

ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

---

Seconda Facoltà di Ingegneria - Sede di Cesena  
Corso di Laurea Magistrale in Ingegneria Elettronica e delle  
Telecomunicazioni per l'Energia

ALGORITMI PER IL MAPPING AMBIENTALE  
MEDIANTE ARRAY DI ANTENNE

Elaborata nel corso di: RETI DI SENSORI WIRELESS PER  
MONITORAGGIO AMBIENTALE LM

*Tesi di Laurea di:*  
JACOPO DALL'ARA

*Relatore:*  
Chiar.mo Prof. DAVIDE  
DARDARI

*Correlatori:*  
Dott.Ing. ANNA GUERRA  
Dott.Ing. FRANCESCO GUIDI

---

ANNO ACCADEMICO 2016–2017  
SESSIONE UNICA







# Acronimi

**BF** filtri di Bayes

**CM** Matrice di Covarianza

**EKF** Extended Kalman Filter

**FB** Feature Based

**GB** Grid Based

**KF** filtri di Kalman

**mmW** onde millimetriche

**OF** Optimal Filtering

**OG** Occupancy Grid

**PDF** densità di probabilità

**PF** Particle Filter

**SLAM** Simultaneous Localization And Mapping

**UWB** Ultra Wide Band



# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Modelli matematici per il mapping ambientale</b>	<b>7</b>
2.1	Optimal Filtering . . . . .	7
2.2	Filtri di Bayes . . . . .	8
2.3	Filtri di Kalman . . . . .	11
2.3.1	Extended Kalman Filter . . . . .	13
2.4	Particle Filter . . . . .	15
2.5	Occupancy Grid . . . . .	16
2.6	Simultaneous Localization And Mapping . . . . .	18
<b>3</b>	<b>Formulazione di un nuovo modello probabilistico per il mapping</b>	<b>23</b>
3.1	Definizione del problema . . . . .	23
3.2	Modello probabilistico adottato . . . . .	29
3.3	Correlazione tra celle nel modello di misura . . . . .	31
3.4	Esempio di calcolo dell'occupancy . . . . .	32
3.4.1	Esempio numerico . . . . .	35
<b>4</b>	<b>Risultati</b>	<b>37</b>
4.1	Introduzione . . . . .	37
4.2	Struttura del simulatore . . . . .	43
4.3	Dati di simulazione . . . . .	44
4.4	Grafici ottenuti dalle simulazioni . . . . .	45
4.4.1	Grafici della mappa stimata relativa al primo scenario	46
4.4.2	Grafici dei confronti tra i tre scenari simulati . . . . .	59
4.5	Commento dei risultati . . . . .	60
<b>5</b>	<b>Conclusioni e Sviluppi Futuri</b>	<b>63</b>
<b>6</b>	<b>Appendice 1 - Grafici rimanenti</b>	<b>65</b>
6.1	Grafici per le restanti mappe analizzate . . . . .	65
6.1.1	Grafici della mappa stimata relativa al secondo scenario	66
6.1.2	Grafici della mappa stimata relativa al terzo scenario	79

6.2	Grafici per i restanti confronti effettuati . . . . .	92
<b>7</b>	<b>Appendice 2 - Codice MatLab</b>	<b>107</b>
	<b>Bibliografia</b>	<b>118</b>



# Capitolo 1

## Introduzione

La capacità di costruire delle mappe riportanti informazioni di tipo statico di un ambiente sconosciuto, tramite dei sensori disposti all'interno di esso, è un problema soggetto a numerose ricerche scientifiche svolte in ogni parte del mondo nelle ultime due decadi, in quanto è collegato ad innumerevoli applicazioni pratiche. Con ambiente si intendono sia i casi outdoor, condizione spaziale per il quale molte tecniche sono state perfezionate, sia indoor dove soluzioni efficienti sono ancora in fase di studio. Le ricostruzioni di mappe ambientali non sono una mera rappresentazione di grandezze fisiche misurate ma anche al rispettivo intervallo di confidenza. Questo tipo di attività è fondamentale in vari ambiti ingegneristici. Un eccellente esempio si rifà al campo della sicurezza: in caso di incendio dentro un edificio potrebbe essere utile avere a disposizione una tecnologia capace di ricostruire lo stato interno degli edifici, e di comunicarne una mappa dell'ambiente stimata al soccorritore, specialmente in caso di scarsa visibilità data dal fumo. Al giorno d'oggi si hanno un gran numero di strumenti matematici che permettono *la costruzione di mappe stimate a partire da dati raccolti da sensori posti in un dispositivo mobile che esplora l'ambiente*. Quest'ultimo, sfruttando le informazioni messe a disposizione dai sensori, applicherà uno degli algoritmi definiti in letteratura, elaborando ed utilizzando i dati per il fine preposto. Le informazioni che si vogliono ottenere dai sensori possono essere la topologia dell'ambiente, oppure quest'ultima unita alla posizione del mobile all'interno dello scenario studiato. Tale problema, estremamente complesso ed importante, prende il nome di Simultaneous Localization And Mapping (SLAM).

Ogni metodo proposto in letteratura per la risoluzione dello SLAM, o della sua versione semplificata chiamata *Mapping*, ovvero quella riguardante solo la ricostruzione dello scenario ma non della posizione del mobile, presenta caratteristiche peculiari, con un proprio livello di precisione e affidabilità. Si possono dividere in varie categorie:

- Approccio *Feature Based (FB)*: misura la posizione di un set di land-

mark (punti di interesse la cui posizione all'interno dell'ambiente va stimata per la risoluzione del problema). Negli spazi aperti con punti predefiniti noti a priori tale metodo risulta il piú indicato. Risulta un approccio computazionalmente veloce, ma la necessità di una conoscenza a priori sulla forma dei landmark rappresenta un grosso limite implementativo.

- Approccio *Occupancy Grid (OG)*: lo spazio é diviso in tantissime celle e si cerca di stimare il valore di ognuna di esse. É generalmente usato per gli spazi indoor. Tipicamente l'OG risulta pesante a livello computazionale, con un grande quantitativo di memoria richiesto, ma é estremamente preciso ed in piú non richiede informazioni a priori.

Tutti e due gli approcci possono essere risolti tramite l'utilizzo degli *Extended Kalman Filter (EKF)*, estensione linearizzata della famiglia dei filtri di Kalman (KF), che a loro volta sono un sottogruppo che utilizza una rappresentazione gaussiana per i filtri di Bayes (BF). Al giorno d'oggi risultano lo strumento piú usato per risolvere il problema dello SLAM. L'approccio dell'OG invece, puó essere risolto anche associando ad ogni cella una certa probabilità di occupazione, famiglia di soluzioni nella quale ricade anche l'algoritmo proposto in questa tesi.

Risulta quindi naturale cercare di capire quali possono essere gli strumenti utili a risolvere il problema dello SLAM nel migliore dei modi. Prendendo come riferimento, per semplicitá di comprensione, un robot il cui compito é quello di esplorare un ambiente sconosciuto e ricostruire una mappa di quest'ultimo, esso sará in possesso di sensori in grado di percepire il mondo esterno e di un microcontrollore per l'elaborazione dei dati raccolti. I metodi matematici, accennati precedentemente, serviranno a dare una rappresentazione piú o meno veritiera del mondo circostante, a meno di incertezze legate alla misura. Il processo di stima sará soggetto a:

- Errori di stima che non saranno mai completamente eliminabili, ed andranno sempre a sporcare la misura, risultando sempre parzialmente imprecisa. Pertanto é importante avere un'idea della qualità della stima oltre che del suo valore puntuale.
- Una limitata portata sia in termini di distanza misurabile che di problemi legati alla presenza di ostacoli ambientali. Essi porteranno alla necessità di far muovere il mobile all'interno dell'ambiente per poterlo ricreare con esattezza.
- Altri fattori legati alla tecnologia utilizzata. Ad esempio, i sistemi di visione (camera-based) o sistemi basati sulla tecnologia laser, pur offrendo le prestazioni migliori, risultano inadatti per ambienti ricchi di ostacoli, e quindi in condizioni di NLOS.

Per superare quest'ultimo inconveniente, possono essere adottate soluzioni radio. In questo ambito, la tecnologia Ultra Wide Band (UWB) risulta essere la migliore dal punto di vista dell'accuratezza del ranging (stima della distanza) grazie all'elevata risoluzione temporale (il segnale trasmesso é un impulso della durata di pochi nano-secondi) e alla robustezza nei confronti del multipath. Tuttavia, nel tentativo di avvicinarsi il piú possibile alla risoluzione angolare del laser, la possibilitá di adottare antenne massive, operanti alle onde millimetriche (mmW) (es: 60 GHz) , risulta una soluzione promettente. Infatti, nonostante il link budget sia sfavorevole a 60 GHz, la lunghezza d'onda é dell'ordine dei millimetri e questo permette di compattare un gran numero di antenne in poco spazio, come ad esempio uno smartphone. Avere molte antenne a disposizione permette a sua volta di avere un guadagno di antenna superiore (mitigando il path-loss) e una direttivitá molta elevata (ad esempio, con 400 antenne la risoluzione angolare é di 6 gradi). Capacitá di ranging e risoluzione angolare risultano i due requisiti principali per applicazioni di mapping e quindi rappresentano le chiavi per discriminare tra una tecnologia e l'altra.

La grande sfida del mapping é quella di riuscire a ricreare la zona circostante il robot, mitigando nel miglior modo possibile tutta una serie di problemi non trascurabili legati all'incertezza delle misure ed ai vincoli tecnologici. I piú importanti sono:

1. Le dimensioni eccessive dell'ambiente da studiare potrebbero portare allo studio di vettori di stato estremamente grandi come nel caso di ambiente diviso in celle, premessa di lavoro del Grid Based (GB) e della soluzione proposta in questo elaborato. Questo problema non porta ad errori aggiuntivi, ma ad una irrisolvibilitá dell'obbiettivo per via dei tempi computazionali potenzialmente infiniti.
2. Spesso ci si trova a studiare ambienti tempo varianti. Questo aumenta ancora la complessitá del problema e puó far escludere alcuni tipi di algoritmi che non tengono conto di ció. Va specificato che questa condizione si verifica poco frequentemente, portando questo problema ad essere assolutamente situazionale. Spesso infatti ambienti tempo varianti, che non modificano drasticamente la propria topologia, possono essere studiati tramite algoritmi tempo invarianti senza commettere grandi errori di stima.
3. Spesso ci si trova senza alcun tipo di conoscenza a priori dell'ambiente da studiare e quindi senza informazioni da passare al robot per iniziare a muoversi nella direzione giusta. Infatti sotto questa condizione non si possono dare input a priori sul movimento da compiere per lo studio completo dell'ambiente. Il robot quindi deve essere programmato per far fronte a tutta una serie di eventuali inconvenienti che

possono emergere durante l'acquisizione di una mappa: vicoli ciechi, informazioni sbagliate, ostacoli non rilevabili, ecc..

Lo scopo di questo elaborato é quello di fornire una veloce introduzione teorica a tali strumenti matematici, evidenziandone pregi e difetti, per poi proporre un metodo nuovo e piú efficiente che vada a sostituire, o parzialmente completare, gli strumenti usati oggi.

La tesi é divisa nei seguenti capitoli:

1. Un'introduzione teorica con nozioni e dimostrazioni di base per comprendere:
  - BF
  - KF
  - Particle Filter (PF)
  - OG
  - SLAM
2. Spiegazione dettagliata dell'algoritmo sviluppato per questa tesi basato sull'occupancy grid e sua implementazione in un simulatore
3. Grafici e discussioni dei risultati ottenuti
4. Conclusioni

## Capitolo 2

# Modelli matematici per il mapping ambientale

### 2.1 Optimal Filtering

Con il termine *Optimal Filtering (OF)* si vuole indicare un metodo, o piú in generale una classe di metodi, che possono essere usati per stimare lo stato di un sistema, indirettamente analizzato partendo da una o piú osservazioni rumorose [5]. Prima di descrivere nel dettaglio i BF, vanno introdotti due concetti chiave:

- Lo *stato del sistema* é una collezione di variabili dinamiche, come ad esempio, posizione, velocitá, orientamento, etc., che descrivono il sistema in esame. Nel caso dello SLAM, lo stato del sistema contiene anche la posizione dei landmark in esame oltre che quella del robot.
- Il *rumore di misura* invece rappresenta l'incertezza dell'osservazione e quindi le misure non saranno funzioni deterministiche dello stato del sistema ma saranno caratterizzate da una certa distribuzione di valori attorno al valore vero.

In generale i sistemi analizzati non sono di tipo stocastico, ovvero le loro manifestazioni non seguono leggi perfettamente casuali analizzabili a livello probabilistico. Tuttavia, a causa di incertezze di varia natura, stocastiche sono le misure da cui lo stato del sistema viene stimato. Nel seguito, si introdurranno i BF capaci di stimare lo stato del sistema data la storia delle misure.

## 2.2 Filtri di Bayes

I BF stimano lo stato del sistema al tempo  $t$  dato un insieme di misure fino all'istante attuale. Definendo [5]:

- $\mathbf{x}_t$ : lo stato del sistema all'istante  $t$
- $\mathbf{x}_{0:T} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_t, \dots, \mathbf{x}_T\}$ : l'insieme delle stime da 0 a  $T$
- $\mathbf{y}_{0:T} = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_t, \dots, \mathbf{y}_T\}$ : l'insieme delle misure da 0 a  $T$ .

L'obiettivo é quello di calcolare la distribuzione congiunta a posteriori di *tutti gli stati date tutte le misure*, ovvero:

$$\text{Bel}(\mathbf{x}_t) = p(\mathbf{x}_{0:T} | \mathbf{y}_{1:T}) = \frac{p(\mathbf{y}_{1:T} | \mathbf{x}_{0:T}) p(\mathbf{x}_{0:T})}{p(\mathbf{y}_{1:T})} . \quad (2.1)$$

Dove:

- $p(\mathbf{x}_{0:T})$ : distribuzione a priori del sistema.
- $p(\mathbf{y}_{1:T} | \mathbf{x}_{0:T})$ : funzione di verosimiglianza per le misure.
- $p(\mathbf{y}_{1:T})$ : costante di normalizzazione pari a:

$$p(\mathbf{y}_{1:T}) = \int_{-\infty}^{\infty} p(\mathbf{y}_{1:T} | \mathbf{x}_{0:T}) p(\mathbf{x}_{0:T}) d\mathbf{x}_{0:T} . \quad (2.2)$$

Sfortunatamente la (2.1) congiunta per tutti gli stati ha il grosso svantaggio di dover essere ricalcolata ogni volta che viene acquisito un nuovo dato. Questo risulta essere decisamente gravoso perché si traduce in una velocità computazionale sempre più decrescente, man mano che l'algoritmo é in corso. Tuttavia, questo problema emerge solo quando si intende calcolare la distribuzione a posteriori completa (quindi congiunta a tutti gli stati) per ogni istante temporale. Per rilassare questo vincolo, é possibile restringere il problema al calcolo della marginale facendo ricorso alle *proprietá di Markov*. In questo caso, il modello degli stati e delle misure sará definito nel seguente modo:

1. Inizialmente viene definita la **densitá di probabilitá (PDF) a priori**  $p(\mathbf{x}_0)$  riferita all'istante iniziale  $t = 0$  che indica la conoscenza iniziale dello stato del sistema;
2. Successivamente viene descritta la **dinamica del sistema**, o modello di mobilitá, e la sua incertezza tramite una catena di Markov, definita in termini di PDF di transizione  $p(\mathbf{x}_t | \mathbf{x}_{t-1})$ ;

3. Infine, il **modello di misura** descrive come la misura al tempo  $t$ , cioè  $\mathbf{y}_t$ , é legata allo stato del sistema  $\mathbf{x}_t$ . Questa dipendenza é espressa tramite una PDF della misura condizionata allo stato, indicata come  $p(\mathbf{y}_t|\mathbf{x}_t)$ .

Di conseguenza, il modello degli stati che si ottiene puó essere scritto nella seguente forma [5]:

$$\begin{aligned} \mathbf{x}_0 &\sim p(\mathbf{x}_0) \\ \mathbf{x}_t &\sim p(\mathbf{x}_t|\mathbf{x}_{t-1}) \\ \mathbf{y}_t &\sim p(\mathbf{y}_t|\mathbf{x}_t) . \end{aligned} \tag{2.3}$$

Questo modello risulta essere Markoviano poiché rispetta le seguenti proprietà:

1. **Proprietá di Markov sugli stati.**

Gli stati  $\mathbf{x}_t$  con  $t = 0, 1, 2, \dots$  formano una catena di Markov se  $\mathbf{x}_t$  dato  $\mathbf{x}_{t-1}$  é indipendente da tutto quello che é avvenuto prima di  $t - 1$ , ovvero:

$$p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{2.4}$$

Lo stesso vale per il passato rispetto agli stati futuri:

$$p(\mathbf{x}_{t-1}|\mathbf{x}_{t:T}, \mathbf{y}_{t:T}) = p(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{2.5}$$

Con la dinamica del sistema si formalizza analiticamente la variazione di stato del sistema nel tempo. Nella stima della posizione questa distribuzione rappresenta analiticamente come si muove il mobile di riferimento: a che posizione si puó trovare il robot al tempo  $t$  considerando solo la posizione all'istante  $t - 1$ ? Questo modello dipende fortemente dalle informazioni disponibili nel processo di stima. Questo valore tramite Markov porta ad avere:

$$p(\mathbf{x}_{0:T}) = p(\mathbf{x}_0) \sum_{t=1}^T p(\mathbf{x}_t|\mathbf{x}_{t-1}) , \tag{2.6}$$

ovvero l'indipendenza dello spostamento tra istanti di tempo non successivi.

2. **Proprietá di Markov sull'indipendenza delle misure.**

La misura attuale  $\mathbf{y}_t$  dato lo stato attuale  $\mathbf{x}_t$  é indipendente dalla storia degli stati e delle misure, ovvero:

$$p(\mathbf{y}_t|\mathbf{x}_{1:t}, \mathbf{y}_{1:t-1}) = p(\mathbf{y}_t|\mathbf{x}_t) \tag{2.7}$$

Il modello di misura descrive la verosimiglianza di una data osservazione  $\mathbf{y}_t$  condizionata allo stato del riferimento  $\mathbf{x}_t$ . Dipende quindi

dallo stato corrente e soprattutto dalla qualità del sensore nel non commettere errori in fase di misurazione. Utilizzando la terza legge di Markov:

$$p(\mathbf{y}_{1:T}|\mathbf{x}_{0:T}) = \prod_{t=1}^T p(\mathbf{y}_t|\mathbf{x}_t) . \quad (2.8)$$

Questa equazione significa che la misura per un qualunque istante temporale, dipende solo dalla posizione nel medesimo tempo.

