punti appartenenti a quella linea non faranno parte del modello e quindi possono essere esclusi. Si é osservato, tuttavia, che negli intorni dell'oggetto da cercare non sono praticamente mai presenti linee che rispecchiano questo andamento, e quindi questo tipo di pre-processamento risulta quasi sempre inutilizzato.

Un'altra tecnica possibile é quella di escludere gruppi di punti poco numerosi: se ad esempio dal processo di clusterizzazione si trovano insiemi formati da 1 solo punto, o anche 2, si potrebbero eliminare perché possono confondere la ricerca. Con questo metodo si raggiungono risultati pessimi poiché a causa della bassa risoluzione del laser impiegato (0.5°), si degrada eccessivamente l'immagine: un oggetto di piccole dimensioni da cercare, come possono essere i blocchetti del pallet, verrà scansionato ottenendo pochi punti, e quindi verrà eliminato tutto o in buona parte.

Un esempio di preprocessing delle scansioni è schematizzato in Figura 94.

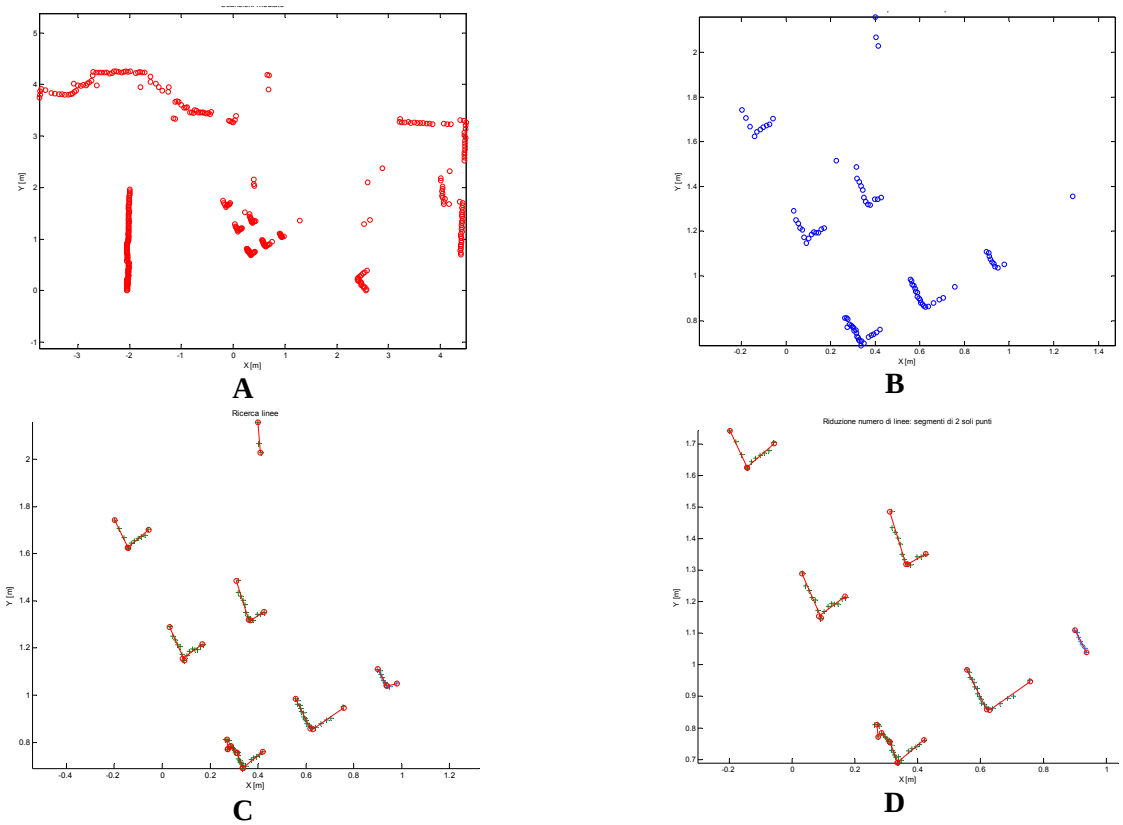


Figura 94: esempio di preprocessing in cui a partire dall'immagine originale dell'ambiente, A, si procede clusterizzando gli elementi in base alla distanza, B, poi con il line fitting, C, ed infine eventualmente eliminando linee composte da pochi punti, D.

4.2 Validazione algoritmo in simulazione

È stato condotto un test di validazione dell'algoritmo di localizzazione in simulazione. La posa dell'oggetto è stata fatta variare in distanza da 2 a 5 metri e in assetto da 0° a 180° rispetto al sensore LIDAR, in modo da rappresentare il più ampio possibile spettro di situazioni reali. Sono state registrate i successi e i fallimenti dell'algoritmo nell'identificare l'oggetto e nello stimarne la posa.

È stato testato inizialmente l'algoritmo utilizzando la funzione di minimizzazione Patternsearch di Matlab, descritta in 3.2.1.1. Si è verificato un fallimento nella convergenza nel 5% delle prove, nei casi in cui la configurazione presentasse particolari simmetrie, con l'assetto del pallet vicino a 0°, 90° e 180° rispetto all'asse perpendicolare al sensore laser. Nel seguito in Figura 95 e Figura 96 vengono presentati alcuni risultati ottenuti con l'algoritmo Patternsearch. Si può notare come vi siano casi in cui l'algoritmo non converga alla posa corretta.

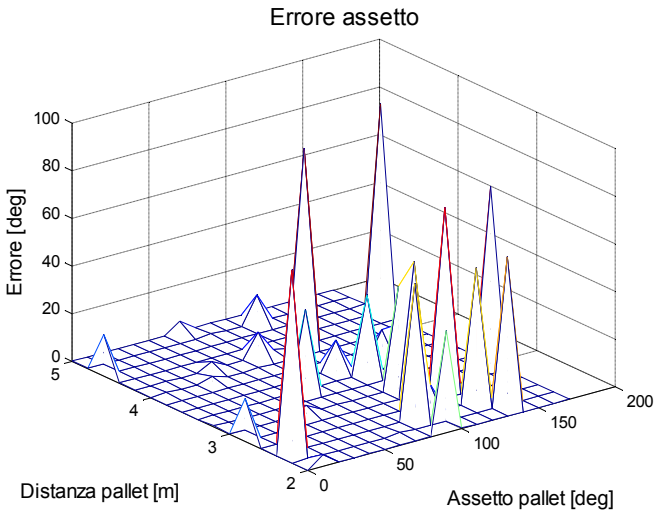


Figura 95: errore su stima assetto, algoritmo Patternsearch

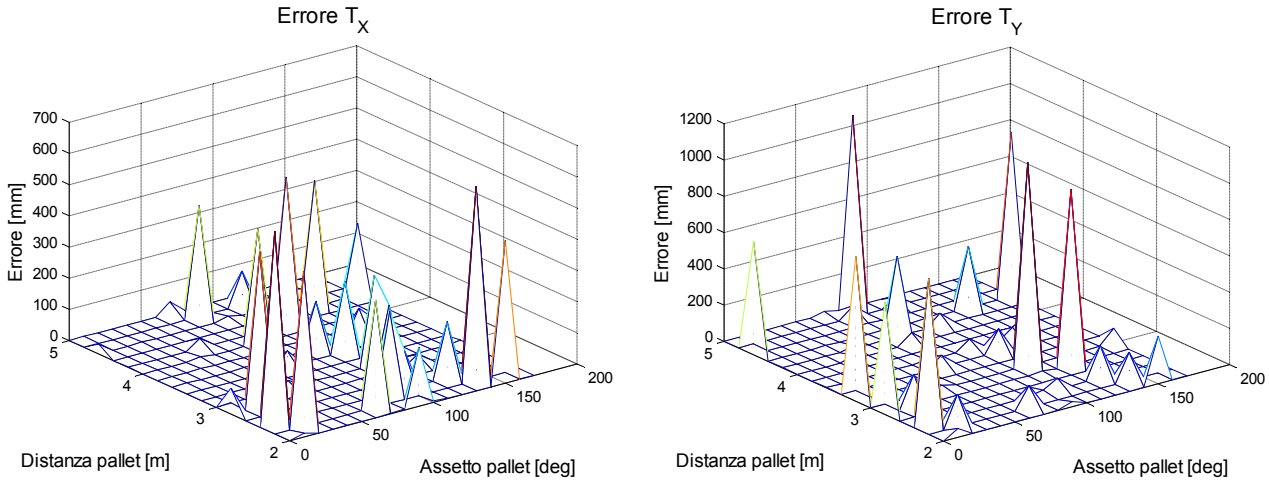


Figura 96: errore su stima coordinata X (sx) e coordinata Y (dx), algoritmo Patternsearch

Sono stati poi testati gli algoritmi genetici descritti in 4.1.3. . In questo caso è stata riscontrata la convergenza nel 99% dei casi. Nel seguito da Figura 97 a Figura 99 vengono presentati alcuni risultati ottenuti. Si può notare come la convergenza sia molto più alta rispetto all'algoritmo Patternsearch.