Attraverso le tre leggi di Markov, la (2.1) da calcolare si semplifica fino a diventare:

$$p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}) \approx p(\mathbf{y}_{1:T}|\mathbf{x}_{0:T}) p(\mathbf{x}_{0:T}) \quad (2.9)$$

Tuttavia, come discusso in precedenza, in applicazioni reali risulta poco pratico implementare la (2.9) data la complessità computazionale. Invece che calcolare la PDF congiunta a tutti gli stati, in casi pratici, ci si concentra nel calcolare la distribuzione marginale a posteriori. Seguendo questo principio, l'obiettivo dei BF é quello di calcolare la seguente distribuzione a posteriori:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \quad (2.10)$$

Le equazioni fondamentali da risolvere per raggiungere questo obiettivo sono [5]:

1. *Inizializzazione*: La ricorsione parte dalla distribuzione a priori:

$$p(\mathbf{x}_0) \quad (2.11)$$

2. *Predizione*:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int_{-\infty}^{-\infty} p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1} . \quad (2.12)$$

3. *Aggiornamento*:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int_{-\infty}^{-\infty} p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})d\mathbf{x}_t} . \quad (2.13)$$



## 2.3 Filtri di Kalman

L'implementazione dei BF richiede la conoscenza analitica dei modelli descritti sopra. I KF rappresentano una soluzione in forma chiusa delle equazioni dei BF in cui sia la dinamica del sistema, sia il modello di misura sono lineari Gaussiani. In particolare, é possibile mostrare come la distribuzione a posteriori diventa [1]:

$$\begin{aligned} \mathbf{Bel}(\mathbf{x}_t) &= p(\mathbf{x}_t | \mathbf{y}_{1:t}) = \mathcal{N}(\mathbf{x}_t | \mathbf{m}_t, \mathbf{P}_t) \\ &= \frac{1}{(2\pi)^{\frac{d}{2}} |\mathbf{P}_t|^{\frac{1}{2}}} \exp \left[ -\frac{1}{2} (\mathbf{x}_t - \mathbf{m}_t)^T \mathbf{P}_t^{-1} (\mathbf{x}_t - \mathbf{m}_t) \right]. \end{aligned} \quad (2.14)$$

Dove:

- $d$ : dimensione dello stato del sistema
- $\mathbf{x}_t$ : stato del sistema all'istante  $t$
- $\mathbf{m}_t$ : valor medio della stima dello stato all'istante  $t$  (momento del primo ordine)
- $\mathbf{P}_t$ : varianza della stima dello stato all'istante  $t$  (momento del secondo ordine).
- $|\mathbf{P}_t|$ : determinante della matrice di varianza della stima dello stato all'istante  $t$ .

Per ottenere l'espressione (2.14), si assume che i modelli di mobilità e delle misure siano Gaussiani. Con riferimento al generico istante temporale  $t$  sono esprimibili nella forma [5]:

$$\begin{aligned} \mathbf{x}_t &= \mathbf{A}_{t-1} \mathbf{x}_{t-1} + \mathbf{q}_{t-1} \\ \mathbf{y}_t &= \mathbf{H}_t \mathbf{x}_t + \mathbf{r}_t. \end{aligned} \quad (2.15)$$

Con:

- $\mathbf{q}_{t-1}$ : rumore di processo gaussiano, pari a:

$$\mathbf{q}_{t-1} \sim \mathcal{N}(0, \mathbf{Q}_{t-1}) \quad (2.16)$$

- $\mathbf{r}_t$ : rumore di misura gaussiano, pari a:

$$\mathbf{r}_t \sim \mathcal{N}(0, \mathbf{R}_t) \quad (2.17)$$

- $\mathbf{A}_{t-1}$ : matrice di transizione della dinamica del sistema
- $\mathbf{H}_t$ : matrice del modello di misura.

In termini probabilistici (2.3) diventa:

$$\begin{aligned} p(\mathbf{x}_0) &\sim \mathcal{N}(\mathbf{m}_0, \mathbf{p}_0) \\ p(\mathbf{x}_t | \mathbf{x}_{t-1}) &\sim \mathcal{N}(\mathbf{x}_t | \mathbf{A}_{t-1} \mathbf{x}_{t-1}, \mathbf{Q}_{t-1}) \\ p(\mathbf{y}_t | \mathbf{x}_t) &\sim \mathcal{N}(\mathbf{y}_t | \mathbf{H}_t \mathbf{x}_t, \mathbf{R}_t) . \end{aligned} \quad (2.18)$$

Le fasi precedentemente definite ovvero (2.12) e (2.13) diventano [5]:

1. *Predizione*

$$\begin{aligned} \mathbf{m}_t^- &= \mathbf{A}_{t-1} \mathbf{m}_{t-1} \\ \mathbf{P}_t^- &= \mathbf{A}_{t-1} \mathbf{P}_{t-1} \mathbf{A}_{t-1}^T + \mathbf{Q}_{t-1} . \end{aligned} \quad (2.19)$$

2. *Aggiornamento*

$$\begin{aligned} \mathbf{v}_t &= \mathbf{y}_t - \mathbf{H}_t \mathbf{m}_t^- \\ \mathbf{S}_t &= \mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{R}_t \\ \mathbf{K}_t &= \mathbf{P}_t^- \mathbf{H}_t^T \mathbf{S}_t^{-1} \\ \mathbf{m}_t &= \mathbf{m}_t^- + \mathbf{K}_t \mathbf{v}_t \\ \mathbf{P}_t &= \mathbf{P}_t^- - \mathbf{K}_t \mathbf{S}_t \mathbf{K}_t^T . \end{aligned} \quad (2.20)$$

Dove:

- $\mathbf{v}_t$ : innovazione, ovvero la differenza tra la misura reale effettuata al tempo  $t$  e il modello di osservazione in base alla belief dell'istante precedente.
- $\mathbf{K}_t$ : guadagno di Kalman. A seconda del suo valore può dare un'idea se la misura all'istante  $t$  è affidabile e quindi va considerata nell'aggiornamento della stima dello stato, o meno.

I KF si dimostrano ottimi stimatori. Il loro grande pregio è la loro bassa complessità computazionale. Questo vantaggio però si paga in un ristretto numero di rappresentazioni in quanto essi possono formalizzare analiticamente solo distribuzioni unimodali. Perciò risultano ottimali solo quando l'incertezza sullo stato non risulti troppo elevata, o quando si hanno a disposizione sensori precisi. Un altro svantaggio risiede nell'approssimazione Gaussiana. Infatti, a volte risulterebbe più appropriata un'approssimazione multi-modale per descrivere lo stato di un sistema. In questi casi, i PF risultano degli stimatori migliori in grado di approssimare distribuzioni qualsiasi.

### 2.3.1 Extended Kalman Filter

In molte applicazioni pratiche ci si può trovare nella condizione di avere i modelli di misura e di mobilità del sistema non lineari, pertanto i KF non risultano appropriati. In questi casi, é possibile linearizzare i modelli tramite serie di Taylor e ricondursi al caso di KF. Proprio per questo motivo si parla di EKF per riferirsi all'estensione dei KF al caso di problemi di filtraggio di tipo non-lineare. I modelli di mobilità e dell'osservazione diventano quindi [5]:

$$\begin{aligned}\mathbf{x}_t &= \mathbf{f}(\mathbf{x}_{t-1}) + \mathbf{q}_{t-1} \\ \mathbf{y}_t &= \mathbf{h}(\mathbf{x}_t) + \mathbf{r}_t .\end{aligned}\tag{2.21}$$

Con:

- $\mathbf{f}(\mathbf{x}_{t-1})$ : funzione della dinamica del sistema
- $\mathbf{h}(\mathbf{x}_t)$ : funzione del modello di misura.

In termini probabilistici si traducono in:

$$\begin{aligned}p(\mathbf{x}_t|\mathbf{x}_{t-1}) &\sim \mathcal{N}(\mathbf{x}_t|\mathbf{f}(\mathbf{x}_{t-1}), \mathbf{Q}_{t-1}) \\ p(\mathbf{y}_t|\mathbf{x}_t) &\sim \mathcal{N}(\mathbf{y}_t|\mathbf{h}(\mathbf{x}_t), \mathbf{R}_t) .\end{aligned}\tag{2.22}$$

L'obbiettivo del EKF é assumere che anche la PDF a posteriori sia ancora Gaussiana, ovvero:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \sim \mathcal{N}(\mathbf{x}_t|\mathbf{m}_t, \mathbf{P}_t) .\tag{2.23}$$

Le fasi per ricavare tale PDF a posteriori sono [5]:

1. *Predizione*

$$\begin{aligned}\mathbf{m}_t^- &= \mathbf{f}(\mathbf{m}_{t-1}) \\ \mathbf{P}_t^- &= \mathbf{F}_x(\mathbf{m}_{t-1})\mathbf{P}_{t-1}\mathbf{F}_x^T(\mathbf{m}_{t-1}) + \mathbf{Q}_{t-1}\end{aligned}\tag{2.24}$$

2. *Aggiornamento*

$$\begin{aligned}\mathbf{v}_t &= \mathbf{y}_t - \mathbf{h}(\mathbf{m}_t^-) \\ \mathbf{S}_t &= \mathbf{H}_x(\mathbf{m}_t^-)\mathbf{P}_t^-\mathbf{H}_x^T(\mathbf{m}_t^-) + \mathbf{R}_t \\ \mathbf{K}_t &= \mathbf{P}_t^-\mathbf{H}_x^T(\mathbf{m}_t^-)\mathbf{S}_t^{-1} \\ \mathbf{m}_t &= \mathbf{m}_t^- + \mathbf{K}_t\mathbf{v}_t \\ \mathbf{P}_t &= \mathbf{P}_t^- - \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^T .\end{aligned}\tag{2.25}$$

Seguendo questi passaggi e opportune semplificazioni é possibile ottenere:

$$\mathbf{m}_t = \mathbf{m}_t^- + \mathbf{P}_t^-\mathbf{H}_x^T(\mathbf{H}_x\mathbf{P}_t^-\mathbf{H}_x^T + \mathbf{R}_t)^{-1} [\mathbf{y}_t - \mathbf{h}(\mathbf{m}_t^-)]\tag{2.26}$$

$$\mathbf{P}_t = \mathbf{P}_t^- + \mathbf{P}_t^-\mathbf{H}_x^T(\mathbf{H}_x\mathbf{P}_t^-\mathbf{H}_x^T + \mathbf{R}_t)^{-1} \mathbf{H}_x\mathbf{P}_t^-\tag{2.27}$$

Com'era facile intuire le formule assomigliano in tutto e per tutto alle analoghe per i KF standard, con la differenza che le funzioni  $\mathbf{f}(\mathbf{x}_{t-1})$  e  $\mathbf{h}(\mathbf{x}_t)$  vengono linearizzate secondo Taylor. Il vantaggio degli EKF é ovviamente quello di poter rappresentare un numero maggiore di casi reali. Il processo di linearizzazione però non é possibile con tutti i tipi di funzione (esempio le funzioni gradino), infatti richiede che il modello di mobilità e dell'osservazione siano funzioni differenziabili.

## 2.4 Particle Filter

Ci sono casi in cui le distribuzioni risultano multimodali oppure discrete. In tale situazione i KF e gli EKF non possono essere utilizzati. I PF risultano essere un'ottima soluzione. Essi rappresentano la formula (2.1) con un set di campioni:

$$\mathbf{Bel}(x_t) \sim \left[ \langle x_t^1, w_t^1 \rangle, \dots, \langle x_t^{N_{\text{landmark}}}, w_t^{N_{\text{landmark}}} \rangle \right] . \quad (2.28)$$

Dove:

- $w_t^i$ : pesi non negativi chiamati *importance factor*

I PF realizzano i BF attraverso un procedimento chiamato *sequential importance sampling with resampling*. Hanno il vantaggio di poter rappresentare qualsiasi PDF. Inoltre a differenza dei KF possono far convergere sistemi non lineari. A confronto invece con algoritmi GB sono ancora piú efficienti perché automaticamente centrano le proprie risorse nella regione dello spazio degli stati con piú probabilità. Siccome i PF dipendono molto dal numero di campioni usati, i maggiori miglioramenti sono stati atti a massimizzare il numero di campioni che si ha a disposizione. Il loro svantaggio principale é che la loro complessità computazionale cresce esponenzialmente con le dimensioni in gioco. C'è quindi da fare attenzione in casi di stima multidimensionale [1].

## 2.5 Occupancy Grid

Quando la posizione e l'orientamento del robot vengono ipotizzati noti, si parla semplicemente di mapping. Il problema del mapping può essere risolto come nella sezione precedente ricorrendo ai KF. In alcuni casi, tuttavia, non è possibile conoscere a priori i landmark presenti nell'ambiente (es: i sensori ambientali) o la loro forma. In queste circostanze, si ricorre ad algoritmi a griglia in cui lo spazio viene suddiviso in celle di cui si cerca di conoscere lo stato di occupazione (cella piena o vuota?). Analizzando il caso bidimensionale, per ogni possibile cella viene effettuato un rapporto tra la probabilità che essa sia piena e il suo reciproco. Questo rapporto è chiamato *Odds* ed è definito come [7]:

$$\begin{aligned} \frac{p(m_i|\mathbf{z}_{1:t}, \mathbf{s}_{1:t})}{p(\bar{m}_i|\mathbf{z}_{1:t}, \mathbf{s}_{1:t})} &= \frac{p(m_i|\mathbf{z}_{1:t}, \mathbf{s}_{1:t})}{1 - p(m_i|\mathbf{z}_{1:t}, \mathbf{s}_{1:t})} \\ &= \frac{p(m_i|\mathbf{z}_t, \mathbf{s}_t)}{1 - p(m_i|\mathbf{z}_t, \mathbf{s}_t)} \frac{1 - p(m_i)}{p(m_i)} \frac{p(m_i|\mathbf{z}_{1:t-1}, \mathbf{s}_{1:t-1})}{1 - p(m_i|\mathbf{z}_{1:t-1}, \mathbf{s}_{1:t-1})}. \end{aligned} \quad (2.29)$$

Dove:

- $m_i$ : generico elemento della mappa  $m$ .
- $\mathbf{z}_{1:t}$ : misure effettuate per  $t$  istanti di tempo.
- $\mathbf{s}_{1:t}$ : posizioni generiche del radar.

Nel caso in cui le posizioni del radar siano note, la formula precedente diventa:

$$\frac{\mathbf{p}(m_i|\mathbf{z}_{1:t})}{\mathbf{p}(\bar{m}_i|\mathbf{z}_{1:t})} = \frac{p(m_i|\mathbf{z}_t)}{1 - p(m_i|\mathbf{z}_t)} \frac{1 - p(m_i)}{p(m_i)} \frac{p(m_i|\mathbf{z}_{1:t-1})}{1 - p(m_i|\mathbf{z}_{1:t-1})}. \quad (2.30)$$

Il significato letterale della formula è che la possibilità che la cella  $m_i$  sia piena dopo  $t$  istanti di misurazione è pari alla moltiplicazione di tre fattori:

1. Il primo termine rappresenta la probabilità che la cella  $m_i$  sia piena data la misura all'istante  $t$ , diviso la probabilità che la cella  $m_i$  sia vuota nel medesimo istante.
2. PDF a priori che la cella  $m_i$  sia piena fratto la PDF a priori che la cella  $m_i$  sia vuota.
3. (2.30) calcolata per la cella  $m_i$  nell'istante precedente  $t - 1$  perché di fatto che la (2.30) è una formula ricorsiva.

La versione logaritmica della (2.30) risulta piú facile da trattare e si calcola come:

$$\begin{aligned} \log \frac{p(m_i | \mathbf{z}_{1:t}, \mathbf{s}_{1:t})}{p(\bar{m}_i | \mathbf{z}_{1:t}, \mathbf{s}_{1:t})} &= \log \frac{p(m_i | \mathbf{z}_{1:t}, \mathbf{s}_{1:t})}{1 - p(m_i | \mathbf{z}_{1:t}, \mathbf{s}_{1:t})} \\ &= \log \frac{p(m_i | \mathbf{z}_t, \mathbf{s}_t)}{1 - p(m_i | \mathbf{z}_t, \mathbf{s}_t)} + \log \frac{1 - p(m_i)}{p(m_i)} + \\ &\quad + \log \frac{p(m_i | \mathbf{z}_{1:t-1}, \mathbf{s}_{1:t-1})}{1 - p(m_i | \mathbf{z}_{1:t-1}, \mathbf{s}_{1:t-1})} . \end{aligned} \quad (2.31)$$

Quest'ultima espressione é molto importante perché risulta notevolmente piú veloce a livello computazionale, ed in piú risolve problemi numerici quando i valori di probabilità diventano troppo piccoli. I grandi vantaggi di tale algoritmo sono l'estrema robustezza e la facilitá di implementazione. Il difetto principale é che tale modello non tiene conto della correlazione tra le celle, supponendole indipendenti. Tale assunzioni, che puó risultare veritiera nel caso del laser data la sua alte precisione, nel caso di utilizzo di antenne risulta una forte approssimazione che non tiene conto delle correlazioni che si vanno a formare dato il diagramma di radiazione. Altro difetto é che l'accuratezza dipende dalla risoluzione della griglia, che tanto piú piccola é la cella tanto piú si é precisi ma questo provoca un peso computazionale notevole [7].