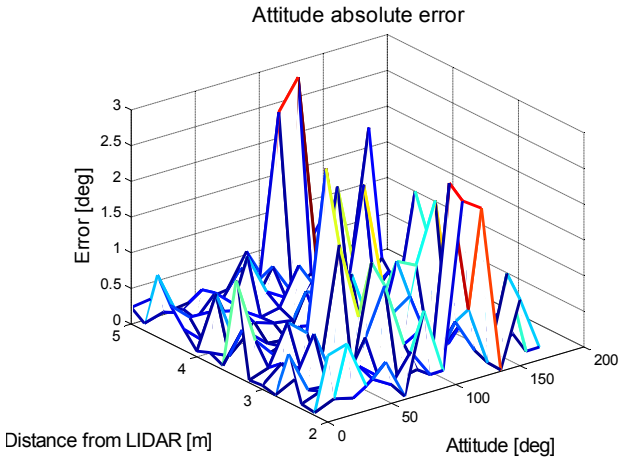


Figura 97: errore su stima assetto, algoritmi genetici

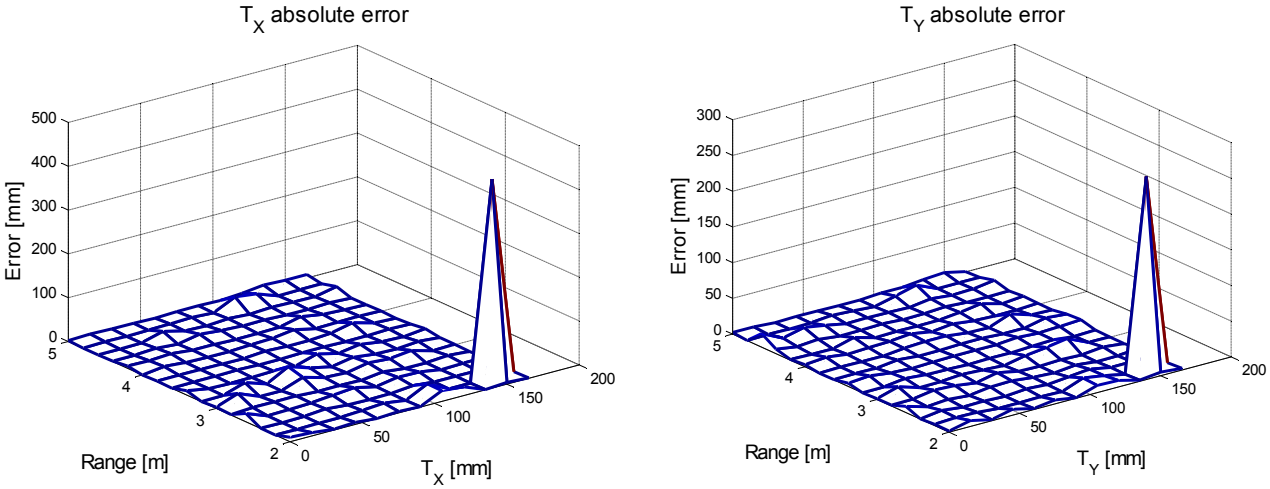


Figura 98: errore su stima coordinata X (sx) e coordinata Y (dx), algoritmi genetici

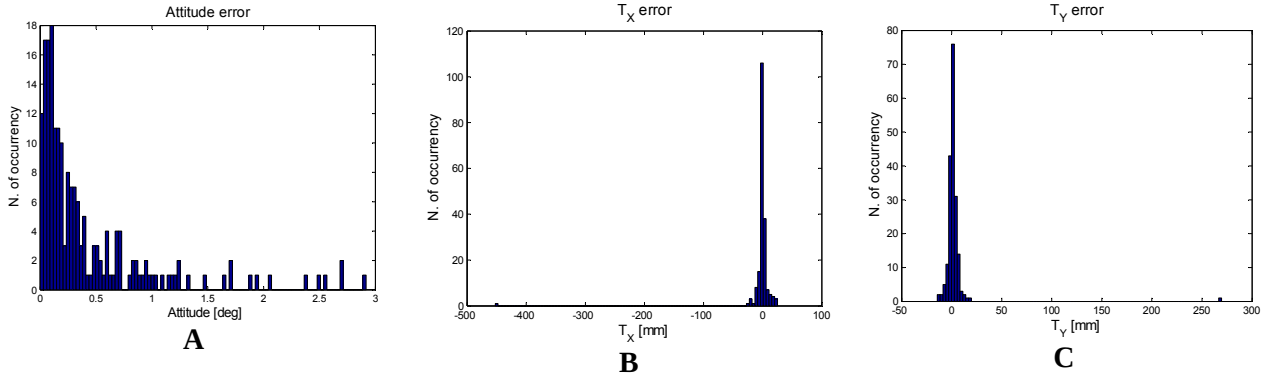


Figura 99: istogramma degli errori su stima assetto (A), su stima coordinata X (B) e coordinata Y (C), algoritmi genetici

Di questi risultati, è stata calcolata media e deviazione standard dopo aver applicato il criterio di Chauvenet per eliminare i casi di non convergenza. I risultati sono riassunti in Tabella 6.

Tabella 6: media e varianza errori, minimizzazione con algoritmi genetici

Attitude [deg]	0.4 +- 0.4
Trasl X [mm]	4.0 +- 5.4
Trasl Y [mm]	2.9 +- 3.1

4.3 Analisi incertezza

É stata condotta una analisi per stimare l'incertezza dell'algoritmo, utilizzando lo stesso setup spiegato in 3.3. La taratura é stata eseguita ponendo il laser su una guida graduata, nella quale era possibile una traslazione ed una rotazione. É stato posizionato un pallet secondo una certa posa rispetto alla guida, ed é stato acquisito l'ambiente spostando il laser. In Figura 43 viene mostrato il sistema utilizzato per la taratura: il laser, il cui sistema di riferimento (sdr) viene definito dalla lettera L, può essere mosso per una qualche distanza d solo lungo la direzione Y del sdr O (sistema world, o assoluto), indicata dalla linea tratteggiata, e può ruotare di un angolo θ rispetto al punto L; il pallet ha il suo sdr chiamato P. In ogni acquisizione, i parametri che cambiano sono d e θ, ossia la posa del sdr L, ma quelli misurati dall'algoritmo sono relativi al sdr P.

Dal momento che la posizione del pallet rispetto al sistema di misura non è nota con precisione per l'incertezza dell'oggetto reale rispetto al modello e l'impossibilità di fissare un sistema di riferimento verificabile, si é scelto di prendere come riferimento una scansione, e di eseguire una taratura differenziale: é nota la differenza di posa tra le scansioni data dalla diversa posizione e assetto del laser misurati sulla slitta graduata, quindi il sistema di riferimento utilizzato é quello relativo allo strumento di misura, e si confrontano i risultati dell'algoritmo con questi valori.

In Figura 100 si puó osservare l'ambiente reale di lavoro; il pallet é rialzato rispetto al pavimento per riuscire a farlo rilevare dal laser. Attorno sono stati messi dei tubi riflettenti o dei cartoni per aumentare la difficoltà nella convergenza dell'algoritmo, allo scopo di testare la robustezza in presenza di disturbi (altri oggetti con cui confondersi) nelle vicinanze. Sono state eseguite delle prove sia con oggetti vicini che senza.

Figura 100: apparato sperimentale in laboratorio

Nella prima immagine di Figura 100 si può notare una delle configurazioni del pallet rispetto al laser più problematiche in quanto a causa della simmetria dell'oggetto vi è una notevole ambiguità nella posa e pochi punti rilevabili (per la geometria proiettiva solo i piedini frontali sono rilevabili dai raggi di scansione). Sono state eseguite numerose prove sia con il pallet in questa configurazione che ruotato di circa 45°. In Figura 101 sono mostrate alcune scansioni acquisite, con il pallet orientato inclinato e non. Il cerchio indica la zona in cui é presente il pallet. Si può osservare in Figura 101 A come l'oggetto sia descritto da pochi punti, e non risulti ben definito come in Figura 101 B.

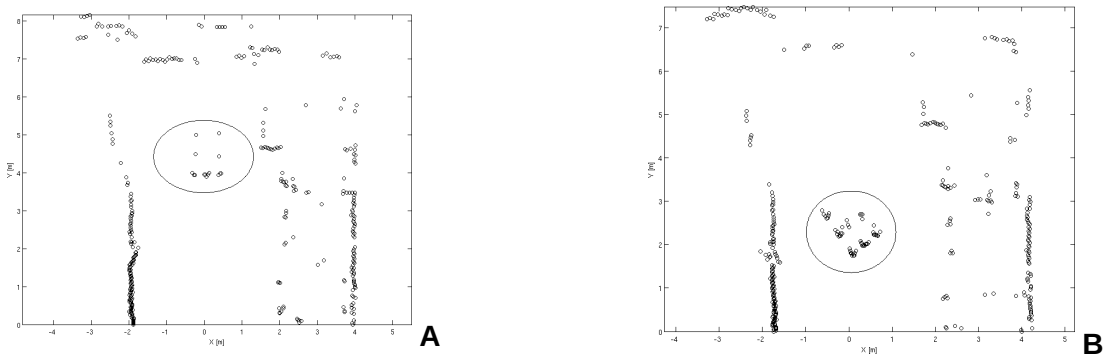


Figura 101: Scansione dell'ambiente. A: pallet frontale, senza disturbi. B: pallet inclinato, senza disturbi

In Figura 102 sono mostrate scansioni simili in cui però sono stati introdotti dei disturbi, ossia degli oggetti nelle vicinanze del pallet. Anche in questo caso é stata evidenziata con un cerchio la zona in cui si trova il pallet, con i tubi o i cartoni posti su entrambi i lati.