## 2.6 Simultaneous Localization And Mapping

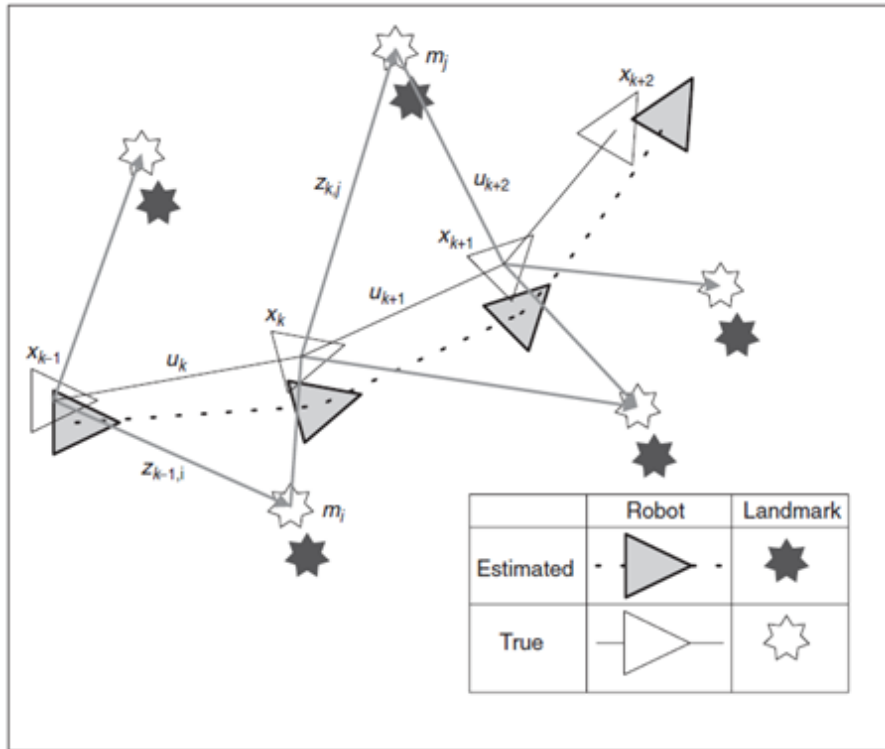


Figura 2.1: Esempio di robot che si muove all'interno di un ambiente sconosciuto [3]

Il SLAM é un problema nel quale un robot cerca di ricostruire la mappa dell'ambiente circostante, e allo stesso tempo di dedurre la sua posizione ed orientamento all'interno di esso. Queste due incognite possono essere ricavate in tempo reale senza bisogno di conoscenza a priori. Nel corso degli anni sono state proposte molte soluzioni per il problema dello SLAM, tra cui la piú importante é quella che prevede l'utilizzo dei KF e degli EKF in quanto forniscono direttamente una soluzione ricorsiva per il problema della navigazione e mezzi di calcolo per la stima dell'incertezza di posizione per il robot e i sensori ambientali.

Considerando un robot che si muove nell'ambiente, capace di raccogliere misure relative a landmark sconosciuti, é possibile definire all'istante  $t$ -esimo le seguenti quantità [3]:

- $\mathbf{x}_t$ : vettore dello stato del sistema, descrive la posizione e l'orientamento del veicolo.
- $\mathbf{u}_t$ : vettore dei segnali di controllo applicati al tempo  $t-1$  per muovere il veicolo all'istante  $t$



- $m_i$ : posizioni dell' $i$ -esimo landmark
- $\mathbf{z}_t$ : vettore dell'osservazione presa dal robot.

Prendendo tali vettori per tutti i possibili istanti temporali si ottiene:

$$\begin{aligned}\mathbf{X}_{0:t} &= \{x_0, \dots, x_t\} \\ \mathbf{U}_{0:t} &= \{u_0, \dots, u_t\} \\ \mathbf{m} &= \{m_1, \dots, m_n\} \\ \mathbf{Z}_{0:t} &= \{z_0, \dots, z_t\} .\end{aligned}\tag{2.32}$$

Dove:

- $\mathbf{X}_{0:t}$ : storia delle pose del veicolo.
- $\mathbf{U}_{0:t}$ : storia dei segnali di controllo.
- $\mathbf{m}$ : insieme di tutti i landmark della mappa ambientale.
- $\mathbf{Z}_{0:t}$ : insieme di tutte le misure fino al tempo  $t$ .

In termini probabilistici, risolvere il problema dello SLAM richiede il calcolo della seguente PDF:

$$\mathbf{p}(\mathbf{x}_t, \mathbf{m} | \mathbf{Z}_{0:t}, \mathbf{U}_{0:t}, \mathbf{x}_0)\tag{2.33}$$

per tutti i possibili istanti temporali. Questa PDF descrive la posizione del robot e dei landmark della mappa, data la storia delle misure e dei segnali di controllo. Basandosi sui KF, il calcolo di (??) prevede la risoluzione delle due fasi [3]:

1. *Predizione*

$$\mathbf{p}(\mathbf{x}_t, \mathbf{m} | \mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t}, \mathbf{x}_0) = \int_{-\infty}^{\infty} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1}, \mathbf{m} | \mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t}, \mathbf{x}_0) d\mathbf{x}_{t-1}\tag{2.34}$$

2. *Aggiornamento*

$$\mathbf{p}(\mathbf{x}_t, \mathbf{m} | \mathbf{Z}_{0:t}, \mathbf{U}_{0:t}, \mathbf{x}_0) = \frac{p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) p(\mathbf{x}_t, \mathbf{m} | \mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t}, \mathbf{x}_0)}{p(\mathbf{z}_t | \mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t})} .\tag{2.35}$$

In molti campi pratici i dati arrivano real time e sono di tipi differenti. Generalmente possono essere suddivisi in due categorie:

- $\mathbf{y}_t$ : misurazioni del mobile.
- $\mathbf{u}_t$ : segnali di controllo.

Nel caso della presenza di segnali di controllo, (2.21) si trasformano in:

$$\begin{aligned}\mathbf{x}_t &= \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_t) + \mathbf{q}_{t-1} \\ \mathbf{y}_t &= \mathbf{h}(\mathbf{x}_{t-1}) + \mathbf{r}_t .\end{aligned}\tag{2.36}$$

In termini probabilistici si traducono in:

$$\begin{aligned}p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t) &\sim \mathcal{N}(\mathbf{x}_t|\mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_t), \mathbf{Q}_{t-1}) \\ p(\mathbf{y}_t|\mathbf{x}_t) &\sim \mathcal{N}(\mathbf{y}_t|\mathbf{h}(\mathbf{x}_t), \mathbf{R}_t) .\end{aligned}\tag{2.37}$$

Una importante caratteristica di questo tipo di problema é dato dalla correlazione degli errori di misura. Analizzando la (Fig. 2.1) si nota come un errore di stima della posizione di uno qualsiasi dei landmark della mappa, sia in comune con altri simili. In pratica questo significa che la distanza relativa, ovvero rispetto al sistema di riferimento dato dal robot in un certo istante, tra due sensori  $d(m_i - m_j)$  puó essere calcolata con estrema cura anche se la posizione assoluta di questi ultimi, ovvero rispetto ad un sistema di riferimento esterno statico, é completamente sconosciuta. In termini probabilistici questo significa che la PDF congiunta  $p(m_i, m_j)$  puó avere un picco anche se la  $p(m_i)$  o la  $p(m_j)$  risultano piatte. Tale correlazione cresce monotonicamente man mano che le misure vengono effettuate, portando il problema a convergere sempre piú al crescere delle misurazioni ambientali. Supponendo che il robot compia molte altre misure spostandosi all'interno dell'ambiente:

In questa immagine si nota come dopo un numero altissimo di misurazioni la correlazione collega la posizione relativa di ogni cella all'altra. La qualità della correlazione, e quindi della ricostruzione della mappa, dipende da:

1. La qualità del sensore posto sul robot
2. La distribuzione dei landmark dell'ambiente all'interno dello scenario. Piú risultano distanti tra loro e meglio é.

Riassumendo, i risultati ottenibili dalla correlazione tra i landmark al crescere del numero di osservazioni effettuate dal robot sono [4]:

- Il determinante di qualsiasi sottomatrice della Matrice di Covarianza (CM) della mappa, che tiene conto della correlazione di ogni cella con tutte le altre, decresce monotonicamente, riducendo l'aleatorietà.
- Le stime delle posizioni dei vari sensori diventano maggiormente correlate.
- La varianza stimata dei singoli landmark dell'ambiente collassa alla prima varianza stimata dal mobile. Questa é una sorta di limite inferiore di precisione.

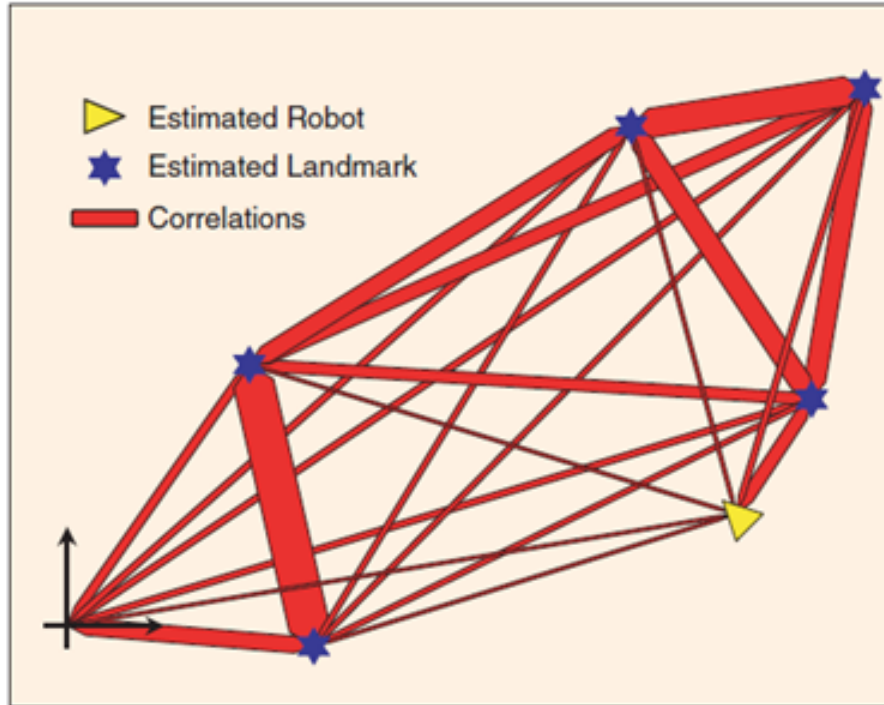


Figura 2.2: Esempio di robot che si muove all'interno di un ambiente sconosciuto dopo tante iterazioni dell'algoritmo di SLAM [3]

L'intero problema dello SLAM dipende dal mantenimento completo della conoscenza delle cross-correlazioni tra i landmark. Man mano che il mobile attraversa l'ambiente, gli errori nella stima di ogni coppia di sensori diventano maggiormente correlati, fino a raggiungere la correlazione completa. Questo significa che data l'esatta localizzazione di un qualsiasi landmark dell'ambiente, e quindi avendo una prima varianza nulla, è possibile determinare la posizione di tutti gli altri con assoluta certezza. Purtroppo tale sistema è molto costoso a livello computazionale, in più si basa su una linearizzazione di un sistema non lineare. Se tale approssimazione risulta molto forte questo metodo diventa totalmente inutilizzabile.



## Capitolo 3

# Formulazione di un nuovo modello probabilistico per il mapping

### 3.1 Definizione del problema

Al giorno d'oggi una soluzione accurata per il problema dello SLAM richiede sia un'alta definizione di stima della distanza, o *ranging*, unita a un fascio di onde estremamente stretto in modo da migliorare la risoluzione angolare. Nello SLAM si usa tipicamente il laser per il suo alto livello di precisione. Tale tecnologia risulta però difficilmente integrabile per vari motivi:

- Alti consumi energetici.
- Problemi di stima in condizioni di scarsa visibilità.
- Bassa possibilità di miniaturizzazione che rende impossibile l'implementazione in piccoli dispositivi elettronici, come ad esempio gli smartphone

Il mondo delle onde radio può sostituire tale tecnologia: guardando verso le tecnologie di nuova generazione (5G), ovvero le mmW ad altissima frequenza, per esempio lavoranti a 60 GHz, si avrà una grande porzione di spettro inutilizzata da poter sfruttare, con una tecnologia che possiede le caratteristiche richieste con in più una possibilità di miniaturizzazione molto maggiore. L'idea di sostituire la tecnologia laser con quella delle mmW non è nuova come idea, ed è stata utilizzata in passato per l'ambiente outdoor. Purtroppo però anche usando un numero di antenne altissimo, le prestazioni del fascio laser risultano impareggiabili, questo perché data un'antenna essa presenterà diversi lobi. La misura quindi sarà influenzata maggiormente sia dal cammino del lobo principale, ma anche da tutti i contributi provenienti da una direzione diversa per via dei lobi secondari.

Nonostante questa grossa limitazione, la soluzione delle mmW per il problema dello SLAM, se adeguatamente progettata, può risultare estremamente efficace.

Considerando di suddividere l'ambiente in una griglia di  $N$  celle, è possibile definire il vettore che rappresenta la mappa che si desidera stimare, chiamato *vettore di stato* [2]:

$$\mathbf{m}(k) = [m_1(k), \dots, m_N(k)]^T \quad (3.1)$$

Graficamente:

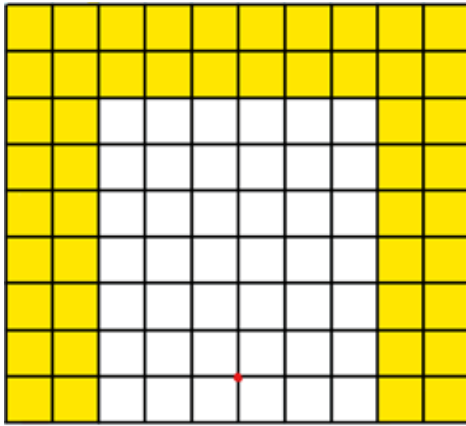


Figura 3.1: Esempio di mappa: le celle colorate di giallo sono considerate piene, mentre quelle lasciate bianche sono da considerarsi vuote. Il marker rosso indica la posizione del radar sulla mappa.

La generica cella  $i$ -esima della mappa può essere descritta come una variabile aleatoria binaria. Infatti, gli unici due valori che può assumere sono quelli di cella piena e di cella vuota. Lo scopo del radar è di ricavare un ricco numero di osservazioni per la susseguente fase di mapping scaturita dall'analisi dei segnali di scattering dell'ambiente circostante, in modo da poter determinare i valori delle celle da cui è composta la mappa. Nel seguito si assumerà per semplicità che:

- Le celle piene siano caratterizzate tutte dallo stesso valore di radar cross section (RCS) all'interno del vettore di stato.
- Le celle vuote daranno tutte valore energetico nullo.

Per una serie di direzioni privilegiate (steering) del lobo principale dell'antenna, il radar invia  $N_p$  impulsi e riceve il segnale riflesso dall'ambiente. Dal segnale ricevuto, viene calcolato il profilo di energia. Il profilo di energia viene suddiviso in  $N_{\text{bins}}$  bin di energia di durata  $T_{\text{ED}}$ . Questo processo viene poi ripetuto per tutte le  $N_{\text{steer}}$  direzioni di steering.

Si definiscono le seguenti quantità [2]:

- $T_f$ : tempo di frame ovvero il tempo che intercorre tra l'invio di un impulso e il successivo.
- $W$ : banda del segnale in trasmissione.
- $T_{ED}$ : durata temporale o *energy bin*, pari a:

$$T_{ED} = \frac{1}{W} \quad (3.2)$$

Trasformando il tempo in spazio, si ottiene la distanza tra due *energy bin*:

$$d_{bin} = \frac{cT_{ED}}{2} . \quad (3.3)$$

Questa é l'unitá di misura fondamentale per catalogare la distanza delle varie celle dalla posizione del mobile.

- $N_{bins}$ : Numero di *bin di energia* pari a:

$$N_{bins} = \frac{T_f}{T_{ED}} . \quad (3.4)$$

- $N_{steer}$ : numero di direzioni di steering dell'antenna. É possibile infatti che si lavori con antenne aventi un certo numero di direzioni privilegiate, dette di *steering*, con un guadagno del lobo principale piú elevato. L'espressione della generica posizione di steering é :

$$\theta_b = -\frac{\pi}{2} \left( \frac{N_{steer-1}}{N_{steer}} \right) + \frac{(b-1)\pi}{N_{steer}} \quad (3.5)$$

$$b = [1, \dots, N_{steer}]$$

- $N_p$ : numero di impulsi, con espressione per il segnale completo pari a:

$$g(t) = \sum_{i=0}^{N_p-1} p(t - iT_f) \quad (3.6)$$

- $N_{obs}$ : numero di osservazioni, pari a:

$$N_{obs} = N_{steer} N_{bins} . \quad (3.7)$$

Per ogni possibile direzione privilegiata dell'antenna si misureranno tutti i possibili  $N_{bins}$ , che daranno una serie di valori energetici. Questa soluzione é molto importante in quanto antenne molto direttive permettono di assegnare alti valori di guadagno solo alle pochissime celle che ricadranno dentro al beam principale dell'antenna, che data

la sua direttività, sarà stretto (di pochi gradi), in modo da escludere i valori delle celle interferenti, ovvero celle che si trovano in direzioni diverse da quella di steering, aumentando notevolmente la precisione della misura.

- $T_{\text{scan}}$ : periodo di scansione, pari a:

$$\begin{aligned} T_{\text{scan}} &= N_{\text{steer}} N_p T_f \\ &= N_{\text{steer}} N_p N_{\text{bins}} T_{\text{ED}} . \end{aligned} \tag{3.8}$$

Nella figura sottostante viene mostrato come ad ogni bin temporale, fissata la direzione di steering, corrispondano tutti i punti di una circonferenza.:

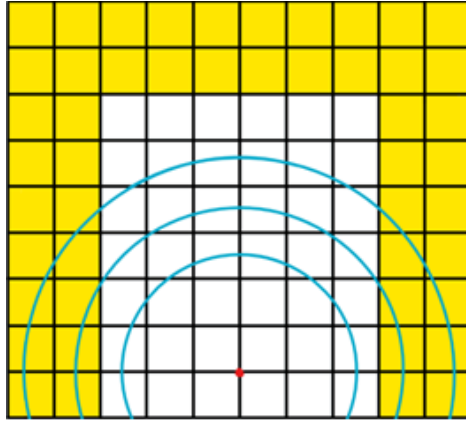


Figura 3.2: Fase di associazione delle celle ai vari bin temporali

Ogni circonferenza intercetta le celle che si trovano alla stessa distanza che saranno quindi associate allo stesso bin temporale. La differenza spaziale tra i cerchi é data dalla formula (3.3) e di conseguenza dalla banda del segnale utilizzato: piú la banda é grande, minore é la distanza che separa due bin consecutivi, incrementando la precisione della ranging. Le celle a seconda della loro distanza dal mobile porteranno un contributo energetico, positivo o nullo, al proprio bin.



Analizzando ad esempio la circonferenza di mezzo nel caso di singolo impulso:

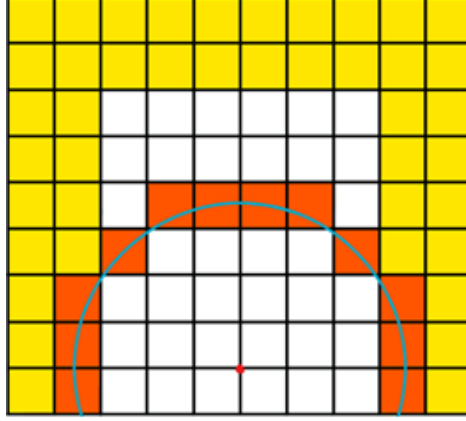


Figura 3.3: Esempio di celle interferenti su una stessa circonferenza, e quindi su uno stesso bin

il valore energetico del bin associato alla circonferenza evidenziata, sarà quindi la somma energetica di:

- 6 celle piene, ovvero quelle piú esterne
- 6 celle vuote, quelle piú interne.