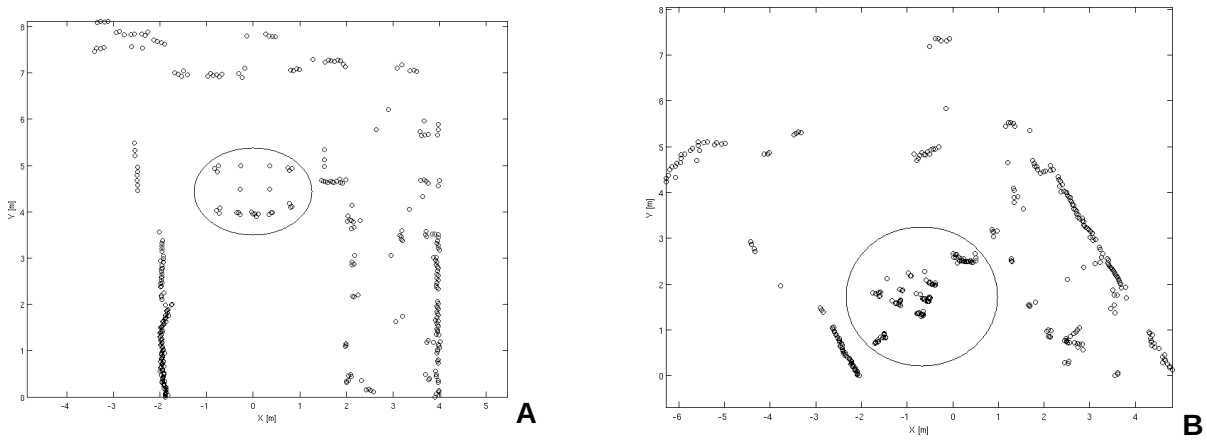


Figura 102: Scansione dell'ambiente. A: pallet frontale, con disturbi. B: pallet inclinato, con disturbi

Oltre che 2 orientazioni del pallet rispetto alla guida, sono state testate 2 distanze, ponendo l'obiettivo della ricerca a 2 m e 4 m circa. É bene ricordare che l'assenza di precisione, come sopra descritto, non é influente ai fini della taratura in quanto le misure sono state eseguite in differenziale, ossia riferendosi ad una scansione base in cui il sdr del laser coincide con quello assoluto della guida.

In Tabella 7 vengono riassunte le prove effettuate con i parametri di posa del laser, secondo i simboli descritti in Figura 43. In totale sono state eseguite 88 misure.

Tabella 7: prove effettuate

d [cm]	Θ [deg]
0	0
0	-10
0	-30
0	10
0	30
10	0
10	-10
10	-30
30	0
30	-10
30	-30

I risultati sono stati ottenuti mediando 20 scansioni per ogni prova. La media é stata eseguita con il metodo descritto in 3.1.2, in modo da ridurre l'incertezza sulla misura circa del 22%. Dal grafico di Figura 103 si può vedere come aumentando ulteriormente il numero di campioni da mediare, cioè il numero di scansioni utilizzate, non si ottenga un miglioramento elevato, in quanto la curva ha una pendenza molto lieve.

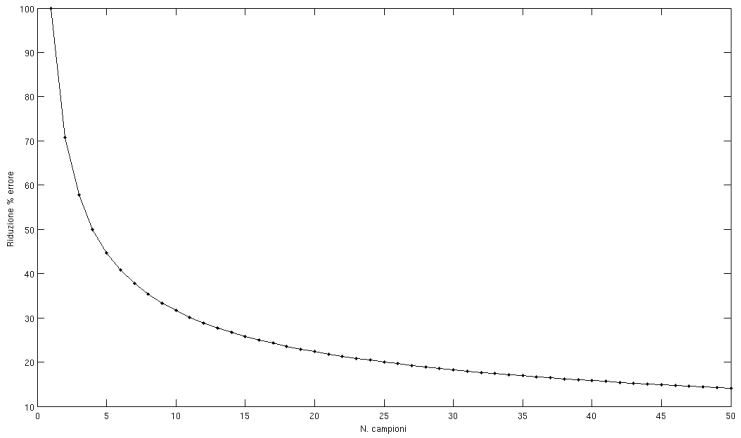


Figura 103: Riduzione dell'incertezza in funzione del numero di campioni mediati

Utilizzando una singola scansione, l'elevata incertezza dello strumento a volte non permette di ottenere risultati apprezzabili: spesso l'algoritmo non rileva oggetti nella scansione. Per questo motivo si é scelto di fare tutte le analisi mediando un certo numero di scansioni.

In output l'algoritmo fornisce un vettore di 3 elementi, contenente la posa (rotazione, coordinate X e Y) del pallet in coordinate laser, denominate θ_m, X_m e Y_m. Da queste, note θ_l, X_l e Y_l, ossia le coordinate del laser nel sistema di riferimento assoluto (date dalla Tabella 7) si trasformato le coordinate del pallet misurate in sdr laser a sdr guida (θ_g, X_g e Y_g), tramite le seguenti formule:

$$\theta_g = \theta_m - \theta_l$$

$$\begin{Bmatrix} X_g \\ Y_g \end{Bmatrix} = \begin{bmatrix} \cos(\theta_l) & \sin(\theta_l) \\ -\sin(\theta_l) & \cos(\theta_l) \end{bmatrix} \begin{Bmatrix} X_m \\ Y_m \end{Bmatrix} + \begin{Bmatrix} X_l \\ Y_l \end{Bmatrix}$$

dove:

- $X_l = 0$
- $Y_l = d$

Sono state calcolate le differenze tra il valore di riferimento della posa del laser, misurato sulla slitta, e il valore calcolato con l'algoritmo. I risultati sono riassunti nelle Tabella 8 e Tabella 9 dove:

- A = pallet a 4 m, isolato
- B = pallet a 4 m, oggetti nelle vicinanze
- C = pallet a 2 m, isolato
- D = pallet a 2 m, oggetti nelle vicinanze

Tabella 8: Differenze rispetto al valore di riferimento per analisi con pallet di fronte

	Scarto medio			Scarto quadratico medio		
	Θ [deg]	X [mm]	Y [mm]	Θ [deg]	X [mm]	Y [mm]
A	1.2	19	6	0.9	10	5
B	0.5	10	6	0.7	7	4
C	0.6	11	9	0.8	6	3
D	1.3	14	13	0.9	12	5
Media	0.9	13	8	0.8	8	4

Tabella 9: Differenze rispetto al valore di riferimento medi per analisi con pallet ruotato

	Scarto medio			Scarto quadratico medio		
	Θ [deg]	X [mm]	Y [mm]	Θ [deg]	X [mm]	Y [mm]
A	0.4	26.1	6	0.4	14	3.7
C	0.2	4.4	7.6	0.1	4.1	7.9
Media	0.31	15.22	6.76	0.25	9.07	5.77

Si puó notare come lo scarto angolare si attesti nell'ordine dei decimi di grado, nel caso di traslazione lungo X nell'ordine della decina di millimetri mentre poco meno lungo la Y. Non ci sono differenze sostanziali tra la presenza di oggetti intorno o meno, e neppure considerando l'orientazione del pallet rispetto alla guida. In questo caso, però, si é notata la facilità dell'algoritmo a convergere o meno: il numero di iterazioni é notevolmente inferiore quando il pallet e messo di lato (70% in meno); lo stesso vale quando il pallet non ha disturbi intorno (50% in meno) e quando é a 2 m (20% in meno). Lo stesso dicasi per i falsi negativi, praticamente assenti nei casi di pallet di lato, ma presenti nel 30% dei casi se orientato di fronte senza disturbi; in caso di presenza di disturbi, la percentuale é salita all'80%.

I risultati sopra riportati sono stati utilizzati anche per ricavare la matrice di covarianza (3x3) delle misure, che può essere utilizzata in un algoritmo di sensor fusion (analogamente a quanto spiegato nel capitolo 2). Utilizzando Matlab, e fornendo in input una matrice le cui righe rappresentano le varie prove, e

le colonne le tre variabili (assetto e coordinate X e Y), il risultato é riportato in Tabella 7. Come si può notare, la matrice non é diagonale, e quindi i vari scarti sono accoppiati tra loro. In particolare si può notare come le variazioni di assetto siano molto più accoppiate alla coordinata x (valore di mezzo della Tabella 10).

Tabella 10: Matrice di covarianza ricerca oggetti

0.58	2.43	1.02
2.43	125.47	1.33
1.02	1.33	29.1

Capitolo 5

Pianificazione e controllo della traiettoria

In relazione alle caratteristiche dei sensori e al tipo di misura che si intende effettuare, quale quella di un veicolo in movimento, è importante che non vi siano slittamenti delle ruote o movimenti bruschi di curvatura poiché in questo modo si deteriorerebbe la misura proveniente da vari sensori (encoder, giroscopi, laser a triangolazione). Nel seguito verranno descritti un algoritmo di pianificazione della traiettoria e la Reactive Simulation, un algoritmo real time di aggiramento ostacoli.