Questo bin sarà misurato  $N_{\text{steer}}$  volte, in modo che la direttività dell'antenna vada ad influire sulle misurazioni energetiche. [2] Risulterà quindi, per ogni singolo impulso, un vettore di misure:

$$\begin{aligned} \mathbf{e}(k) &= [e_{11}(k), \dots, e_{N_{\text{steer}}N_{\text{bins}}}(k)] \\ &= [e_{11}(k), \dots, e_{N_{\text{bins}}1}, e_{12}(k), \dots, e_{N_{\text{steer}}N_{\text{bins}}}(k)] . \end{aligned} \quad (3.9)$$

Si costruirá ora un modello di osservazione gaussiano da cui poter stimare (3.1):

$$\begin{aligned} \mathbf{z}(k) &= [z_{11}(k), \dots, z_{N_{\text{steer}}N_{\text{bins}}}(k)] \\ &= [z_{11}(k), \dots, z_{N_{\text{bins}}1}, z_{12}(k), \dots, z_{N_{\text{steer}}N_{\text{bins}}}(k)] \\ &= \mathbf{h}(\mathbf{m}(k)) + \mathbf{v}(k) . \end{aligned} \quad (3.10)$$

Dove:

- $z_{bs}(k)$ : energia dell'osservazione, corrispondente al bin  $s$  e alla direzione di steering  $b$ .
- $\mathbf{h}(\cdot)$ : funzione di osservazione che relaziona il vettore di stato con le osservazioni.

Ogni misura sar  sporcata da errore gaussiano:

$$\begin{aligned} \mathbf{v}(k) &= [v_{11}(k), \dots, v_{N_{\text{steer}}N_{\text{bins}}}(k)] \\ &= [v_{11}(k), \dots, v_{N_{\text{bins}}1}, v_{12}(k), \dots, v_{N_{\text{steer}}N_{\text{bins}}}(k)] , \end{aligned} \quad (3.11)$$

a valor medio nullo e CM:

$$\mathbf{R} = \begin{bmatrix} \sigma_{11}^2 & 0 & \dots & 0 & 0 \\ 0 & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & 0 \\ 0 & 0 & \dots & 0 & \sigma_{N_{\text{obs}}N_{\text{obs}}}^2 \end{bmatrix} \quad (3.12)$$

Un qualsiasi valore sulla diagonale principale ha espressione:

$$\sigma_{ii}^2 = N_{\text{ext}0} * (N_{\text{p}} \cdot W * T_{\text{ED}} * N0 + 2e_{bs}) . \quad (3.13)$$

Il generico elemento  $z_{bs}(k)$  pu  essere scritto:

$$\begin{aligned} z_{bs}(k) &= h_{bs}(\mathbf{m}(k)) + v_{bs}(k) \\ &= \mathbb{E}[e_{bs}(k)] + v_{bs}(k) . \end{aligned} \quad (3.14)$$

Dove:

$$\mathbb{E}[e_{bs}(k)] = T_{\text{f}} \cdot P_{bs}(k) \quad (3.15)$$

Il termine  $P_{bs}(k)$  dipende dal vettore di stato e rappresenta la potenza media ricevuta al bin  $s$  per la  $b$ -esima direzione di steering.

$$P_{bs}(k) = P_{\omega} + \sum_{i \in \mathcal{R}(s)} \int_W \frac{P_t(f) c^2 m_i(k) G^2(\theta_i - \theta_b, f) N_{\text{p}}}{f^2 (4\pi)^3 d_i^4} \quad (3.16)$$

Dove:

- $P_{\omega}$ : potenza di rumore, pari a:

$$P_{\omega} = \frac{\sigma^2 T_{\text{ED}} N_{\text{p}}}{T_{\text{f}}} . \quad (3.17)$$

- $P_t(f)$ : densit  spettrale di potenza di trasmissione.
- $d_i$ : distanza tra il mobile e la cella  $i$ -esima.
- $\theta_i$ : angolo tra il mobile e la cella  $i$ -esima.

### 3.2 Modello probabilistico adottato

Definiamo il set di misure fino all'istante  $t$ :

$$\mathbf{z}_{1:t} = [\mathbf{z}(1), \dots, \mathbf{z}(t)] , \quad (3.18)$$

Si vuole calcolare la probabilità di occupazione della cella  $i$ -esima al  $t$ -esimo istante temporale. Questa cella porterá un contributo energetico, se occupata, riferito ad un bin nel vettore delle misure che dipende dalla distanza a cui si trova rispetto al radar. A tale bin contribuiranno anche altre celle, che verranno chiamate interferenti. L'insieme degli indici delle celle che si trovano alla stessa distanza verrà indicato come  $\mathcal{R}(s)$  dove  $s = |d/d_{\text{bin}}|$  con  $d$  distanza tra la cella e il radar. Tale insieme avrà cardinalità  $L$ . Assumendo per semplicitá che la probabilità di occupazione di una cella sia indipendente dalle altre celle, ci si concentrerá nel calcolo della PDF a posteriori della singola cella  $i$ -esima, ovvero:

$$\begin{aligned} \text{Bel}(m_i^t) &= p(m_i^t | \mathbf{z}_{1:t}) \\ &= \sum_{j=1}^{N_{\text{comb}}} p(m_i | m_j, \mathbf{z}_{1:t}) p(m_j) \\ &\approx \sum_{j=1}^{N_{\text{comb}}} p(m_i | m_j, \mathbf{z}_{1:t}) \text{Bel}(m_j^{t-1}) . \end{aligned} \quad (3.19)$$

Dove  $N_{\text{comb}}$  é il numero di possibili combinazioni di stato delle celle interferenti con quella di riferimento  $i$  ed é pari a:

$$N_{\text{comb}} = 2^{N_{\text{tot}}-1} , \quad (3.20)$$

con  $N_{\text{tot}}$  pari al numero di celle intervenenti nel bin temporale in cui staziona la cella  $i$ .

Approssimato la PDF a priori delle celle interferenti con la loro belief all'istante precedente, ovvero:

$$\text{Bel}(m_j^{t-1}) = \prod_{l \in \mathcal{R}(s)} \text{Bel}(m_l^{t-1}) , \quad (3.21)$$

Applicando la regola di Bayes, si trova che:

$$p(m_i | m_j, \mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t | m_i, m_j, \mathbf{z}_{1:t-1}) p(m_i | m_j, \mathbf{z}_{1:t-1})}{p(\mathbf{z}_t | \mathbf{z}_{1:t-1})} , \quad (3.22)$$

ed applicando le semplificazioni di Markov, si ottiene:

$$p(m_i | m_j, \mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t | m_i, m_j) p(m_i | m_j, \mathbf{z}_{1:t-1})}{p(\mathbf{z}_t | \mathbf{z}_{1:t-1})} . \quad (3.23)$$

Siccome la cella puo assumere solo due valori, é possibile scrivere anche la belief per il caso negato ottenendo:

$$\begin{aligned}
 1 - p(m_i|m_j, \mathbf{z}_{1:t}) &= p(\overline{m}_i|m_j, \mathbf{z}_{1:t}) \\
 &= \frac{p(\mathbf{z}_t|\overline{m}_i, m_j, \mathbf{z}_{1:t-1})p(\overline{m}_i|m_j, \mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \\
 &= \frac{p(\mathbf{z}_t|\overline{m}_i, m_j)p(\overline{m}_i|m_j, \mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})}.
 \end{aligned} \tag{3.24}$$

Eseguendo il rapporto tra (3.23) e (3.24):

$$\begin{aligned}
 \frac{\text{Bel}(m_i^t)}{\text{Bel}(\overline{m}_i^t)} &= \frac{\text{Bel}(m_i^t)}{1 - \text{Bel}(m_i^t)} \\
 &= \frac{\sum_{j=1}^{N_{\text{comb}}} p(m_i|\mathbf{m}_j, \mathbf{z}_{1:t})\text{Bel}(\mathbf{m}_j^{t-1})}{\sum_{j=1}^{N_{\text{comb}}} p(\overline{m}_i|\mathbf{m}_j, \mathbf{z}_{1:t})\text{Bel}(\mathbf{m}_j^{t-1})} \\
 &= \frac{\sum_{j=1}^{N_{\text{comb}}} p(\mathbf{z}_t|m_i, \mathbf{m}_j, x_t)p(m_i|\mathbf{m}_j, \mathbf{z}_{1:t-1})\text{Bel}(\mathbf{m}_j^{t-1})}{\sum_{j=1}^{N_{\text{comb}}} p(\mathbf{z}_t|\overline{m}_i, \mathbf{m}_j, x_t)p(\overline{m}_i|\mathbf{m}_j, \mathbf{z}_{1:t-1})\text{Bel}(\mathbf{m}_j^{t-1})} \\
 &= \frac{\sum_{j=1}^{N_{\text{comb}}} p(\mathbf{z}_t|m_i, \mathbf{m}_j, x_t)\text{Bel}(m_i^{t-1}|\mathbf{m}_j)\text{Bel}(\mathbf{m}_j^{t-1})}{\sum_{j=1}^{N_{\text{comb}}} p(\mathbf{z}_t|\overline{m}_i, \mathbf{m}_j, x_t)\text{Bel}(\overline{m}_i^{t-1}|\mathbf{m}_j)\text{Bel}(\mathbf{m}_j^{t-1})} \\
 &\approx \frac{\sum_{j=1}^{N_{\text{comb}}} p(\mathbf{z}_t|m_i, \mathbf{m}_j, x_t)\text{Bel}(m_i^{t-1}|\mathbf{m}_j)\text{Bel}(\mathbf{m}_j^{t-1})}{\sum_{j=1}^{N_{\text{comb}}} p(\mathbf{z}_t|\overline{m}_i, \mathbf{m}_j, x_t)\text{Bel}(\overline{m}_i^{t-1}|\mathbf{m}_j)\text{Bel}(\mathbf{m}_j^{t-1})}.
 \end{aligned} \tag{3.25}$$

Invertendo la formula precedente, si arriva ad ottenere:

$$\text{Bel}(m_i^t) = \left[ 1 + \frac{\sum_{j=1}^{N_{\text{comb}}} p(\mathbf{z}_t|\overline{m}_i, \mathbf{m}_j)(1 - \text{Bel}(m_i^{t-1}|\mathbf{m}_j))\text{Bel}(\mathbf{m}_j^{t-1})}{\sum_{j=1}^{N_{\text{comb}}} p(\mathbf{z}_t|m_i, \mathbf{m}_j)\text{Bel}(m_i^{t-1}|\mathbf{m}_j)\text{Bel}(\mathbf{m}_j^{t-1})} \right]^{-1}. \tag{3.26}$$

Tale formula si divide quindi in 5 elementi, divisibili in tre gruppi:

- La PDF dell'osservazione condizionata dalla mappa in quell'istante  $p(\mathbf{z}_t|m_i, \mathbf{m}_j)$  e relativo reciproco  $p(\mathbf{z}_t|\overline{m}_i, \mathbf{m}_j)$ .
- La Belief dell'istante precedente relativa alla cella studiata  $\text{Bel}(m_i^{t-1}|\mathbf{m}_j)$  e il suo reciproco  $1 - \text{Bel}(m_i^{t-1}|\mathbf{m}_j)$
- Le informazioni a priori sulle celle interferenti ovvero la formula (3.21).

Rispetto ai modelli presenti in letteratura, (3.26) considera la dipendenza tra celle nel modello dell'osservazione tramite la PDF condizionata  $p(\mathbf{z}_t|m_i, \mathbf{m}_j)$ . Questo permette di migliorare l'accuratezza nella ricostruzione della mappa rispetto al caso di totale indipendenza tra celle.

### 3.3 Correlazione tra celle nel modello di misura

Nella (3.26) si é inserito il prodotto di tutte le belief dei vicini che tramite (3.21) risultano indipendenti tra loro. Il contributo portato dalle PDF a priori delle celle interferenti é molto importante in quanto tramite esso si ricava un'informazione aggiuntiva per il calcolo della belief della cella in esame. Supponendo di voler studiare la cella  $i$  all'istante  $t$  che contribuisce al bin numero 20 nel vettore dell'energia ricevuta ad un dato istante e fissata la direzione di steering. Allo stesso bin contribuiscono altre due celle che indicheremo con gli indici  $v$  e  $k$ . Il vettore delle misure é indicato come  $\mathbf{y}$ . All'istante precedente si aveva un valore per le Belief dei vicini pari a:

- $\text{Bel}(m_{jv}^{t-1}) = 0.1$
- $\text{Bel}(m_{jk}^{t-1}) = 0.8$

Le possibili combinazioni danno luogo a:

- se  $m_v$  e  $m_k$  sono vuote:

$$\begin{aligned} \text{Bel}(\mathbf{m}_j) &= \text{Bel}(\bar{m}_{jv}) \cdot \text{Bel}(\bar{m}_{jk}) \\ &= (1 - \text{Bel}(m_j)) \cdot (1 - \text{Bel}(m_k)) \\ &= (1 - 0.1) \cdot (1 - 0.8) = 0.18 \end{aligned} \quad (3.27)$$

- se  $m_v$  é vuota e  $m_k$  piena:

$$\begin{aligned} \text{Bel}(\mathbf{m}_j) &= \text{Bel}(\bar{m}_{jv}) \cdot \text{Bel}(m_{jk}) \\ &= (1 - \text{Bel}(m_j)) \cdot \text{Bel}(m_k) \\ &= (1 - 0.1) \cdot (0.8) = 0.72 \end{aligned} \quad (3.28)$$

- se  $m_v$  é piena e  $m_k$  vuota:

$$\begin{aligned} \text{Bel}(\mathbf{m}_j) &= \text{Bel}(m_{jv}) \cdot \text{Bel}(\bar{m}_{jk}) \\ &= \text{Bel}(m_j) \cdot (1 - \text{Bel}(m_k)) \\ &= (0.1) \cdot (1 - 0.8) = 0.02 \end{aligned} \quad (3.29)$$

- se  $m_v$  e  $m_k$  sono piene:

$$\begin{aligned} \text{Bel}(\mathbf{m}_j) &= \text{Bel}(m_{jv}) \cdot \text{Bel}(m_{jk}) \\ &= (0.1) \cdot (0.8) = 0.08 \end{aligned} \quad (3.30)$$

In base al valore delle belief calcolate agli istanti precedenti la combinazione piú probabile risulta quella in cui la cella  $j$ -esima é vuota e quella  $k$ -esima piena. Questa conoscenza a priori servirá a pesare l'informazione tratta dalla misura stessa: se dalla misura verrà dedotto un valore di occupazione diverso (quindi una delle altre combinazioni), allora l'informazione a priori ci dirá di essere cauti nel considerare la misura affidabile; al contrario, se la misura confermerá l'ipotesi ( $m_j = 0, m_k = 1$ ), allora la conoscenza a priori rafforzerá la decisione presa.

### 3.4 Esempio di calcolo dell'occupancy

Questa sezione discuterà il significato dei modelli dell'osservazione  $p(\mathbf{z}_t|m_i, \mathbf{m}_j)$  e il suo reciproco  $p(\mathbf{z}_t|m_i, \mathbf{m}_j)$ . I passi necessari sono:

1. Si ipotizzi di avere una mappa composta da  $N$  punti, un numero di bin massimi pari a  $X$  ed un mobile di riferimento provvisto di antenna. Al tempo  $t$  è in posizione nota  $(x_0, y_0)$  ed effettua una misurazione dell'ambiente. Ipotizzando che ogni cella dia un contributo:
  - $R RCS$  in caso di cella piena
  - 0 in caso di cella vuota

Si ottiene un vettore di contributi di energie del tipo (3.9) con le singole misure soggette a rumore additivo gaussiano bianco (3.11) con relativa CM (3.12). La somma di questi due vettori darà luogo a:

$$\begin{aligned} e_{bsReal} &= [E_1 + V_1, \dots, E_X + V_X] \\ &= [E_{r1}, \dots, E_{rX}] \end{aligned} \quad (3.31)$$

2. Si definisce la matrice  $RSS$ :

$$RSS = \begin{bmatrix} R_{1.1} & \dots & \dots & R_{1.X} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ R_{N.1} & \dots & \dots & R_{N.X} \end{bmatrix} \quad (3.32)$$

Tale matrice raggruppa una serie di valori che mettono in relazione l'indice delle celle (righe) con l'indice dei bin (colonne). Questi valori saranno delle stime create tramite modello matematico della possibile energia che una data cella, residente ad una certa distanza di bin, possieda nel caso in cui essa sia piena. Il fatto che non si crei una matrice analoga per il caso in cui la cella sia vuota, ipotesi del tutto logica in quanto una cella vuota non possiede ovviamente nessuna energia, porta ad un'asimmetria del problema che aiuterà enormemente la stima dell'ambiente nel suo complesso.

3. Ipotizzando di voler analizzare la cella 18 residente a distanza di bin 5 con vicini le celle 7 e 23. Utilizzando la formula (3.20) si ottiene che il numero di possibili combinazioni dei valori dei 2 vicini è 4. È possibile quindi costruire una matrice con le possibili combinazioni binarie delle celle interferenti:

$$combos = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \quad (3.33)$$

In questo esempio:

- la colonna di sinistra é riferita all'indice del vicino 7
- la colonna di destra é riferita all'indice del vicino 23

Nei valori pari a 1 vanno sostituiti i valori della matrice  $RSS$  risiedenti:

- nelle righe 7 e 23 ovvero gli indici dei vicini
- nella colonna 5 ovvero l'indice del analizzato.