5.1 Pianificazione dinamica della traiettoria

La pianificazione e il controllo della traiettoria vengono gestiti dall'interazione di diversi moduli:

- il Path Control che fornisce i segnali di comando ai driver
- il Planner che esplora la scansione attuale fornita dal laser scanner (mappa attuale dell'ambiente) e che imposta i target locali da raggiungere
- il Graph Manager che memorizza i target locali, sia già raggiunti che da raggiungere
- la Reactive Simulation che computa una predizione della traiettoria per l'aggiramento ostacoli
- la Sentinel che monitora le scansioni del laser per rilevare eventuali ostacoli e permettere un moto in sicurezza.

Il controllo della traiettoria è effettuato dal modulo Path Control secondo il seguente metodo (si veda la Figura 104): il centro del veicolo è proiettato sul segmento *P1-P2* (dove *P1* è il target appena raggiunto e *P2* è il nuovo obiettivo pianificato), ottenendo *Pp*; il punto *Pi* è calcolato con la formula:

$$Pi = Pp + d_0 \cdot e_{12} \qquad (5.1)$$

dove d_0 è la distanza tra *Pp* e *Pi*; e_{12} è il versore tra i due estremi del segmento.

La velocità angolare di riferimento del veicolo è calcolata con la formula:

$$\omega = k\theta$$

dove k è un fattore proporzionale tarato sperimentalmente e θ è la differenza angolare tra l'assetto del veicolo e il segmento *Pr-Pi*

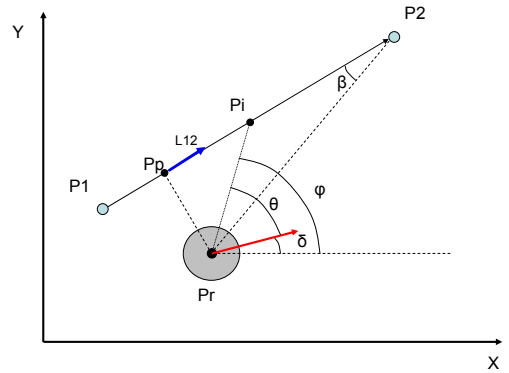


Figura 104: schema di controllo della traiettoria

Il modulo che si occupa della pianificazione della traiettoria, il Planner, imposta dinamicamente dei target locali per raggiungere una certa posizione finale, senza informazioni a priori sulla conformazione dell'ambiente in cui opera il sistema. Le uniche informazioni a disposizione sono la posa iniziale del veicolo e la posizione dell'obiettivo da raggiungere. Il Planner cerca spazi liberi e porte, le cosiddette "aperture",

sceglie un target locale (il punto 1 in Figura 105), lo fornisce al Path Control e il veicolo si muove verso di esso.

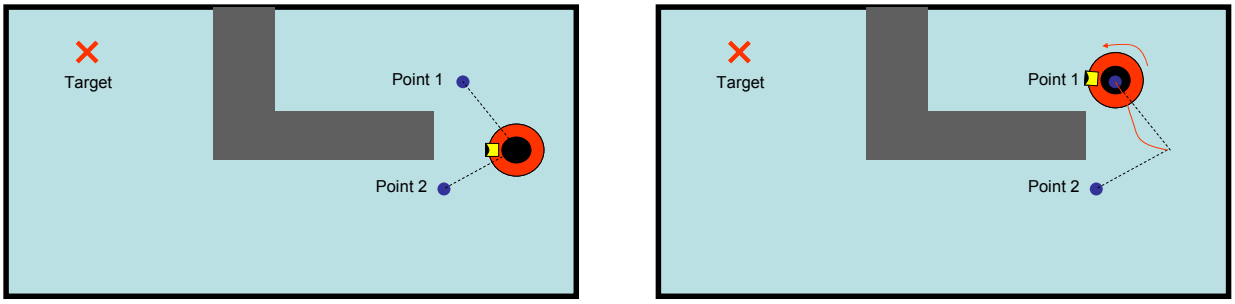


Figura 105: esempio di pianificazione dinamica della traiettoria. Il Planner sceglie un punto (Point 1) vicino ad un'apertura

Dinamicamente il Planner rappresenta l'ambiente visitato basandosi su un grafo, in cui memorizza tutte le aperture. Questo schematismo permette di risolvere il problema dei dead-ends senza risultare troppo oneroso in termini di memoria e richieste computazionali (come potrebbe essere effettuare una mappatura completa dell'ambiente): quando non vengono trovate aperture, il veicolo compie una rotazione di 180°, compie una nuova esplorazione dell'ambiente e se le attuali aperture rilevate sono già memorizzate nel grafo, il Planner realizza di trovarsi probabilmente in un percorso chiuso (il Point 1 in Figura 105). A questo punto il nuovo percorso viene pianificato a partire dalle aperture memorizzate nel grafo: il veicolo si muove verso un obiettivo memorizzato ma non ancora visitato (il punto 2' in Figura 106), selezionato con l'algoritmo di ricerca chiamato A* search. Si tratta di un algoritmo comunemente utilizzato per l'esplorazione dei grafi basato su una funzione euristica che permette di effettuare una scelta localmente ottima (J. Chestnutt et al. 2003) (A. Stentz, 1994).

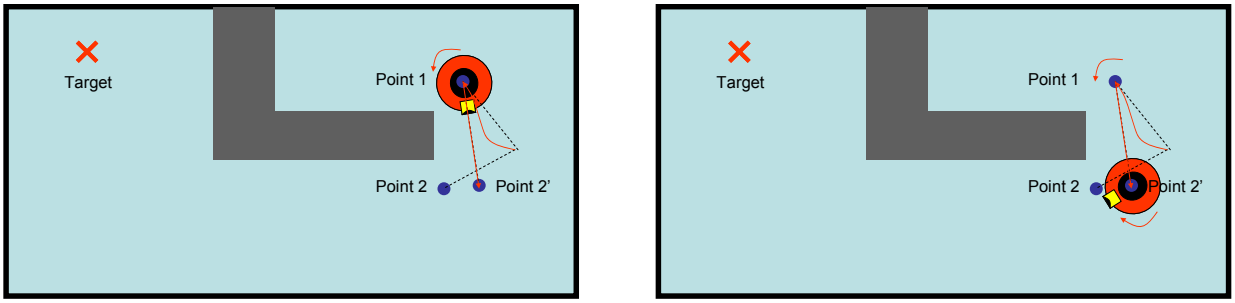


Figura 106: è possibile risolvere il problema dei dead-ends basandosi sulla memorizzazione delle aperture in un grafo

A questo punto, il veicolo continua l'esplorazione dell'ambiente fino al raggiungimento del target finale (punto 3 in Figura 107).

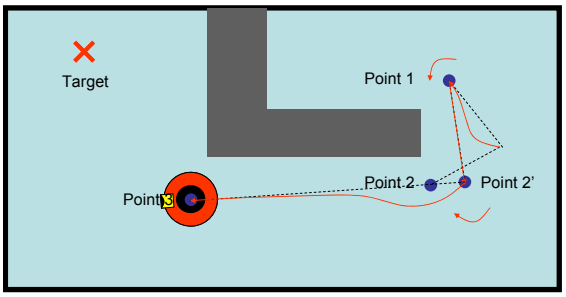


Figura 107: il veicolo continua l'esplorazione dell'ambiente

5.2 Aggiramento degli ostacoli

L'aggiramento degli ostacoli imprevisti o in movimento è un aspetto fondamentale per la navigazione autonoma sia in ambienti strutturati che semi o non strutturati. Il problema non riguarda solo possibili danni fisici al veicolo, all'ambiente o peggio ancora a persone che si trovino nelle vicinanze, ma è legato anche a una corretta stima della posa. Infatti sensori come encoder e piattaforme inerziali in seguito a un urto fornirebbero una posa irreversibilmente compromessa, a meno di fondere poi le misure coi dati forniti da un sensore riferito all'ambiente, ma questo potrebbe comunque non essere sufficiente a ripristinare una stima accurata della posa.

Solitamente il task dell'aggiramento ostacoli si considera distinto dalla pianificazione della traiettoria o path planning, anche se in un ambiente semi-strutturato, in cui cioè il veicolo non dispone di informazioni riguardo la morfologia dell'ambiente in cui si muove, i due aspetti si possono ritenere molto simili. In molti sistemi infatti la traiettoria viene dinamicamente ripianificata con un certo periodo temporale, e poi inseguita solo per un piccolo tratto fino alla nuova riprogrammazione e si può intuire come tale tecnica tenga in considerazione intrinsecamente cambiamenti nell'ambiente, o di oggetti che si muovono lentamente. Nello stesso tempo però è evidente che non sia suffiente a evitare collisioni con oggetti in rapido movimento. Normalmente quindi l'aggiramento ostacoli è implementato come una legge di controllo reattiva, distinta dalla pianificazione della traiettoria, che opera in real-time durante l'inseguimento di una traiettoria al fine di evitare collisioni.