Si ottiene:

$$combos = \begin{bmatrix} 0 & 0 \\ 0 & R_{23.5} \\ R_{7.5} & 0 \\ R_{7.5} & R_{23.5} \end{bmatrix} \quad (3.34)$$

La matrice  $combos$  ora va sdoppiata, in modo che lavorando separatamente sulle due copie si possano esplicitare le PDF cercate:

$$c_0 = \begin{bmatrix} 0 & 0 \\ 0 & R_{23.5} \\ R_{7.5} & 0 \\ R_{7.5} & R_{23.5} \end{bmatrix} \quad c_1 = \begin{bmatrix} 0 & 0 \\ 0 & R_{23.5} \\ R_{7.5} & 0 \\ R_{7.5} & R_{23.5} \end{bmatrix} \quad (3.35)$$

Vanno aggiunte ora delle colonne di 0 e 1 che saranno riferite alla cella studiata ovvero quella con indice 18. Tali nuove entrate andranno posizionate in modo che gli indici delle celle siano in ordine crescente da sinistra a destra:

$$c_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & R_{23.5} \\ R_{7.5} & 0 & 0 \\ R_{7.5} & 0 & R_{23.5} \end{bmatrix} \quad c_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & R_{23.5} \\ R_{7.5} & 1 & 0 \\ R_{7.5} & 1 & R_{23.5} \end{bmatrix} \quad (3.36)$$

Eseguendo la medesima sostituzione avvenuta prima, negli 1 della matrice  $c_1$  andranno posizionati i valori presi dalla matrice  $RSS$  per la cella 18:

$$c_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & R_{23.5} \\ R_{7.5} & 0 & 0 \\ R_{7.5} & 0 & R_{23.5} \end{bmatrix} \quad c_1 = \begin{bmatrix} 0 & R_{18.5} & 0 \\ 0 & R_{18.5} & R_{23.5} \\ R_{7.5} & R_{18.5} & 0 \\ R_{7.5} & R_{18.5} & R_{23.5} \end{bmatrix} \quad (3.37)$$

Sommando le matrici per righe:

$$\begin{aligned} \mathbf{z}_0 &= [0, R_{23.5}, R_{7.5}, R_{7.5} + R_{23.5}] \\ &= [Z_{01}, Z_{02}, Z_{02}, Z_{02}] \\ \mathbf{z}_1 &= [R_{18.5}, R_{18.5} + R_{23.5}, R_{7.5} + R_{18.5}, R_{7.5} + R_{18.5} + R_{23.5}] \\ &= [Z_{11}, Z_{12}, Z_{12}, Z_{12}] \end{aligned} \quad (3.38)$$

4. Da  $E_{bsReal}$  si costruisce un altro vettore lungo  $N_{combos}$  con il valore all'indice del bin di riferimento ripetuto:

$$e_{18} = [E_{r5}, E_{r5}, E_{r5}, E_{r5}] \quad (3.39)$$

Si definiscano ora le matrici:

$$i_0 = \begin{bmatrix} I0_{11} & \dots & \dots & I0_{1X} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ I0_{N_{combos}1} & \dots & \dots & I0_{N_{combos}X} \end{bmatrix} \quad i_1 = \begin{bmatrix} I1_{11} & \dots & \dots & I1_{1X} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ I1_{N_{combos}1} & \dots & \dots & I1_{N_{combos}X} \end{bmatrix} \quad (3.40)$$

Con ogni valore inizialmente pari a 0. Effettuando le differenze:

$$(e_{18} - \mathbf{z}_0)^T (e_{18} - \mathbf{z}_1)^T, \quad (3.41)$$

ed inserendole nella colonna pari al bin della cella di riferimento, ovvero il solito indice 5, si ottengono:

$$i_0 = \begin{bmatrix} 0 & \dots & 0 & E_{r5} - Z0_1 & 0 & \dots & 0 \\ 0 & \dots & 0 & E_{r5} - Z0_2 & 0 & \dots & 0 \\ 0 & \dots & 0 & E_{r5} - Z0_3 & 0 & \dots & 0 \\ 0 & \dots & 0 & E_{r5} - Z0_4 & 0 & \dots & 0 \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0 & I0_1 & 0 & \dots & 0 \\ 0 & \dots & 0 & I0_2 & 0 & \dots & 0 \\ 0 & \dots & 0 & I0_3 & 0 & \dots & 0 \\ 0 & \dots & 0 & I0_4 & 0 & \dots & 0 \end{bmatrix} \quad (3.42)$$

$$i_1 = \begin{bmatrix} 0 & \dots & 0 & E_{r5} - Z1_1 & 0 & \dots & 0 \\ 0 & \dots & 0 & E_{r5} - Z1_2 & 0 & \dots & 0 \\ 0 & \dots & 0 & E_{r5} - Z1_3 & 0 & \dots & 0 \\ 0 & \dots & 0 & E_{r5} - Z1_4 & 0 & \dots & 0 \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0 & I1_1 & 0 & \dots & 0 \\ 0 & \dots & 0 & I1_2 & 0 & \dots & 0 \\ 0 & \dots & 0 & I1_3 & 0 & \dots & 0 \\ 0 & \dots & 0 & I1_4 & 0 & \dots & 0 \end{bmatrix} \quad (3.43)$$

5. L'ultimo passaggio é la moltiplicazione vettoriale, prendendo solo una riga delle matrici  $i_0$  e  $i_1$  alla volta:

$$\begin{aligned} p0_i &= EXP \left[ [0, \dots, 0, I0_i, 0, \dots, 0] \quad R_k^{-1} \quad [0, \dots, 0, I0_i, 0, \dots, 0]^T \right] \\ p1_i &= EXP \left[ [0, \dots, 0, I1_i, 0, \dots, 0] \quad R_k^{-1} \quad [0, \dots, 0, I1_i, 0, \dots, 0]^T \right] \\ i &\in [1, \dots, N_{combos}] \end{aligned} \quad (3.44)$$

Ottenendo infine:

$$\begin{aligned} p(Z_t | \overline{m}_i, m_j, x_t) &= [p0_1, \dots, p0_{N_{combos}}] \\ p(Z_t | m_i, m_j, x_t) &= [p1_1, \dots, p1_{N_{combos}}] \end{aligned} \quad (3.45)$$



### 3.4.1 Esempio numerico

Ipotizzando di studiare la cella numero 18 piena residente al bin 5 in condivisione con le celle di indice 7 e 23 avendo i seguenti valori:

- $\text{Bel}(18^{t-1}) = 0.4$
- $p(\mathbf{z}_t | m_i, m_j, x_t) = [0.2, 0.4, 0.6, 0.8]$
- $p(\mathbf{z}_t | \bar{m}_i, m_j, x_t) = [0.1, 0.3, 0.5, 0.7]$
- $\text{Bel}([m_7, m_{23}]^T) = [0.18, 0.72, 0.02, 0.08]$

La (3.26) risulta essere:

$$\begin{aligned} \text{Bel}(m_{18}^{t-1}) &= \left[ 1 + \frac{(1 - 0.4)((0.1)(0.18) + (0.3)(0.72) + (0.5)(0.02) + (0.07)(0.08))}{(0.2)(0.18)(0.4) + (0.4)(0.72)(0.4) + (0.6)(0.02)(0.4) + (0.8)(0.08)(0.4)} \right]^{-1} \\ &\approx 0.76 \end{aligned} \tag{3.46}$$

Da questo semplice esempio si può notare come col metodo utilizzato il valore  $\text{Bel}(18)$  sia passato da 0.4 a 0.76, correggendo il valore che stava tendendo precedentemente verso l'estremo sbagliato.



# Capitolo 4

## Risultati

### 4.1 Introduzione

In questo capitolo, verranno presentati i risultati di mapping per scenari diversi, confrontando diversi tipi di antenna massive. La qualità della ricostruzione della mappa verrà valutata in relazione a quella che è il reale stato di occupazione delle celle. Con il colore blu indicativo per le celle vuote mentre il colore bordeaux le celle piene, le figure successive mostrano le mappe generate:

1. Il primo scenario rappresenta una stanza  $10 \times 10 \text{ m}^2$  con celle di area  $0.5 \times 0.5 \text{ m}^2$ . In questo caso il numero di celle vuote è superiore al numero di celle piene (60% del totale), infatti  $N_0 = 245$  e  $N_1 = 155$ .

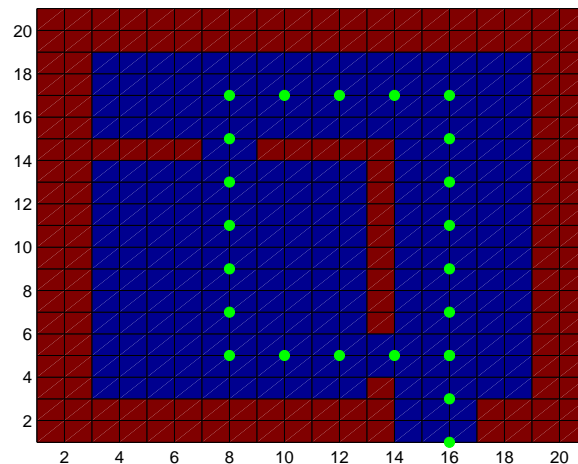


Figura 4.1: Mappa vera: Scenario 1. Ambedue gli assi indicano gli indici delle celle.

2. Il secondo scenario ha dimensioni pari a  $10 \times 15 \text{ m}^2$  con celle sempre di area pari a  $0.5 \times 0.5 \text{ m}^2$ . In questo caso il numero di celle piena e vuote piú o meno si equivale, infatti  $N_0 = 288$  e  $N_1 = 312$ .

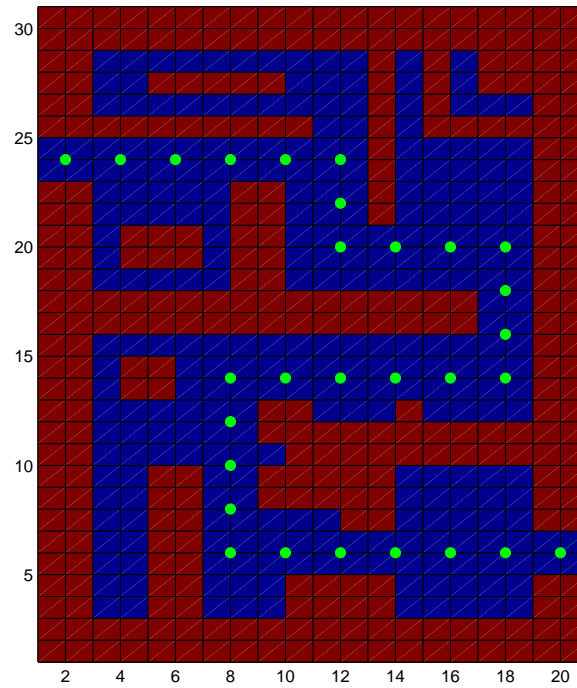


Figura 4.2: Mappa vera: Scenario 2. Ambedue gli assi indicano gli indici delle celle.

3. Il terzo scenario ha dimensioni pari a  $10 \times 15 \text{ m}^2$  con celle di  $0.5 \times 0.5 \text{ m}^2$ . Anche in questo caso il numero di celle piene piú o meno equivale quello delle celle vuote, infatti  $N_0 = 286$  e  $N_1 = 314$ . La differenza é che in questo caso le celle sono distribuite in maniera piú compatta.

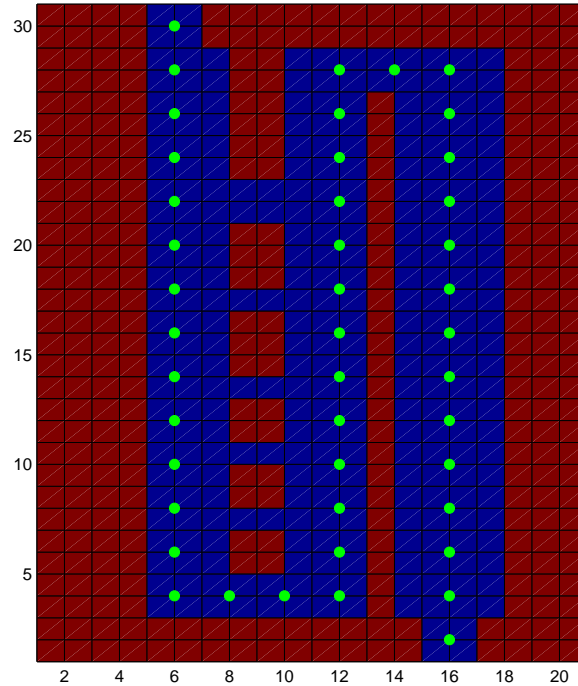


Figura 4.3: Mappa vera: Scenario 3. Ambedue gli assi indicano gli indici delle celle.

Tali mappe sono diverse sia come rapporto  $N_1/N_0$ , sia come topologia. Le simulazioni sono state effettuate ipotizzando lo scenario di un mobile in movimento che scansiona l'ambiente circostante, percorrendo la traiettoria indicata dai punti verdi. Ogni cella presenta valore pari a:

- RCS: in caso di cella piena.
- 0: in caso di cella vuota.

Le simulazioni sono state effettuate considerando errori di varia natura per emulare casi piú simili alla realtà, di seguito si riporta una lista delle principali cause di non idealità. Essi si dividono in:

1. *Mismatch fra il valore reale di RRCS e quello utilizzato dal modello*( $E_{RRCS}$ ): si somma al valore nominale di RCS delle celle piene per modellare

un mismatch tra il valore vero di RCS di ogni cella e quello ipotizzato dal modello quando suppone la cella piena. Infatti, nelle simulazioni, come già anticipato in precedenza, si é adottato un modello di RCS uguale per tutte le celle e pari a [6]:

$$\text{RCS} = 4\gamma_0^2 A_{\text{cella}} , \quad (4.1)$$

dove  $A_{\text{cella}} = 0.5 \times 0.5 \text{ m}^2$  é l'area della cella, mentre  $\gamma_0$  dipende dal materiale e risulta pari a:

$$\gamma_0 = \left| \frac{1 - \sqrt{\epsilon_r - j \epsilon_r \tan(\delta)}}{1 + \sqrt{\epsilon_r - j \epsilon_r \tan(\delta)}} \right| , \quad (4.2)$$

con  $\epsilon_r = 2.26$  e  $\tan(\delta) = 0.0491$  (cemento armato). Sostituendo questi valori si ottiene un valore di RCS pari a 0.040, e quindi  $\text{RRCS} = \sqrt{\text{RCS}} = 0.2$ . Nella realtà questa ipotesi non é mai verificata data la varietà di materiali presenti nell'ambiente. Per i risultati di questa tesi, il valore di mismatch per una qualsiasi cella:

- Sarà indipendentemente dalla posizione del mobile rispetto alla cella stessa. Se ad esempio una cella di metallo la supponiamo di legno, sbagliando il suo valore di RCS, tale valore sarà sbagliato in egual misura per qualsiasi posizione che il mobile possa assumere nello scenario.
- Non varierá mai per ogni simulazione effettuata.

Le simulazioni andranno a variare la percentuale di errore dal valore utilizzato:

$$E_{\text{RRCS}} = \mathcal{N}(0, \sigma_e^2) = \mathcal{N}(0, (x \cdot \text{RRCS})) \quad (4.3)$$

Dove il valore di  $x$  é stato scelto per coprire l'intervallo dal 10 al 50%.

2. *Rumore nel modello di misura*: classico rumore additivo gaussiano a valor medio nullo, che influenzerá ogni misura effettuata dal mobile. Esso sarà diverso per ogni step di movimento ma restando fisso per ogni simulazione.

*N.B.:* i due tipi di rumore rimangono gli stessi per ogni simulazione effettuata, in quanto il seme del generatore di numeri casuali é lo stesso, in modo da poter confrontare le medesime a paritá di condizioni.

Per quanto riguarda le antenne, si sono considerati array planari quadrati, confrontando i risultati al variare del numero di antenne.

Le configurazioni di antenne usate sono:

- $N_{\text{array}}$  con  $N_x = 4$  e  $N_y = 4$ :

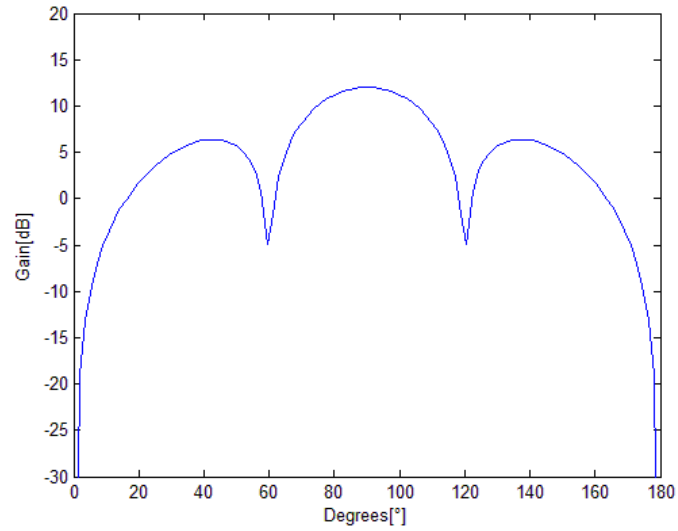


Figura 4.4: Diagramma di radiazione per  $N_{\text{array}} = 16$

- $N_{\text{array}}$  con  $N_x = 10$  e  $N_y = 10$ :

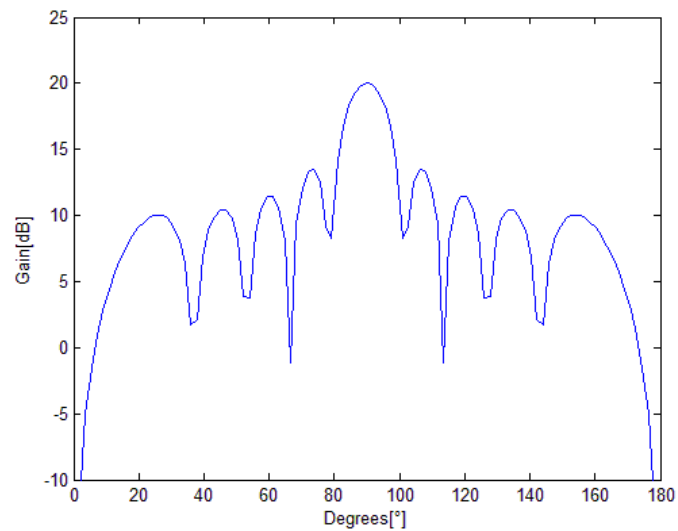


Figura 4.5: Diagramma di radiazione per  $N_{\text{array}} = 100$

- $N_{\text{array}}$  con  $N_x = 15$  e  $N_y = 15$ :

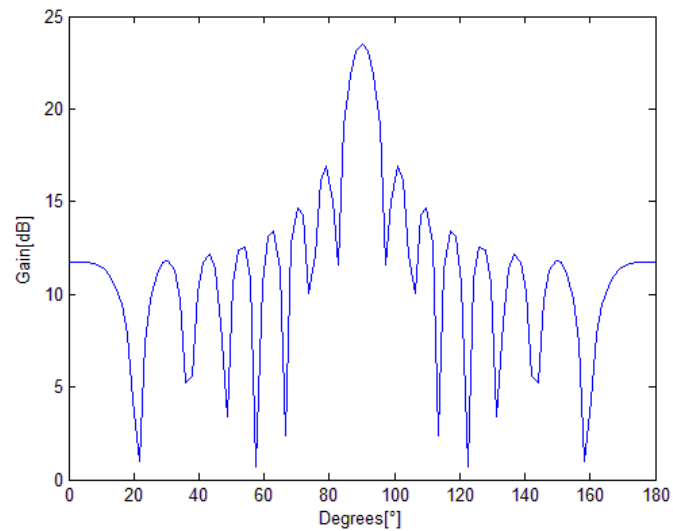


Figura 4.6: Diagramma di radiazione per  $N_{\text{array}} = 225$

Come si può notare, l'array con 225 elementi risulta molto più direttivo di un array con solo 16 elementi, questo fa sì che la risoluzione angolare sia migliore. L'obiettivo delle simulazioni è in parte quello di capire come impatta il numero di antenne, e quindi la forma del diagramma di radiazione, nell'accuratezza del mapping. Dall'altro lato si è voluta analizzare la robustezza dell'algoritmo a mismatch presenti tra modello di misura e misura reale.