5.3 Reactive Simulation

La Reactive Simulation è un algoritmo real time sviluppato per l'aggiramento ostacoli, ad integrare la pianificazione dinamica della traiettoria descritta in §5.1. Si tratta di una simulazione che prende in considerazione la cinematica del veicolo, il modello dinamico e la sua incertezza, le condizioni iniziali, le misure dell'ambiente e l'incertezza dei sensori per calcolare in real-time la traiettoria che compierebbe il veicolo per raggiungere un target locale.

La pianificazione dinamica della traiettoria risulta molto veloce computazionalmente ma ha lo svantaggio di una sostanziale differenza tra il cammino pianificato e quello realmente eseguito, di conseguenza è stata sviluppata la Reactive Simulation per computare una predizione più accurata della traiettoria.

La Reactive Simulation viene calcolata ogni volta che un nuovo target locale è pianificato durante il moto in risposta al rilevamento di un ostacolo: se l'output della simulazione è un aggiramento ostacoli sicuro, il veicolo continua a inseguire la nuova traiettoria impostata, in caso contrario si ferma per evitare pericolose collisioni e pianificare un nuovo percorso sicuro.

Un vantaggio di questo approccio è che l'uso di un modello per la simulazione, se integrato con un algoritmo di identificazione dei parametri, permetterebbe la stima on-line dei parametri cinematici. In questo modo si potrebbe tener conto di variazioni parametriche come diversi raggi delle ruote causati dall'usura, inerzia delle masse, etc, che potrebbero influenzare significativamente il moto del veicolo (M. De Cecco, 2002).

L'algoritmo è stato implementato su un prototipo di veicolo autonomo a guida differenziale (Figura 108), equipaggiato con un PXI con un sistema operativo real-time (RTOS) utilizzato per controllare il robot e implementare la Reactive Simulation.

Figura 108: prototipo di AGV a guida differenziale

5.3.1 Modello cinematico

Il prototipo di AGV utilizzato nelle prove è dotato di guida differenziale (Figura 109)

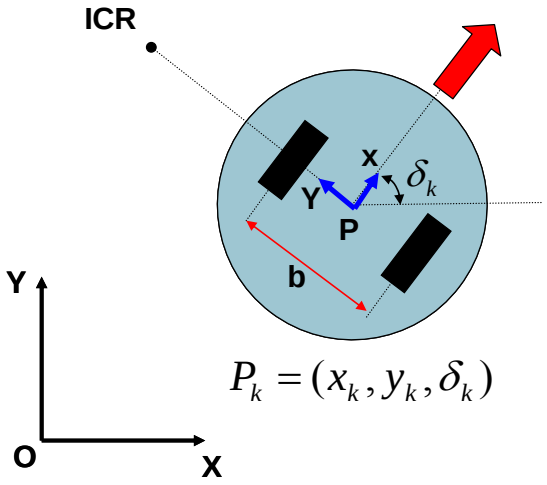


Figura 109: schema del modello differential drive

La posa del veicolo è $P_k = (x_k, y_k, \delta_k)$ dove x_k e y_k sono le coordinate del punto medio dell'asse *b* nel sistema di riferimento ambiente xOy, e l'assetto δ_k è l'angolo tra il sistema xOy e il sistema di riferimento mobile xPy solidale con robot.

Le equazioni di navigazione Inerziale-Odometrica in forma discreta sono le seguenti:

$$\begin{cases} \delta_k = \dfrac{(v_{rk} - v_{lk})}{b} T_c + \delta_{k-1} \\ x_k = (v_k \cos\delta_k) T_c + x_{k-1} \\ y_k = (v_k \sin\delta_k) T_c + y_{k-1} \end{cases} \qquad (5.2)$$

dove b è la distanza tra i centri delle due ruote, v_{rk} e v_{lk} sono le velocità lineari della ruota destra e della ruota sinistra, v_k è la velocità lineare del veicolo relativa al punto medio dell'asse b:

$$v_k = \frac{1}{2} * (v_{rk} + v_{lk})$$

T_c è il periodo del task critico che stima la posa del robot.

5.3.2 Ottimizzazione della simulazione

I parametri cinematici sono stati misurati e poi ottimizzati minimizzando una funzione costo che considera la posa stimata per via odometrica e quella stimata da un sistema di riferimento a triangolazione nell'infrarosso (M. De Cecco, 2000).

Per quanto riguarda i motori CC, è stato utilizzato questo modello:

-parte elettrica

$$V_a(t) = Ri(t) + L\frac{di(t)}{dt} + K_\Phi \omega(t)$$

dove R è la resistenza, L è l'induttanza del circuito elettrico, K_Φ è la costante di coppia, V_a è la tensione di alimentazione, ω è la velocità angolare

-parte meccanica:

$$\tau_m(t) = K_\Phi i(t) = J\dot{\omega}(t) + B\omega(t)$$

Dove τ_m è la coppia motrice, J è l'inerzia rotorica, B è l'attrito viscoso.

Le due parti combinate forniscono il seguente schema a blocchi:

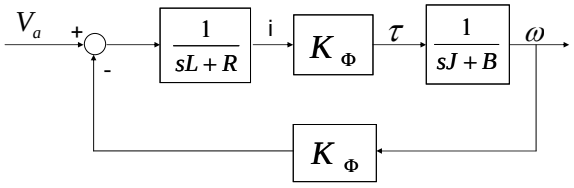


Figura 110: schema a blocchi di motore CC

Il passo di integrazione nella simulazione è stato scelto come un compromesso tra il tempo computazionale (si tratta di una simulazione da effettuare in real-time) e l'accuratezza della traiettoria calcolata. In Tabella 11 sono riportate le differenze tra la simulazione effettuata con passo di integrazione di 1 ms e simulazioni con passo di integrazione crescente, e riassumono un set di test effettuato a diverse velocità (0.2, 0.4, 0.6 m/s) e traiettorie (su una distanza di 4 m) con diversi assetti iniziali rispetto al punto da raggiungere ((90°, 60°, 30°).

Tabella 11: massime differenze simulate aumentando lo step di integrazione

Step di integrazione	Massimo errore simulato
25 ms	8 mm
50 ms	16 mm
100 ms	32 mm
200 ms	61 mm

Il tempo computazionale della Reactive Simulation è un aspetto molto importante nelle applicazioni real-time. È influenzato da vari fattori: dalla lunghezza della traiettoria di cui si fa la predizione , dalla velocità attuale del veicolo, dalla posa iniziale, dallo step di integrazione. In Tabella 12 si riportano i tempi computazionali della Reactive Simulation: è stata simulata una traiettoria di 2 m alla velocità di 0.4 m/s con diversi step di integrazione, e per ogni step con differenti assetti iniziali del veicolo rispetto al punto da raggiungere.

Tabella 12: Tempo computazionale della Reactive Simulation: ogni tempo riportato è la media di simulazioni effettuate con un diverso assetto iniziale del veicolo rispetto al punto da raggiungere

Step of integration	Mean Time
10 ms	69 ms
20 ms	36 ms
30 ms	25 ms
40 ms	19 ms
50 ms	16 ms

Un'incertezza massima di 8 mm può essere considerate nei limiti di un moto sicuro per il robot, mentre un'incertezza massima di 61 mm risulterebbe in una predizione troppo inaccurata. Ovviamente una simulazione più veloce deve essere preferita per le applicazioni real-time: in base a queste considerazioni uno step di integrazione di 50 ms è stato considerato un buon compromesso.

5.3.3 Il sistema di misura

Il prototipo di AGV utilizzato nelle prove sperimentali è equipaggiato con un encoder per ogni ruota di guida (quindi due trattandosi di un veicolo a differential drive) e di un laser scanner.

Per caratterizzare il laser scanner, che fornisce le uniche informazioni riguardo alla distanza degli oggetti presenti nell'ambiente di lavoro, è stato utilizzato un setup di calibrazione, si veda la Figura 111, composto da una barra per l'allineamento ottico con una traslazione misurata con un'accuratezza di ±1 mm, un pannello imperniato centralmente sull'asse ortogonale della barra, la cui rotazione è misurata con un encoder incrementale con 14400 impulsi per giro (ppr).

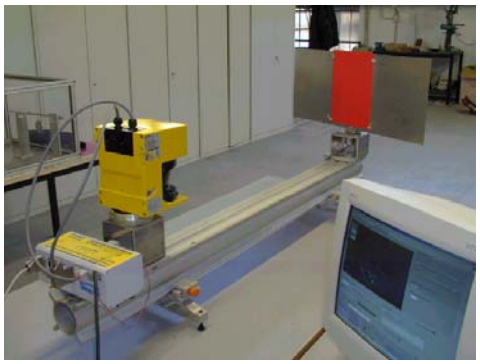


Figura 111: apparato sperimentale utilizzato per la calibrazione del laser

Con questo setup è stato stimata la deviazione standard del rumore, risultata di circa 4 mm, quindi l'effetto dei seguenti parametri di influenza sull'accuratezza del laser: la deriva dovuta alla temperatura, la distanza, il colore delle superfici, il materiale delle superfici, l'angolo di incidenza rispetto all'oggetto.

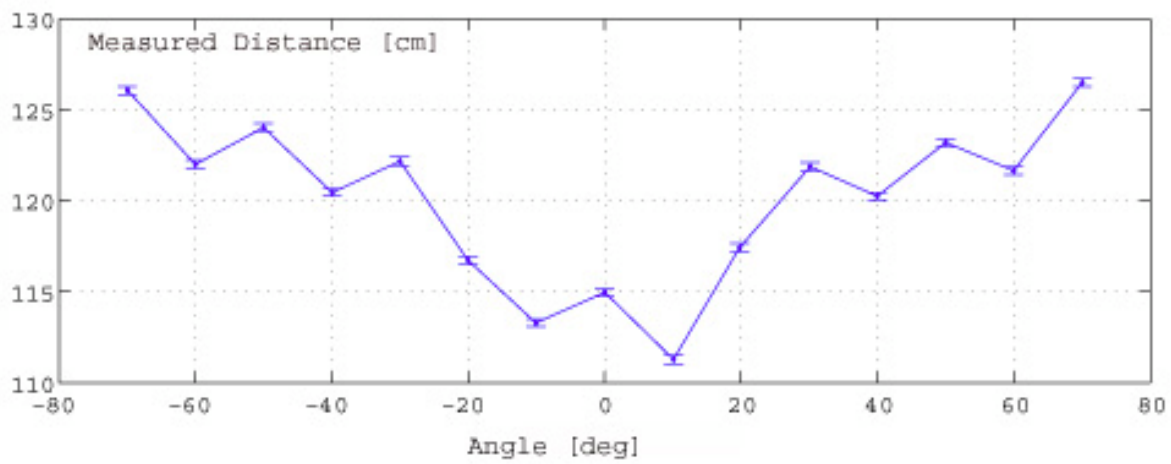


Figura 112: misure in funzione dell'angolo di incidenza. Le misure sono state prese con il target a una distanza fissa di 115 cm con l'apparato di Figura 111

Tra tutte le grandezze di influenza prese in considerazione, l'angolo di incidenza è la variabile che contribuisce maggiormente a degradare l'accuratezza della misura. L'effetto dell'angolo di incidenza sulla distanza misurata è mostrato in Figura 112: è evidente che incertezze di circa 150 mm tra le stime di un profilo di un oggetto si possono verificare solo a causa della rotazione o di effetti prospettici.

Inoltre l'incertezza del laser dipende dalla velocità lineare e angolare del veicolo su cui è montato nel momento in cui viene effettuata la scansione (soprattutto da quella angolare poiché il laser si basa su uno specchietto che ruota).

Il fattore di incertezza è stato combinato con le ipotesi di non correlazione tra i differenti fattori, portando al seguente modello:

$$\varepsilon_{lk}^2 = \varepsilon_0^2 + \varepsilon_\theta^2(\theta_r) + \varepsilon_\omega^2(\omega) + \varepsilon_v^2(v)$$

Dove ε_{lk} è l'incertezza corrispondente ai dati della scansione, ε_0 è il termine dovuto alla ripetibilità, ε_θ è il termine dovuto all'angolo di incidenza, ε_ω è il termine dovuto alla variazione dell'assetto del veicolo, ε_v è il termine dovuto alla velocità lineare del veicolo.

La stima dell'incertezza così ottenuta è stata utilizzata per applicare un margine di sicurezza all'output della Reactive Simulation.

Questa caratterizzazione del laser scanner è importante anche perché tale tipo di sensore può essere utilizzato per stimare la posa del veicolo (M. De Cecco, et al, 2006).

5.3.4 Schema logico della Reactive Simulation

L'interazione tra i diversi moduli spiegati in precedenza è schematizzata in Figura 113:

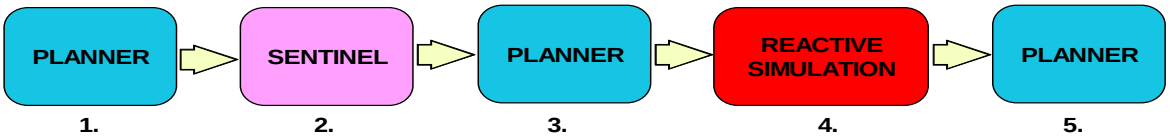


Figura 113: schema logico dell'interazione fra i differenti moduli

Inizialmente il Planner sceglie un target locale nei pressi di una delle aperture rilevate, e il veicolo si muove verso di esso: a titolo di esempio in Figura 114 il veicolo si muove verso il target finale.

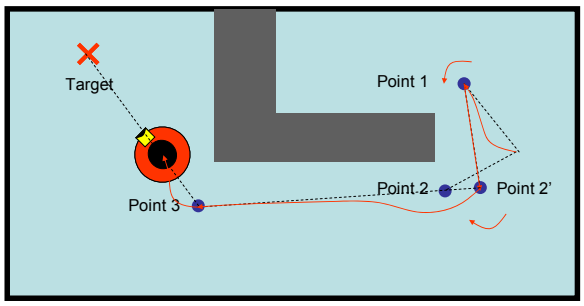


Figura 114: il veicolo si muove verso il target finale

Mentre il veicolo si muove, la Sentinel controlla le scansioni del laser per rilevare eventuali ostacoli. In pratica viene controllato un corridoio che parte dal robot e finisce sul punto finale del segmento P2, il target locale, come è raffigurato in Figura 115

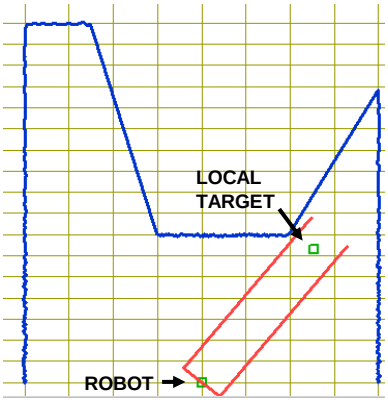


Figura 115: esempio di scansione del laser (linea blu) con "Sentinel", rappresentata dalla linea rossa rettangolare. La Sentinel controlla un corridoio che parte dalla posizione attuale del veicolo e finisce sul target locale.

Se vengono rilevati ostacoli di fronte al veicolo, all'interno del corridoio di sicurezza, la Sentinel allerta il Planner che trova un nuovo punto da raggiungere, con una pianificazione reattiva molto semplice (il punto 5 in Figura 116).

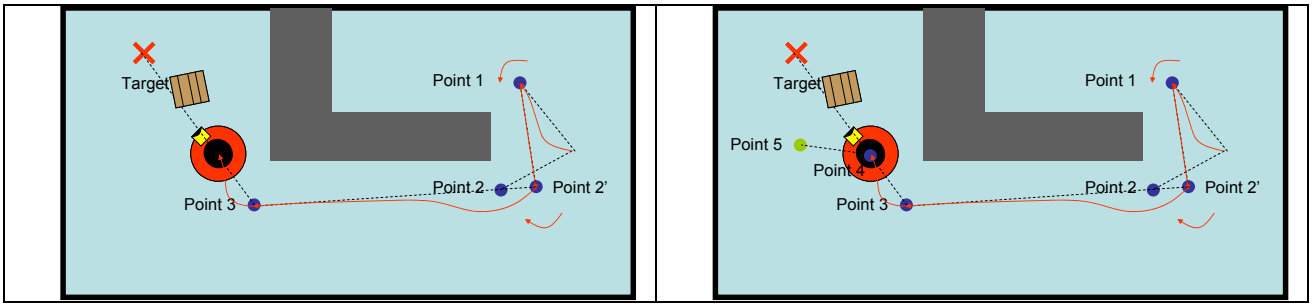


Figura 116: appare un ostacolo (ad esempio uno scatolone o un uomo in un magazzino), il Planner pianifica il punto 5 (reactive path planning)

A questo punto la Reactive Simulation simula la traiettoria che il veicolo compierebbe per raggiungere il target locale, verifica che la traiettoria calcolata non incontri ostacoli (Figura 117) e comunica il risultato al Planner. In questa verifica devono essere tenuti in considerazione l'incertezza del modello, l'incertezza dei sensori e adeguati margini di sicurezza.

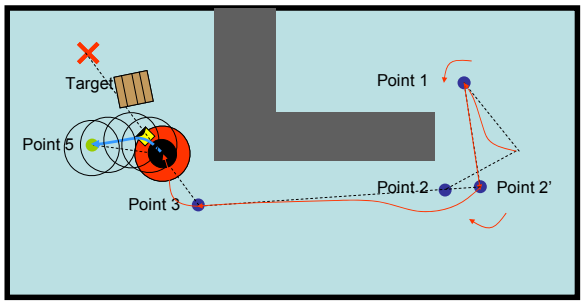


Figura 117: La Reactive Simulation calcola la traiettoria per raggiungere il target locale e verifica se è sicura o meno

Con l'uscita della Reactive Simulation, il Planner decide se continuare a raggiungere il target pianificato oppure no. In quest'ultimo caso, si fermerà per evitare collisioni pericolose e pianificherà un nuovo percorso sicuro. In il veicolo continua a percorrere la traiettoria fino a raggiungere il target.