## 4.2 Struttura del simulatore

La struttura del simulatore é tale da seguire i seguenti punti:

- *Calcolo dell'array factor*: permette il calcolo del diagramma di radiazione dell'array di antenne, in funzione del numero di elementi.
- *Calcolo delle celle visibili al radar*: siccome un'antenna presenta spesso un cono d'ombra alla proprie spalle, é stato implementato un codice che permette di considerare, per un dato step di movimento, solo le celle che si trovano nel semipiano frontale al mobile.
- *Creazione della mappa vera*: in questo script sono stati inseriti i comandi per creare la mappa da stimare, impostando per le celle il valore 0 o il valore RRCS. L'output di questo script é la mappa vera, che verrá considerata come benchmark per valutare l'accuratezza della stima.
- *Movimento del radar*: Il radar si muove con velocità costante velocità costante di  $1m/s$  lunga una traiettoria definita da questo script.
- *Parametri di inizializzazione*.: Questi parametri vengono suddivisi in:
  - *Constant Parameters*: dati sempre costanti al variare delle simulazioni (un esempio é il valore della velocità della luce).
  - *Variable Parameters*: dati che variavano con le simulazioni (esempio la dimensione dell'array di antenne).
- *Programma*: Il cuore centrale del progetto si divide nei seguenti blocchi:
  1. *Inizializzazione*: questa fase comprende il richiamo delle variabili e costanti in gioco, la creazione dell'array di antenne e della mappa ed infine l'inizializzazione dei vettori usati in seguito
  2. *Movimento*: simulazione del movimento del mobile
  3. *Misura*: calcolo del vettore di misura.
  4. *Rk*: creazione della (3.12).
  5. *Predizione*: calcolo della misura che mi aspetterei di ricevere nel caso l'i-esima cella fosse piena e nel caso fosse vuota.
  6. *Algoritmo di mapping*: calcolo della (3.26) che esprime la probabilità che la cella i-esima sia occupata.
  7. *Plot*: plot real time della mappa e del diagramma di irradiazione.

### 4.3 Dati di simulazione

Le simulazioni sono state effettuate utilizzando:

- $N_{\text{array}} \in [16, 100, 225]$
- $\sigma_e^2 \in [10\%, 20\%, 30\%, 40\%, 50\%]$  RRCS

Combinando i possibili valori di  $N_{\text{array}}$  con quelli di  $E_{\text{RRCS}}$  si ottengono le 15 simulazioni effettuate per ognuno dei 3 scenari. I valori utilizzati per i parametri in gioco sono:

- $F = 4$  dB: cifra di rumore
- $T_0 = 290$  K: temperatura dell'ambiente
- $T_f = 0$  ns
- $W = 1$  GHz
- $N_p = 1$
- $f_0 = 60$  GHz
- $N_{\text{steer}} = 30$
- EIRP = 0.1 mW
- RCS = 0.04 [ $m^2$ ]

## 4.4 Grafici ottenuti dalle simulazioni

I risultati si dividono in due parti:

- Per ognuna delle tre mappe prese singolarmente sono stati creati 13 grafici suddivisibili in due categorie:
  - Stime delle mappe variando i possibili array di antenne e al crescere dell'errore di modello, nella quale rientrano 5 grafici. Per le singole celle la probabilità di occupazione segue la seguente gradazione cromatica:

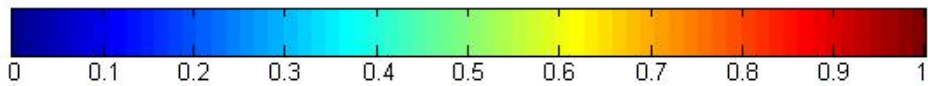


Figura 4.7: Barra cromatica indicante i livelli di probabilità di riempimento delle varie celle. I valori variano dal blu ovvero 0% al bordeaux ovvero 100%

- Percentuali di errori di stima delle celle vuote, piene e generiche, al variare di una soglia (threshold) sopra la quale la cella é da considerarsi piena, nella quale rientrano 8 grafici in funzione del numero di elementi dell'array di antenne e di  $\sigma_e^2$ .

Per alleggerire la lettura dell'elaborato i grafici seguenti, di questa sezione, si riferiscono alla prima mappa, mentre quelli relativi agli altri due scenari sono allegati in appendice.

- Grafici che mettono in relazione le varie mappe per ogni possibile simulazione effettuata, utili per confrontare come a seconda della topologia variano le percentuali di errori di stima delle celle piene, vuote o generiche. Anche in questo caso per alleggerire la lettura é stato incluso un solo caso, relegando gli altri in appendice

## 4.4.1 Grafici della mappa stimata relativa al primo scenario

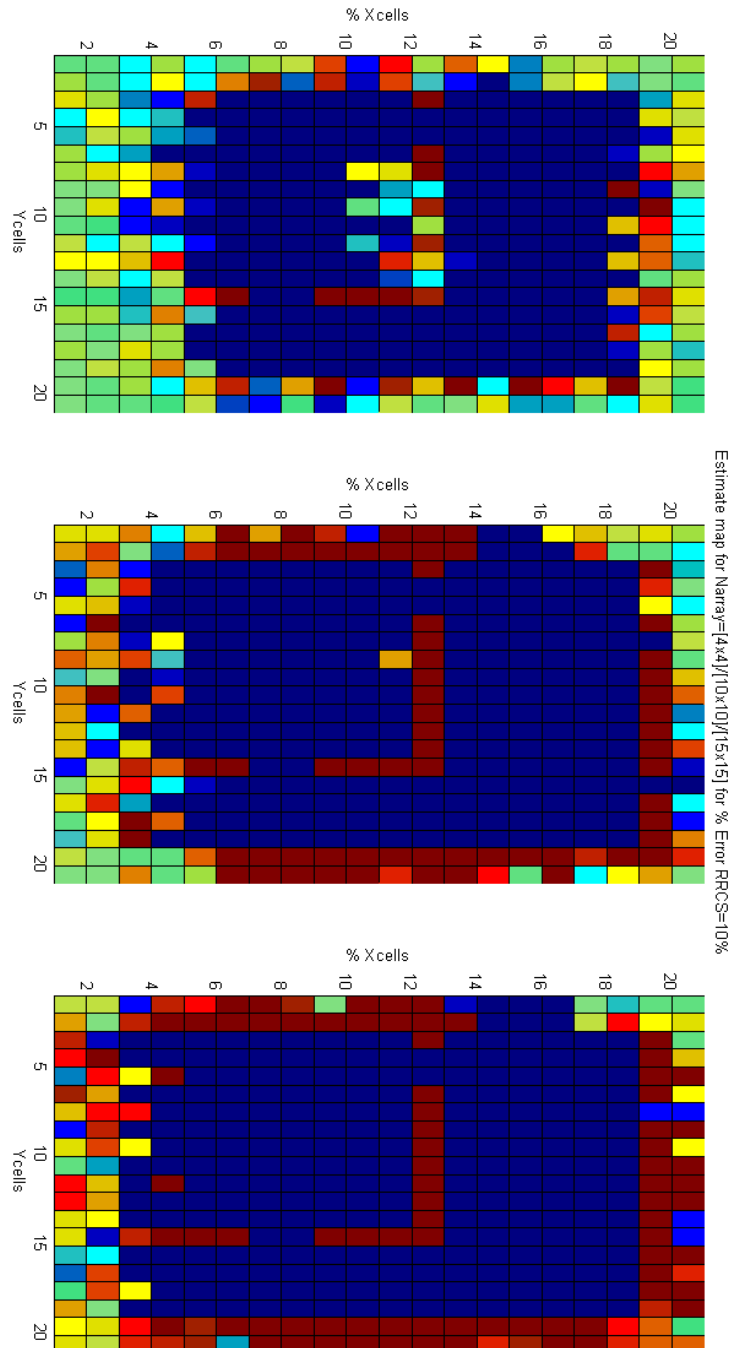


Figura 4.8: Mappa stimata per  $\sigma_e^2 = 10\%$  RRCS e al variare del numero di antenne (da sx a dx: 16, 100, 225)

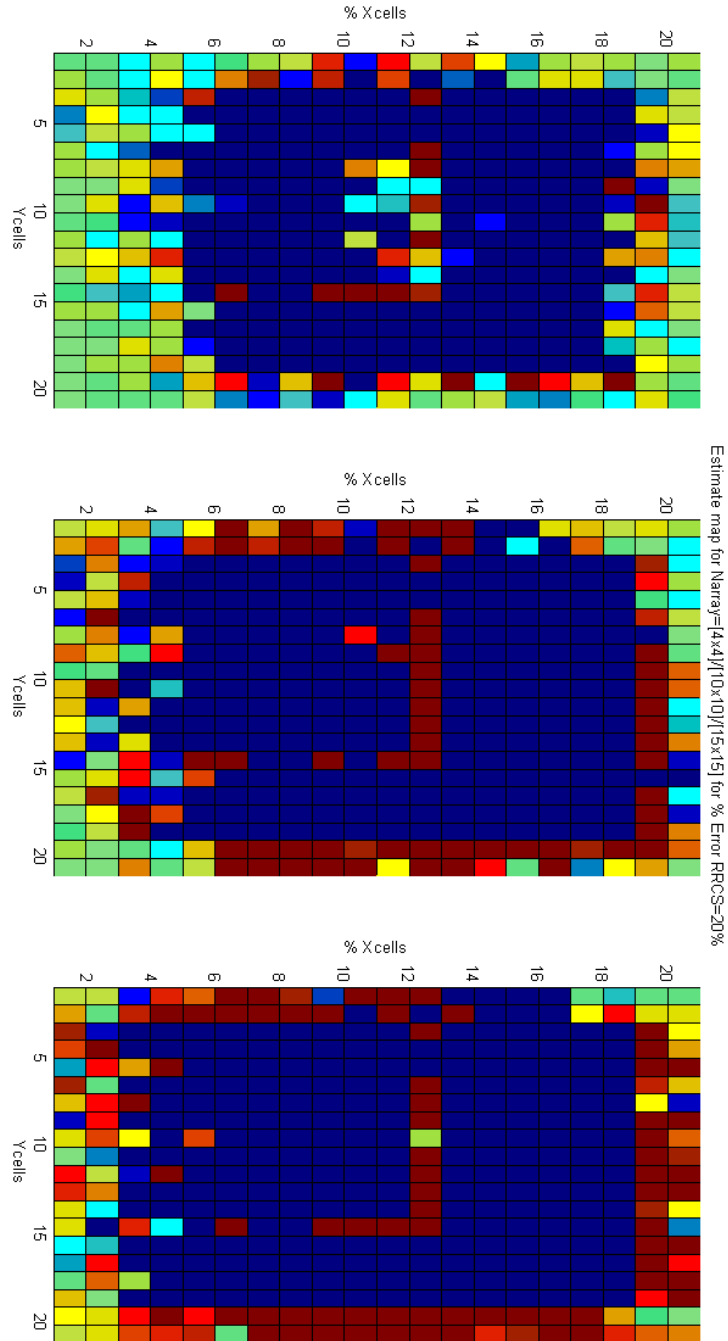


Figura 4.9: Mappa stimata per  $\sigma_e^2 = 20\%$  RRCS e al variare del numero di antenne (da sx a dx: 16, 100, 225)

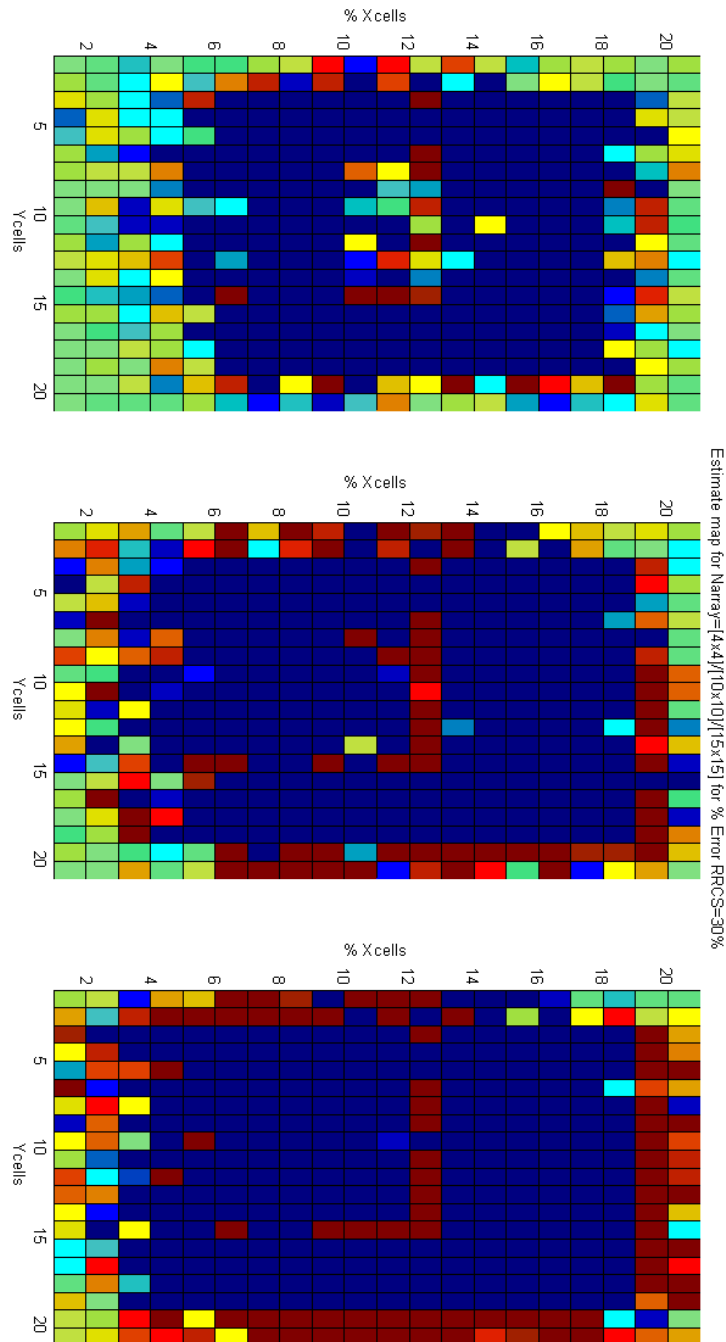


Figura 4.10: Mappa stimata per  $\sigma_e^2 = 30\%$  RRCS e al variare del numero di antenne (da sx a dx: 16, 100, 225)

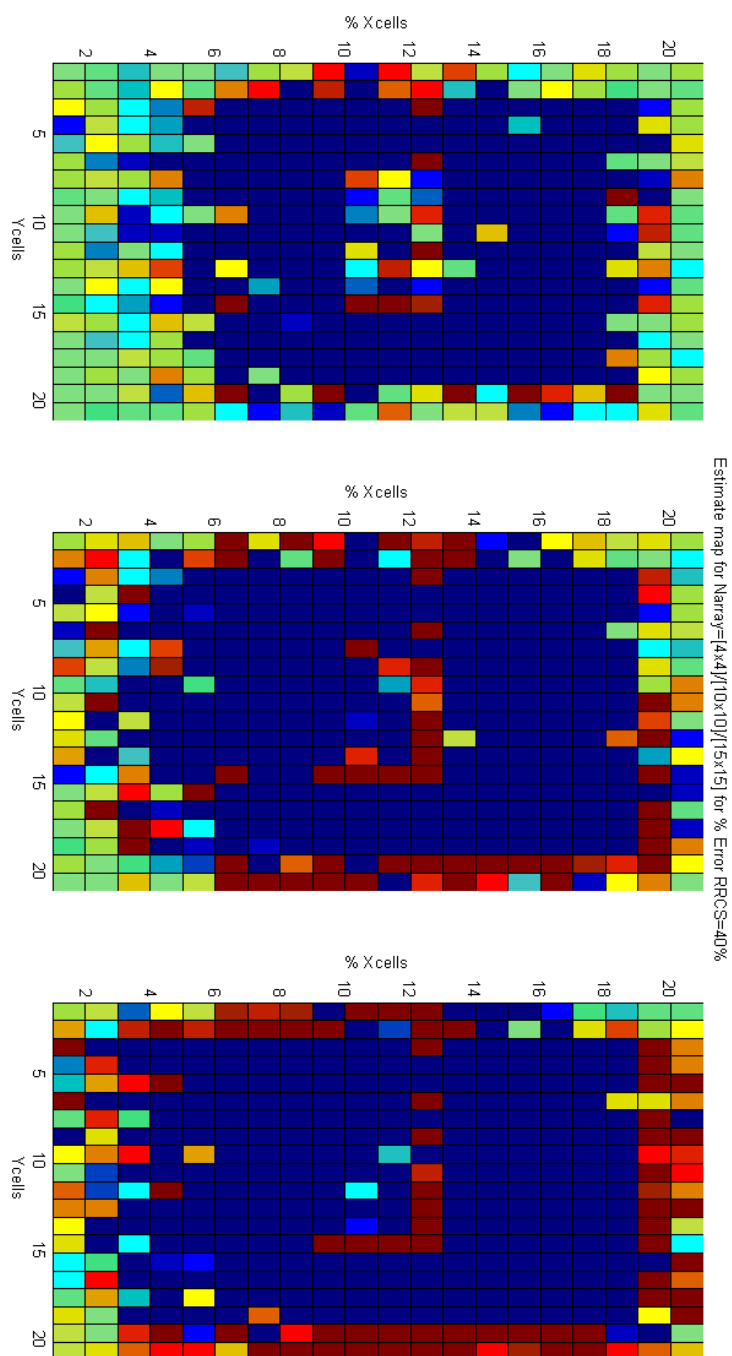


Figura 4.11: Mappa stimata per  $\sigma_e^2 = 40\%$  RRCS e al variare del numero di antenne (da sx a dx: 16, 100, 225)

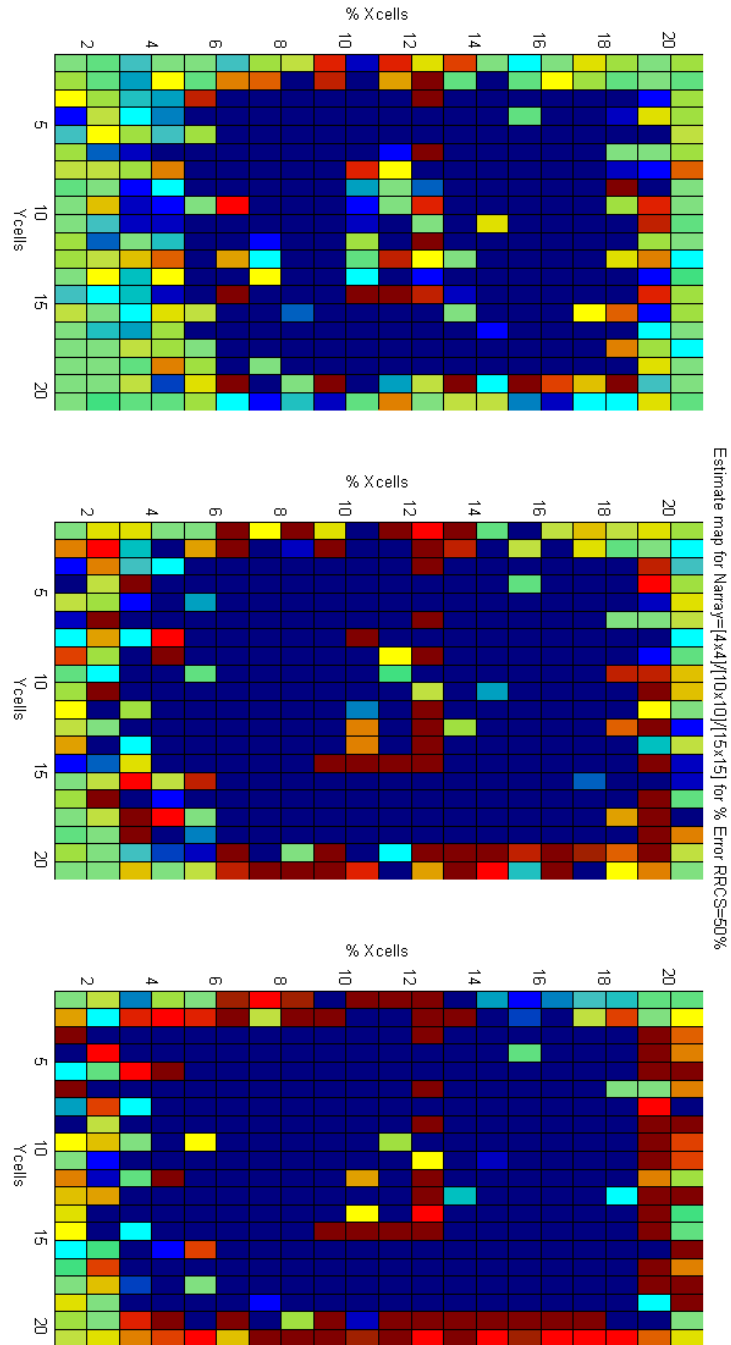


Figura 4.12: Mappa stimata per  $\sigma_e^2 = 50\%$  RRCS e al variare del numero di antenne (da sx a dx: 16, 100, 225)



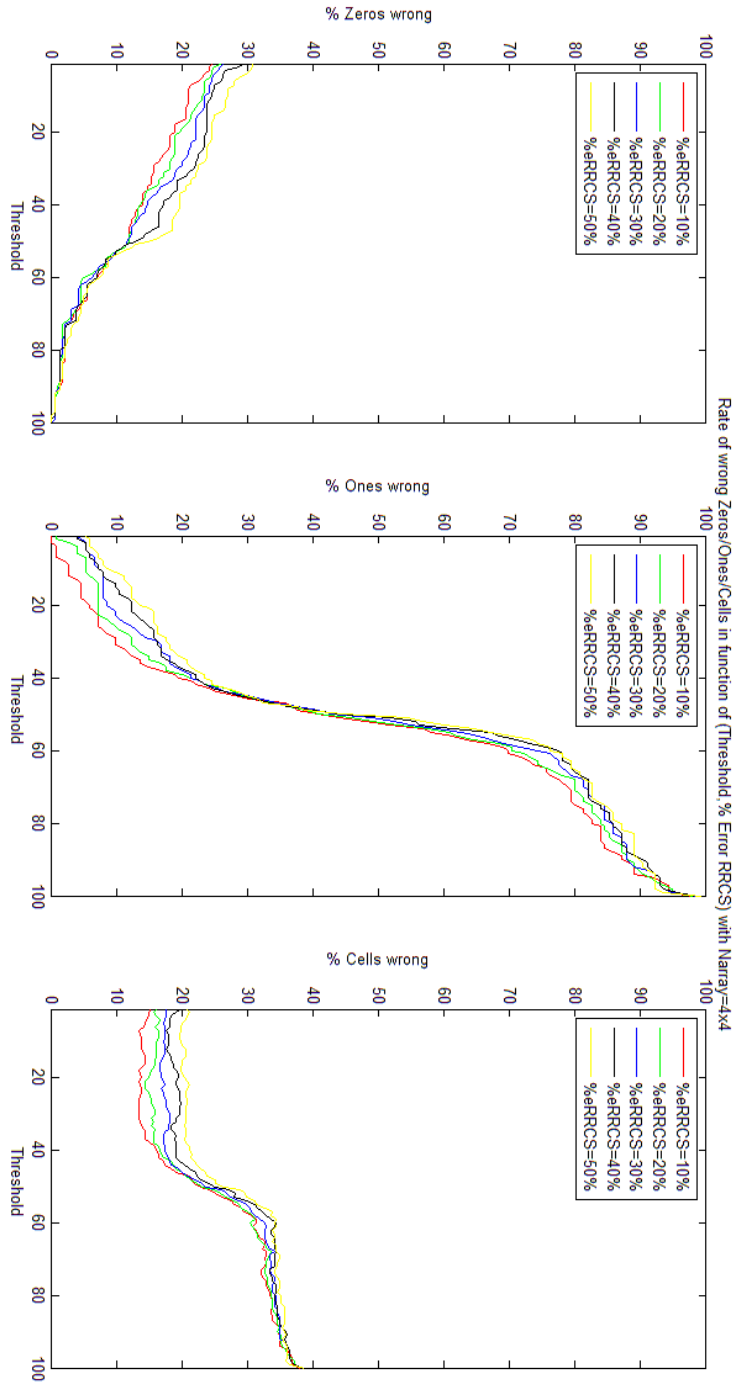


Figura 4.13: % di celle errate in funzione della soglia per  $N_{\text{array}} = 16$  al variare di  $\sigma_c^2$

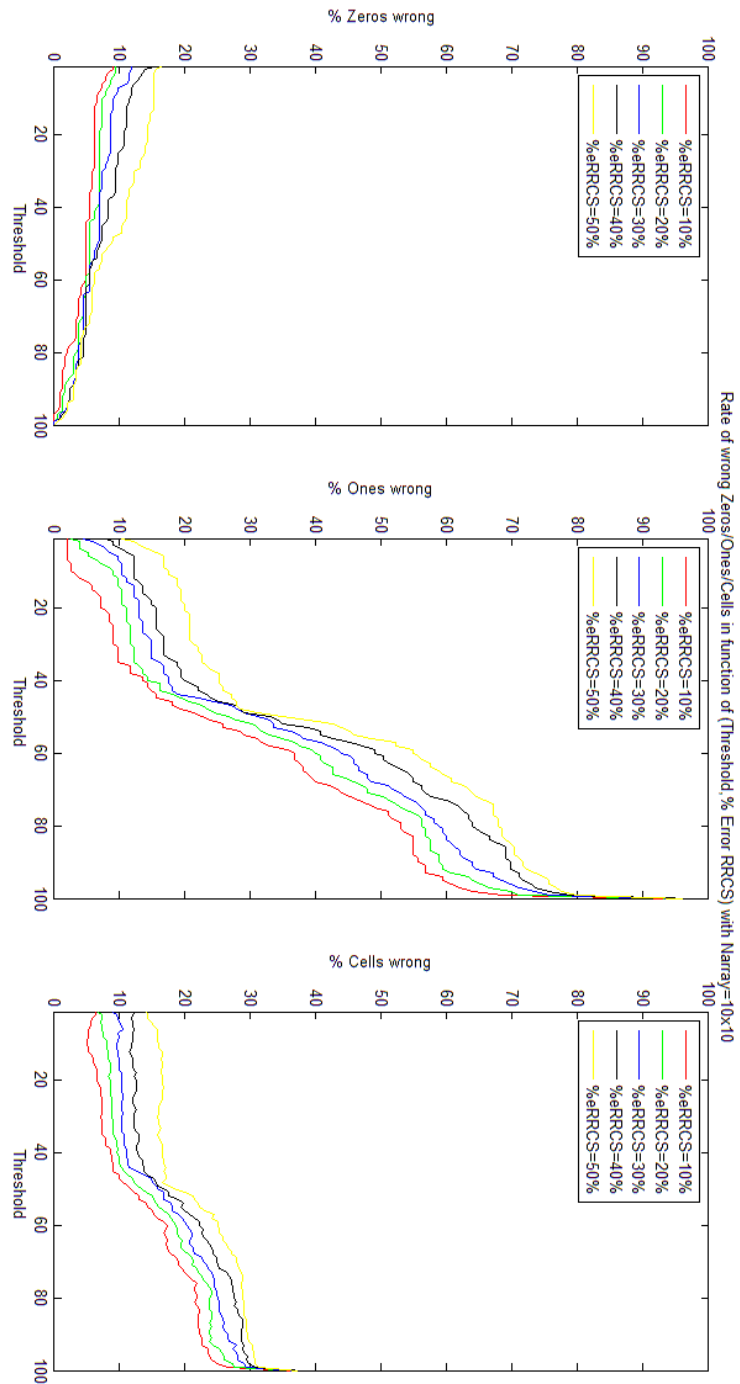


Figura 4.14: % di celle errate in funzione della soglia per  $N_{\text{array}} = 100$  al variare di  $\sigma_c^2$

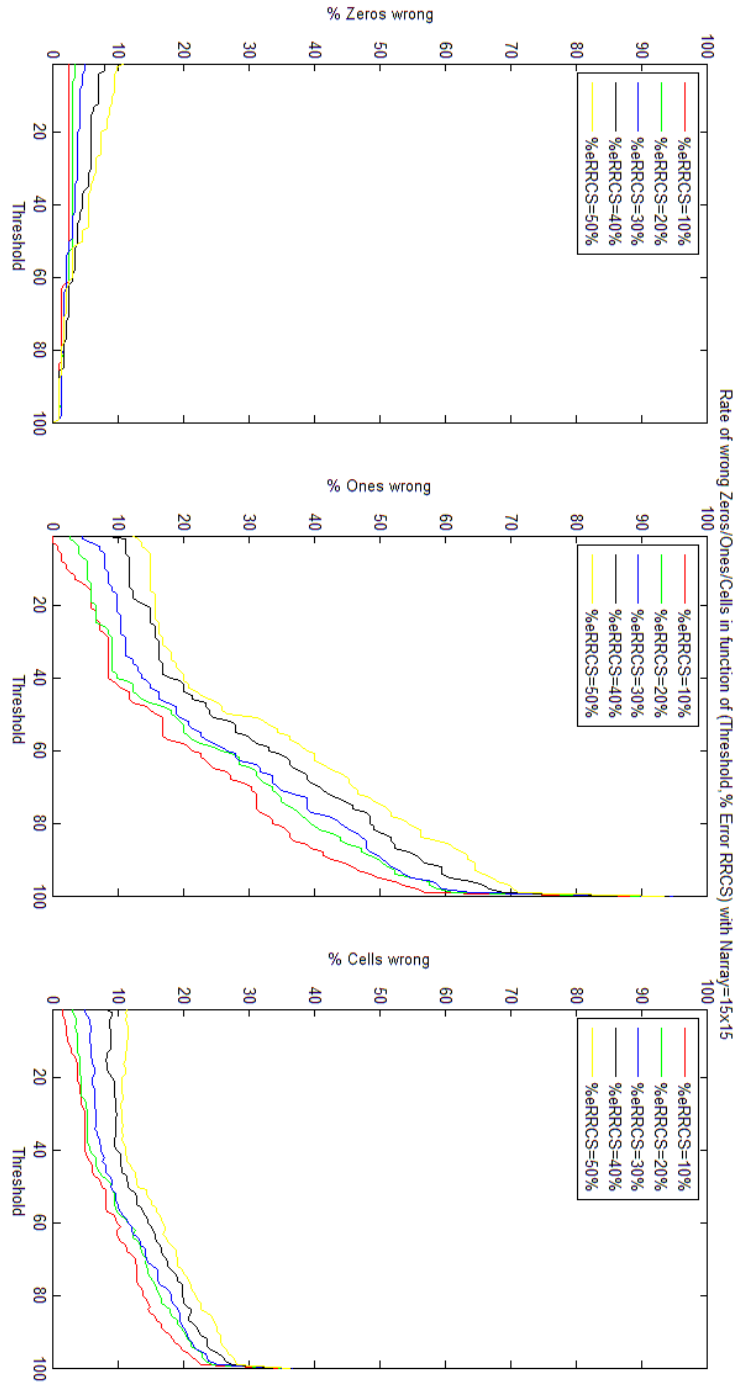


Figura 4.15: % di celle errate in funzione della soglia per  $N_{array} = 225$  al variare di  $\sigma_c^2$

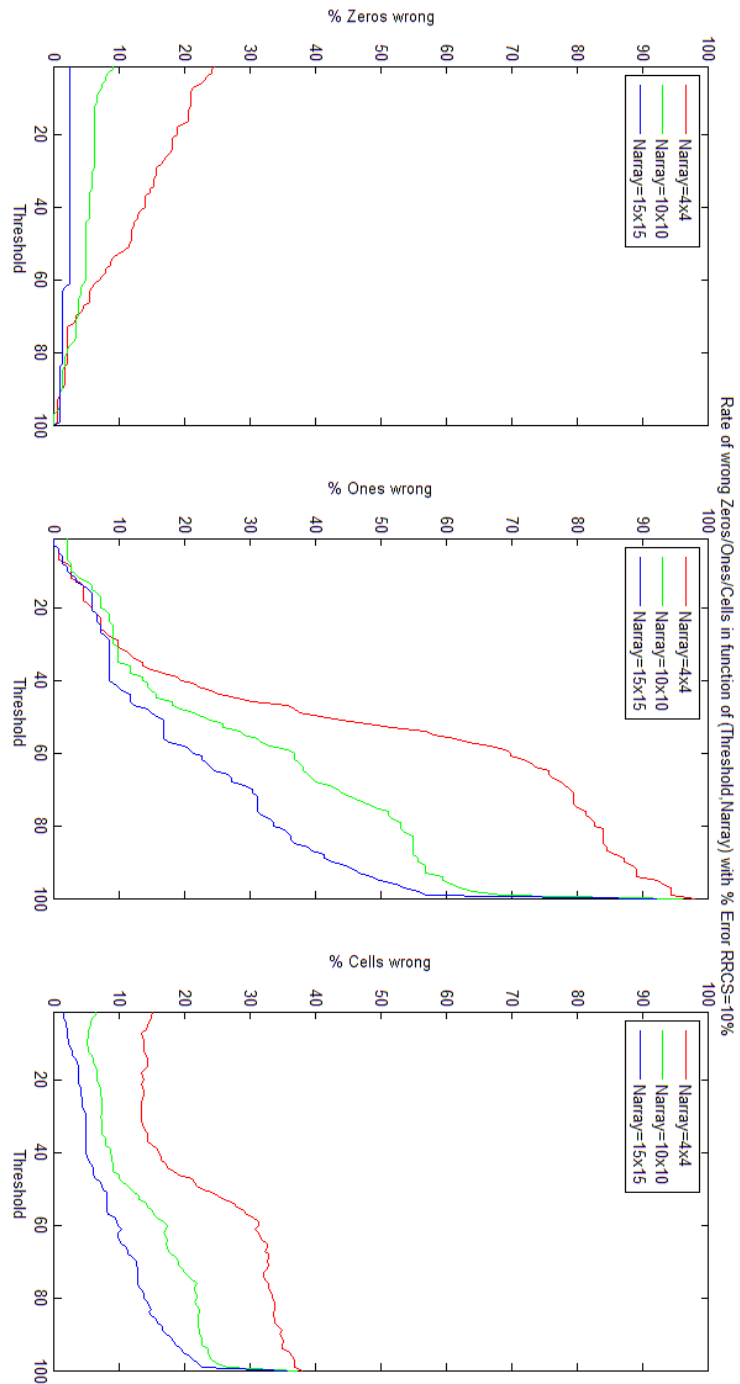


Figura 4.16: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 10\%RRCs$  al variare di  $N_{array}$

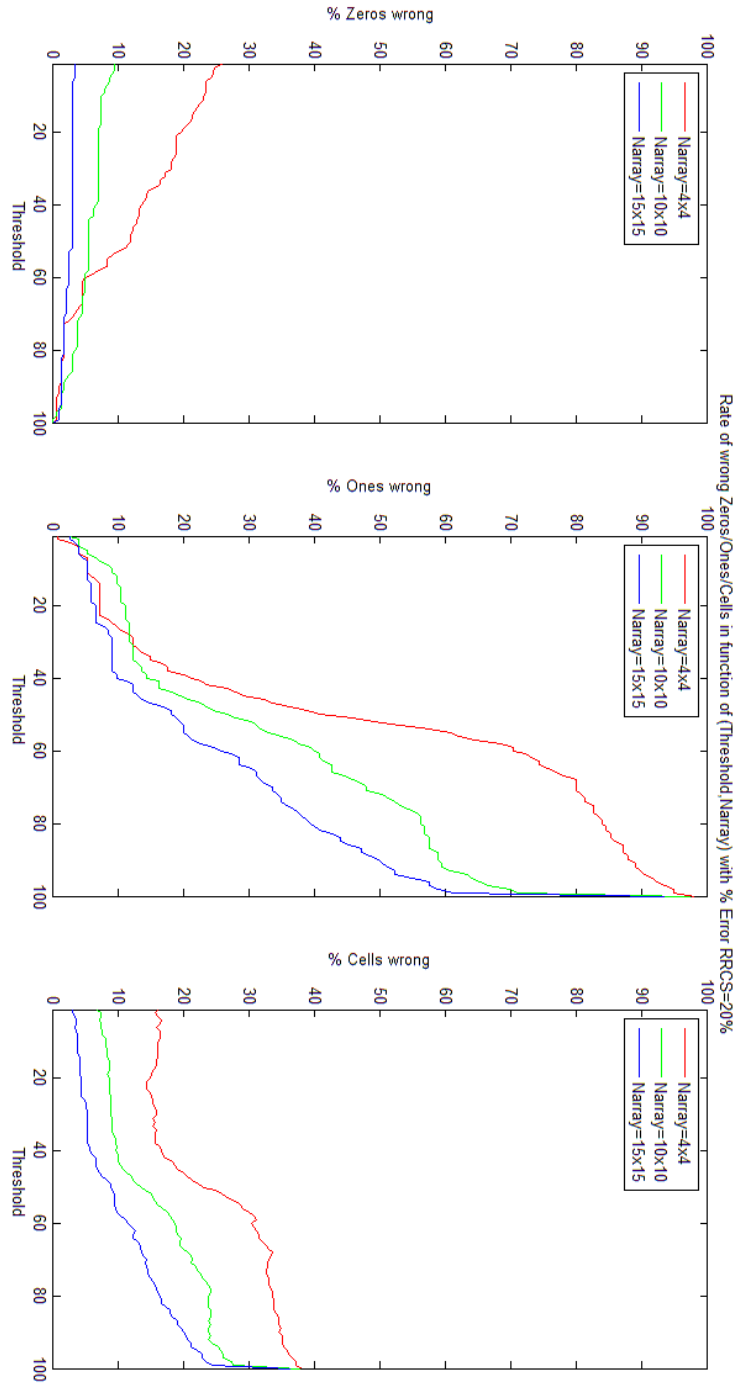


Figura 4.17: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 20\%RRCS$  al variare di  $N_{array}$