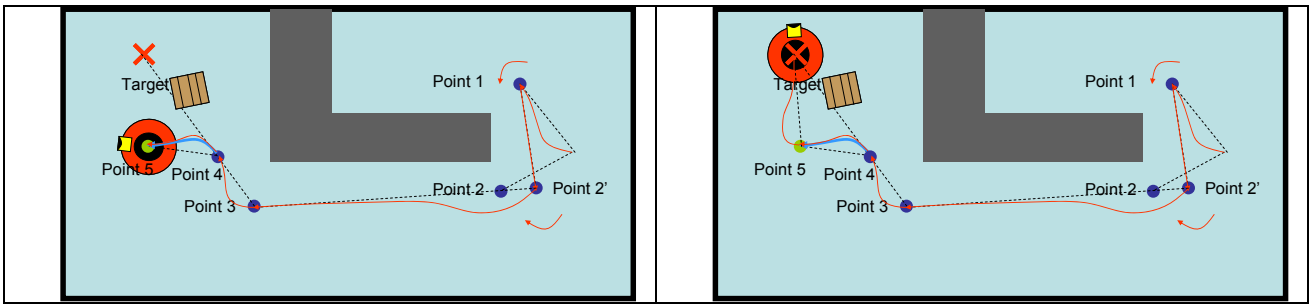


Figura 118: il veicolo raggiunge il punto 5, che la Reactive Simulation ha giudicato raggiungibile in sicurezza, e poi al target finale.

Se integrata con un algoritmo di identificazione on-line dei parametri, la Reactive Simulation può tenere in considerazione variazione dei parametri cinematica, come il raggio delle ruote che può mutare con l'usura, l'inerzia della masse, angoli di riferimento, ecc, che possono influenzare sensibilmente il movimento del veicolo.

Uno schema completo del modulo di guida (Drive Module), che comprende tutti i vari blocchi spiegati in precedenza, è mostrato in Figura 119.

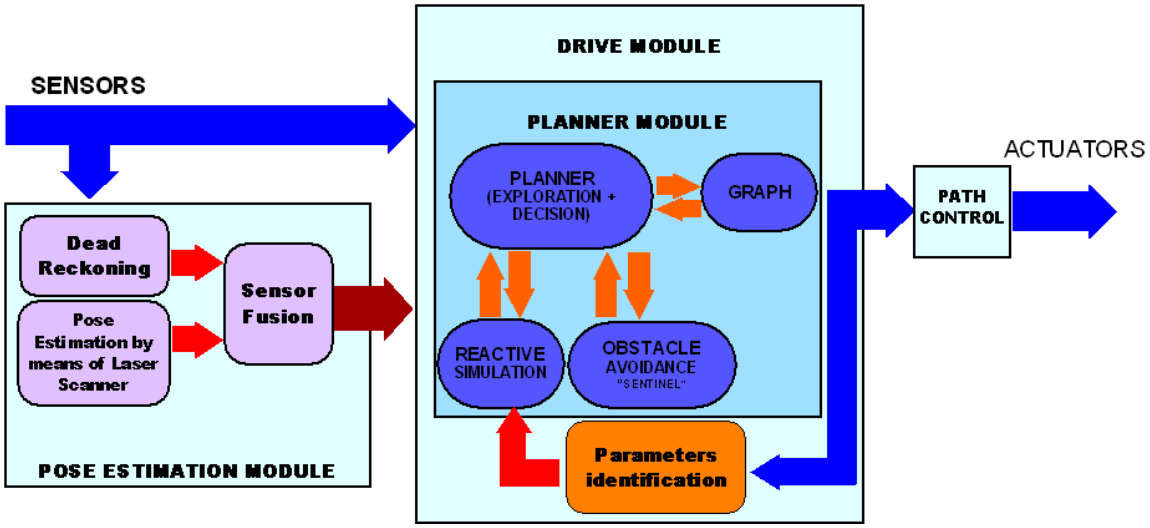


Figura 119: schema completo del modulo di guida (Drive Module), e interazione con l'algoritmo di sensor fusion e un eventuale algoritmo di identificazione on-line dei parametri.

5.3.5 Implementazione Real-time

La Reactive Simulation è stata implementata e testate in applicazioni real-time utilizzando il prototipo di veicolo autonomo mostrato in Figura 108. Il sistema è equipaggiato con un PXI (National Instruments) a 333 MHz fornito di un sistema operativo real-time (RTOS). Il software, come schematizzato in Figura 120, è composto da tre task principali:

- TASK 1: stima la posa per via odometrica coi dati forniti dagli encoder e si occupa dell'inseguimento di traiettoria
- TASK 2: comunica col laser scanner
- TASK 3: contiene il Planner, il Graph Manager, la Sentinel, il Path Control e la Reactive Simulation.

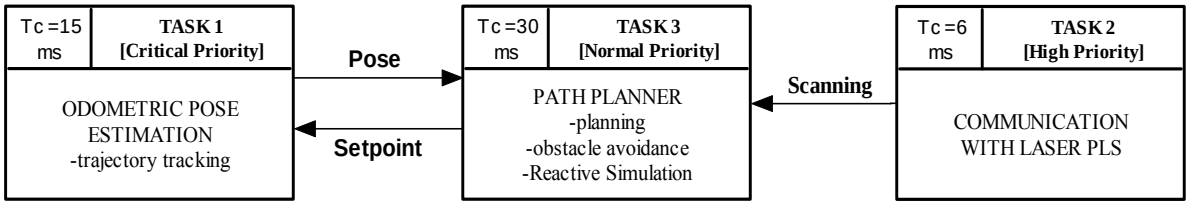


Figura 120: schema dei tre task principali

La priorità statica dei task è stata impostata in accordo con l'algoritmo rate monotonic, il quale assegna la priorità a ogni task in relazione al suo periodo, in modo che il periodo più breve abbia la priorità più alta. Si è quindi impostate le priorità:

- TASK 2: ha un tempo di esecuzione nel caso peggiore di circa 2-3 ms, eccetto nel caso in cui una scansione viene ricevuta, un periodo di 6 ms, e quindi ha la priorità critica
- TASK 1: ha un tempo di esecuzione nel caso peggiore di circa 4-5 ms, un periodo di 15 ms, e ha quindi priorità alta
- TASK 3: ha un tempo di esecuzione nel caso peggiore di circa 14-16 ms, un periodo di 30 ms, e ha quindi priorità normale

Con queste priorità, quando viene ricevuta una scansione dal laser, ogni 200 ms, il TASK 2 deve eseguire un grande numero di calcoli utilizzando il processore per circa 14 ms e quindi ritardando la stima della posa, che invece è un aspetto critico. Di conseguenza al TASK 1 è stata assegnata priorità critica, e al TASK 2 priorità alta. L'algoritmo di Reactive Simulation è compreso nel Planner Module (TASK 3), e quindi ha priorità normale. In Figura 121 è rappresentato un esempio del tempo impiegato dal processore a eseguire i

task con le priorità così impostate: si nota che il task a priorità critica è sempre all'interno del periodo assegnato, e gli altri due task raramente eccedono il proprio periodo (per meno di 5 ms ad esclusione del comando di start).

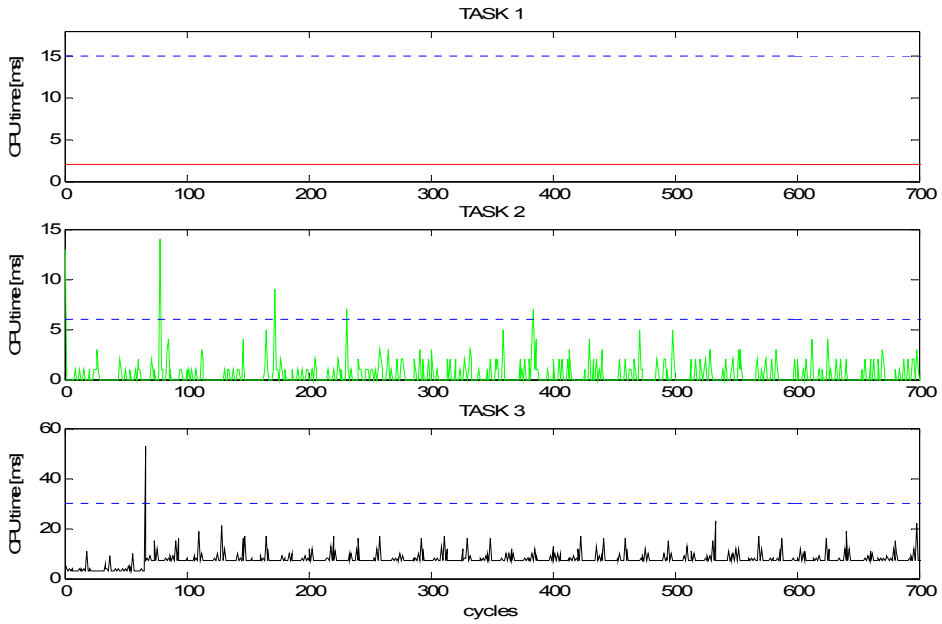


Figura 121: tempo di calcolo del processore per i tre task. La linea tratteggiata è il periodo assegnato ai task. Il primo picco che appare nel task 3 è dovuto al comando di start.

5.3.6 Verifiche sperimentali

Il prototipo utilizzato nelle prove sperimentali è dotato di cinematica differenziale. Ha un diametro di 1.05 m, è alto 0.9 m e la velocità massima è di circa 2.5 m/s. E' dotato di un PXI (National Instruments) con un sistema operativo real-time (RTOS), su cui è stata implementata le Reactive Simulation.

Sono state testate molte traiettorie con ostacoli improvvisi e in movimento. È stata considerata l'incertezza di modello e del laser, la velocità lineare e angolare del veicolo corrispondenti all'istante di tempo di arrivo della scansione laser per eseguire la Reactive Simulation e quindi un inseguimento di traiettoria sicuro. Infatti, la traiettoria viene pianificata basandosi su una scansione più vicina al veicolo rispetto a quella realmente ricevuta (si ricorda che un scansione è composta da misure di distanza degli oggetti separate da un certo step angolare) di una quantità proporzionale all'incertezza del laser e di modello (si veda la linea tratteggiata in Figura 122: e Figura 123 che rappresenta la scansione avvicinata). Deve essere sottolineato che l'incertezza del laser dipende soprattutto dalla velocità lineare e angolare attuale del veicolo.

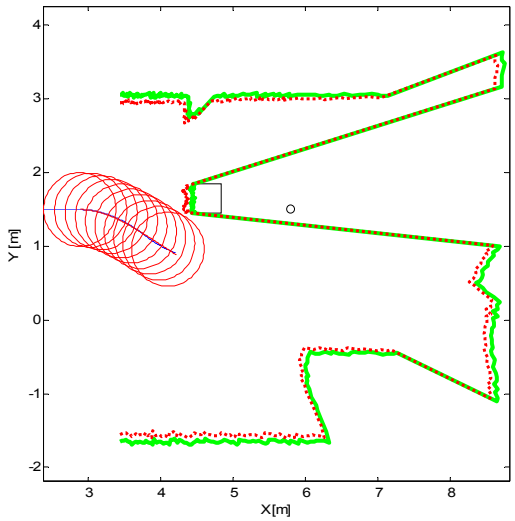


Figura 122: un nuovo target locale è pianificato e la Reactive Simulation calcola la traiettoria. La linea continua è la scansione del laser ricevuta, mentre quella tratteggiata è la scansione calcolata più vicina al veicolo.

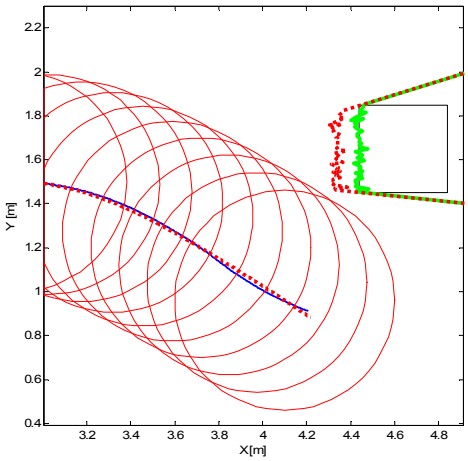


Figura 123: dettaglio della Reactive Simulation dove la linea continua è la reale traiettoria fatta dal veicolo dopo la simulazione (linea tratteggiata)

Un esempio di aggiramento ostacoli è mostrato nelle seguenti figure, dove un ostacolo, uno scatolone, appare improvvisamente di fronte al veicolo in traiettoria. Il Planner pianifica un nuovo target locale e la Reactive Simulation predice la traiettoria (tra il secondo e il terzo frame). Quindi il veicolo continua a percorrere la traiettoria sicura pianificata.

Figura 124: esempio di aggiramento ostacoli

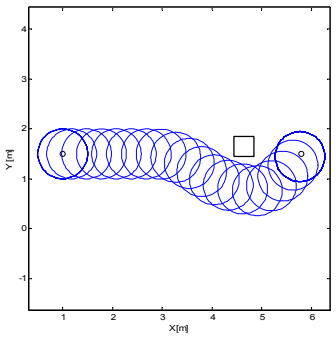


Figura 125: intera traiettoria realmente compiuta

Bibliografia

Maybeck P.S., The Kalman Filter. An introduction to concepts. In *Autonomous Robot Vehicles*, I.J. Cox, G.T Wilfong Editors, Springer-Verlag.

S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artec House, 1999.

R.E. Moore. *Interval Analysis*. Prentice Hall, 1966

D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, 1980.

M. De Cecco, 2000, A new concept for triangulation measurement of AGV attitude and position, in: *Measurement Science and Technology,* vol 11, pp 105-110

M. De Cecco, 2002, Self-Calibration of AGV Inertial-Odometric Navigation Using Absolute-Reference Measurement, in: *IEEE Instrumentation and Measurement Technology Conference*, Anchorage, AK, USA, 21-23 May

M. De Cecco, L. Baglivo, E. Ervas, E. Marcuzzi, 2006, Asynchronous And Time-Delayed Sensor Fusion Of A Laser Scanner Navigation System And Odometry, *XVIII Imeko World Congress Metrology For a Sustainable Development, September, 17 – 22, 2006, Rio de Janeiro, Brazil, in publication*

S Shoval, I Zeitoun, E Lenz, *Implementation of a Kalman filter in positioning for Autonomous Vehicles, and its Sensitivity to the Process Parameters*, Int J Adv Manuf Technol, Vol 13, pp 738-746, 1997.

J. Borenstein, L Feng, *Measurement and Correction of Systematic Odometry Errors in Mobile Robots*, IEEE Transactions on Robotics and Automation, Vol 12, No 5, Oct. 1996.

Adam, E Rivlin, H. Rotstein, *Fusion of Fixation and Odometry for Vehicle Navigation*, IEEE Int. Conf. On Robotics and Automation, Detroit, Michigan, May 1999.

S. I. Roumeliotis, G. S. Sukhatme, G. A. Bekey, *Circumventing Dynamic Modeling: Evaluation of the error-state Kalman Filter applied to mobile robot localization*, Proc. 1999 IEEE International Conference on Robotics and Automation, Detroit, May 10-15, pp. 1656-1663, 1999.

Y. K. Tham, H. Wang, E. K. Teoh, *Multi-sensor fusion for steerable four-wheeled industrial vehicles*, Control Engineering Practice, vol. 7, pp. 1233-1248, 1999.

Guide to Expression of Uncertainty in Measurements 1993, ISO.

J. Chestnutt, J. Kuffner, K. Nishiwaki, S. Kagami, 2003, *Planning Biped Navigation Strategies in Complex Environments,* in:IEEE Int'l Conf. on Humanoid Robotics (Humanoids 2003)

A. Stentz, 1994, *Optimal and Efficient Path Planning for Partially-Known Environments,* Proc. Of IEEE International Conference on Robotics and Automation, May 1994

Enrico Ervas, *Autolocalizzazione e mappatura ambientale mediante laser a scansione in veicoli autonomi*. Master's thesis, Universitá degli studi di Padova, Corso di Laurea in Ingegneria Elettronica, Dipartimento di Elettronica e Informatica, 2001-02

Cang Ye and Johann Borenstein. *Characterization of a 2-d laser scanner for mobile robot obstacle negotiation*, In International Conference on Robotics and Automation. IEEE, May 2002

Feng Lu and Evangelos Milios. *Robot pose estimation in unknown environment by matching 2d range scans*. 1994.

Andrew W. Fitzgibbon. *Robust registration of 2d and 3d point sets*. 2001.

Martin D. Adams. *Coaxial range measurement - current trends for mobile robotic applications.* IEEE SENSORS, 2(1), February 2002.

Peter Biber. *The normal distributions transform: A new approach to laser scan matching*. 2003.

Daniel Lecking, Oliver Wulf, and Bernardo Wagner. *Variable pallet pick-up for automatic guided vehicles in industrial environments*. 2006.

Viet Nguyen, Agostino Martinelli, Nicola Tomatis, and Roland Siegwart. *A comparison of line extraction algorithms using 2d laser range Finder for indoor mobile robotics*. In International Conference on Intelligent Robots and Systems.

P. de Groen. *An introduction to total least squares*. In Nieuw Archief voor Wiskunde, number 14 in Vierde, pages 237_253. 1996.

Michael Krystek and Mathias Anton, *A weighted total least-squares algorithm for fitting a straight line*, MEASUREMENT SCIENCE AND TECHNOLOGY 18 (2007) 3438–3442

Richard O. Duda and Peter E. Hart. *Use of the hough transformation to detect lines and curves in pictures.* In Comm. ACM, volume 15, pages 11_15. Arti_cial Intelligence Center, January 1972.

Wright, Alden H. *Genetic Algorithms for Real Parameter Optimization.* 1991.

Peter Knoll, Bob Kenny, Siroos Mirzaei, Karl Koriska, Horst Köhn, Martin Neumann. *A mu+lambda Evolutionary Algorithm for Reconstruction*. IEEE Transaction on Numerical Computation. Dicembre 2002, Vol. 6, 6.

Kumara Sastry, David Goldberg, Graham Kendall. *Genetic Algoritms.*

P. Besl and N. McKay. *A method for registration of 3D shapes*. TPAMI 14(2):239-256, 1992.