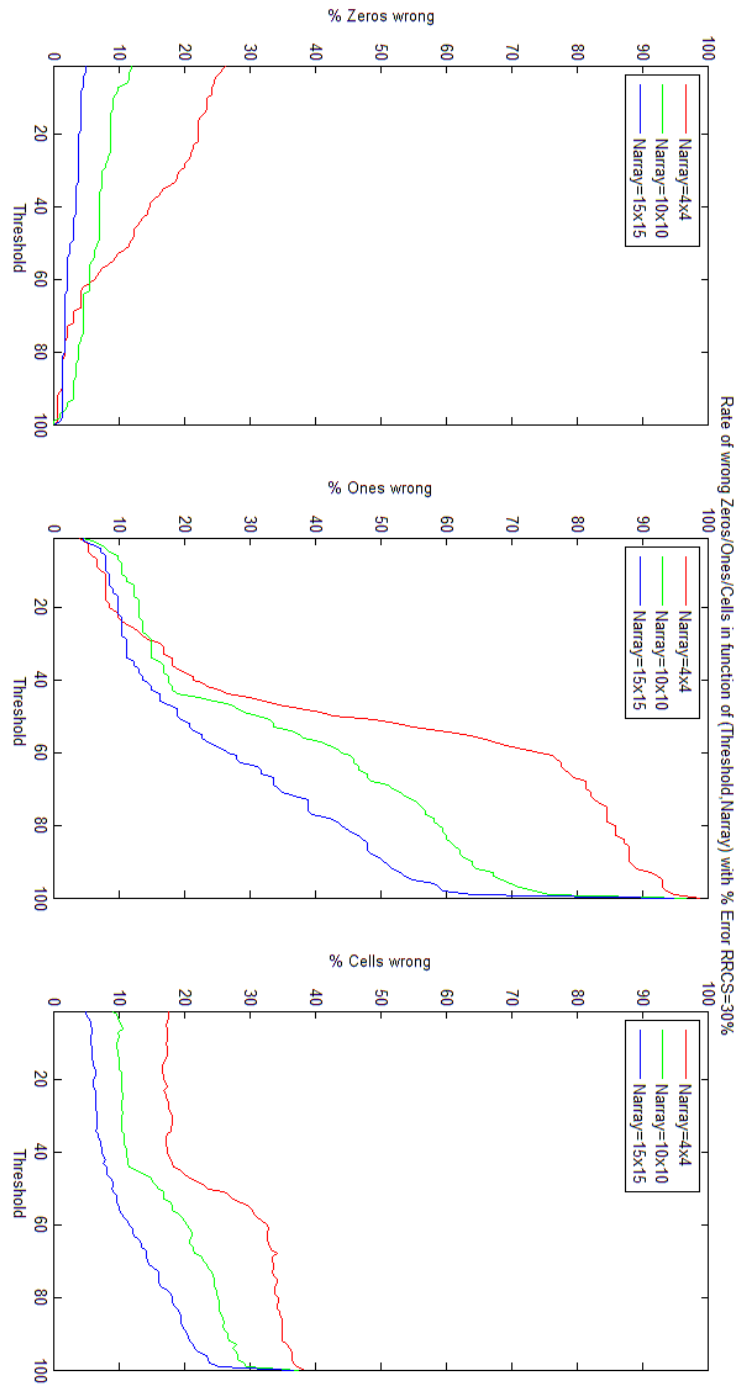


Figura 4.18: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 30\%RRCs$  al variare di  $N_{array}$

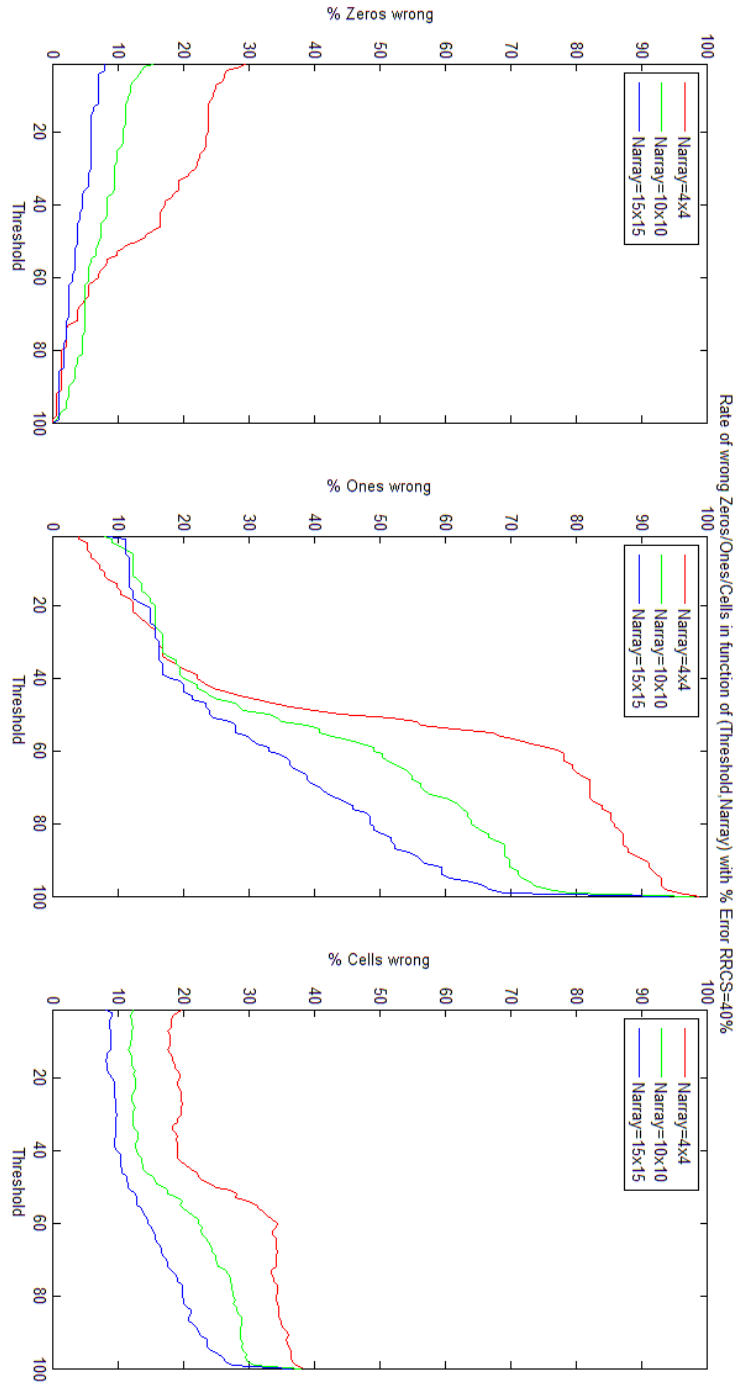


Figura 4.19: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 40\%$ RRCS al variare di  $N_{array}$

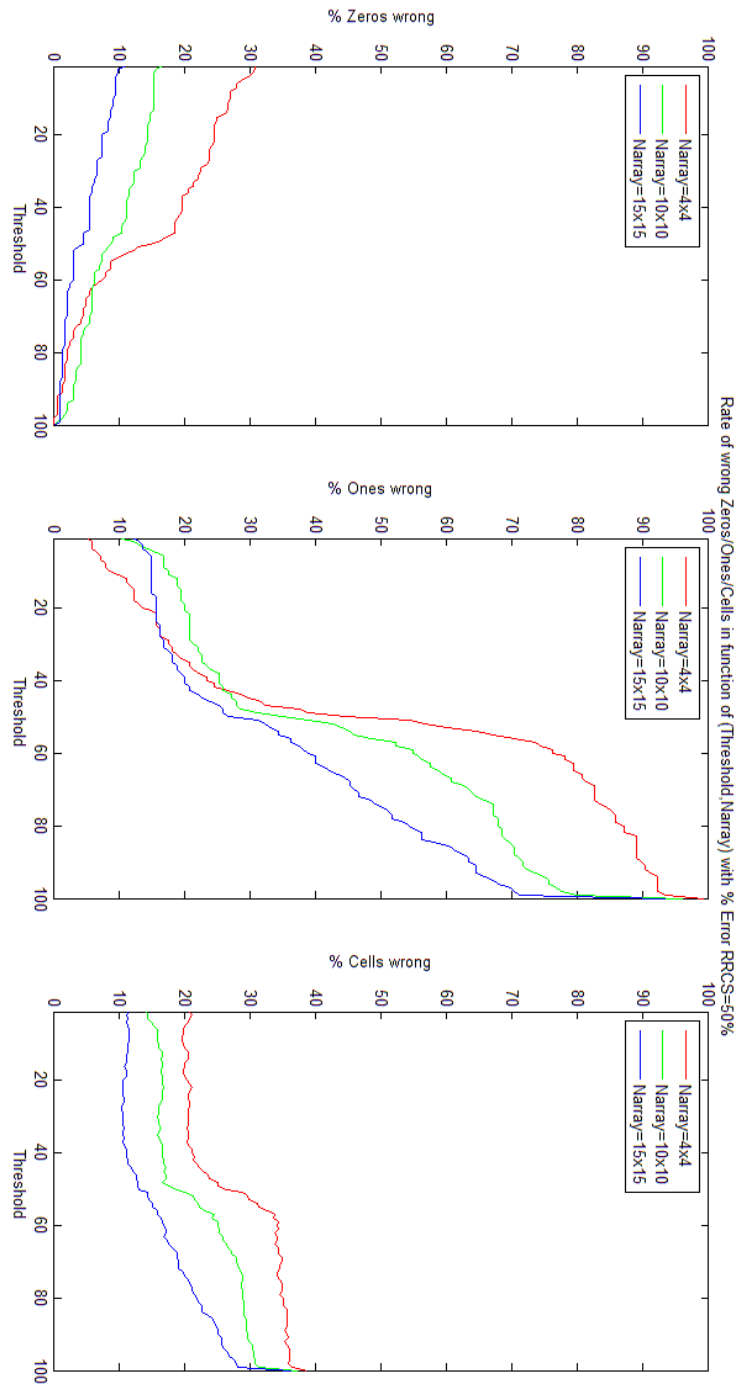


Figura 4.20: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 50\%RRCs$  al variare di  $N_{array}$



## 4.4.2 Grafici dei confronti tra i tre scenari simulati

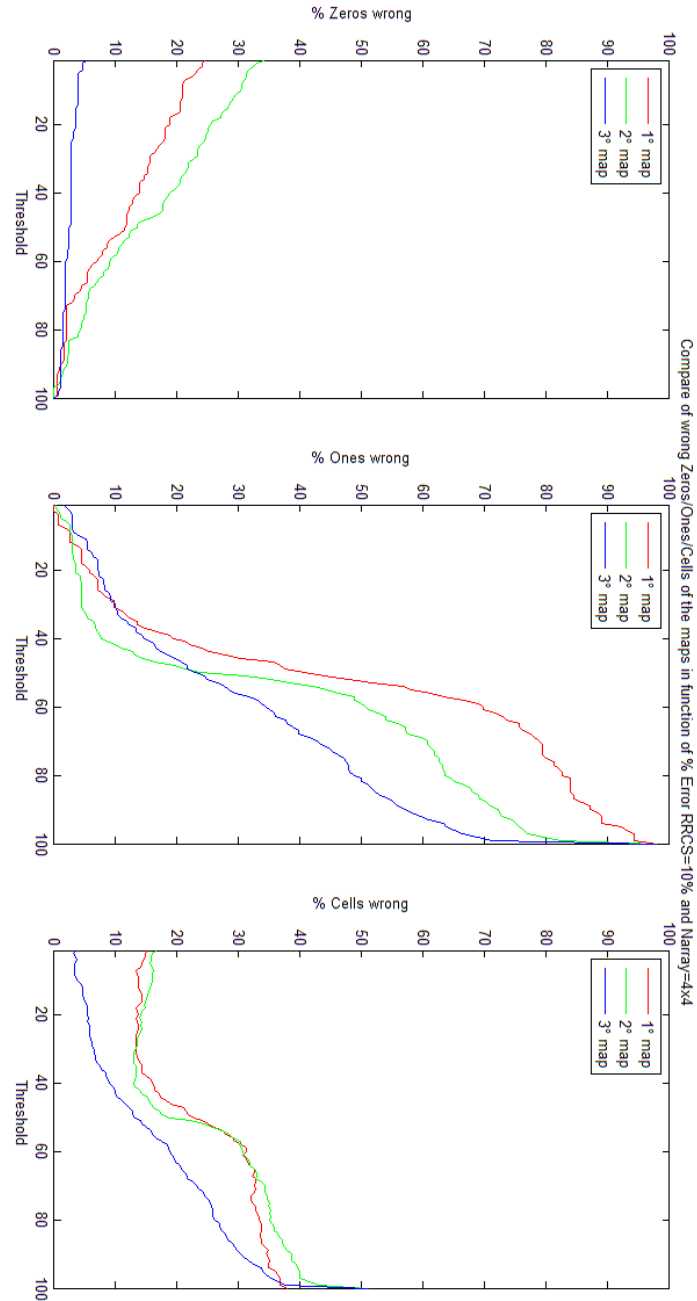


Figura 4.21: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 10\%RRCS$  e  $N_{array} = 16$

## 4.5 Commento dei risultati

Da tali risultati, indipendentemente dalla mappa presa in esame, si evince che:

- Da (Fig. 4.8) a (Fig. 4.12):  
al crescere della grandezza dell'array di antenne, il lobo principale risultando piú stretto, migliora la precisione del mapping. Questo é abbastanza evidente confrontando le mappe stimate dove si nota che, indipendentemente dall'incertezza del modello  $\sigma_e$ , la precisione nella ricostruzione dello scenario migliora sensibilmente aumentando il numero di elementi dell'antenna.
- Da (Fig. 4.8) a (Fig. 4.12):  
si puó facilmente osservare che al crescere di  $\sigma_e$  la ricostruzione della stanza peggiora notevolmente indipendentemente dal valore dell'array di antenne
- Da (Fig. 4.13) a (Fig. 4.20):  
Indipendentemente dal numero di elementi presente nell'antenna, grazie all'asimmetria del problema spiegato precedentemente, al crescere della soglia le curve decrescenti degli 0 e le curve crescenti degli 1 non sono speculari. Si noti infatti come la percentuale di errori commessi sulla stima delle celle vuote é enormemente piú basso rispetto al corrispettivo per le piene.
- Da (Fig. 4.13) a (Fig. 4.15):  
la crescita di  $\sigma_e$  influenza in maniera meno significativa l'array di antenne piú piccolo (figura 4.13), dove le 5 curve collassano tutte piú o meno verso la medesima forma (specialmente quella che riguarda gli 1 errati), mentre risulta molto piú marcato al crescere del numero di elementi. Si noti inoltre come la forma delle funzioni, a seconda del setup dell'antenna, cambi radicalmente, passando da una funzione con 2 flessi a un andamento di tipo simile ad un esponenziale.
- Da (Fig. 4.16) a (Fig. 4.20):  
interessante notare come per valori bassi di soglia, l'array di antenne che effettua meno errori nella decisione degli stati delle celle piene sia quello con numero di elementi minore. Questo potrebbe derivare dal fatto che il guadagno massimo influenza il valore delle belief, e con un numero di elementi basso il suo valore non é tale da poter spingere cosí in basso il valore. Questo che a prima vista potrebbe sembrare un vantaggio, in realtà é estremamente sfavorevole perché si ripercuote anche sulle scelte giuste, impedendo anche alle celle effettivamente piene il raggiungimento del valore massimo.

- Da (Fig. 4.21):  
Piú la mappa risulta densa di celle piene, distribuite una vicina all'altra, piú la qualità della ricostruzione cresce. Per tale motivo la mappa relativa allo scenario 3 risulta sempre la migliore in qualsiasi condizione.
- Da (Fig. 4.21):  
In un qualunque grafico di qualsiasi mappa, la precisione di ricostruzione degli uni, degli zeri e delle celle generiche segue sempre lo stesso ordine. Se ad esempio studiando le celle piene si ha, leggendo i grafici dall'alto verso il basso, un certo ordine di comparizione delle funzioni che significa un miglioramento della percentuale di errori di stima, tale ordine é rispettato anche per le celle vuote e le celle generiche. Confrontando le mappe invece si scopre che, indipendentemente dal valore di  $\sigma_e$  e del numero di elementi dell'antenna:
  - Diminuendo il numero di celle piene si alza la percentuale di celle piene sbagliate. Questo porta al fatto che nella la mappa relativa allo scenario 1 si commettono piú errori nella stima delle celle piene rispetto alle alte.
  - Tenendo fisso il numero di celle piene ma sparpagliandole per la stanza si peggiora la qualità di ricostruzione degli spazi vuoti. La mappa 2 ricostruisce quindi peggio le celle vuote, in netta contrapposizione con il punto precedente.



## Capitolo 5

# Conclusioni e Sviluppi Futuri

In questa tesi si é affrontato lo sviluppo di una soluzione originale per il problema del mapping. Il tipo di algoritmo utilizzato si é rivelato essere un ottimo stimatore per ambienti indoor. I risultati ottenuti simulando vari tipi di scenari, profondamente diversi sia per numero di celle occupate sia per topologia, hanno confermato che:

- Tale algoritmo riesce ad ottenere risultati piú che soddisfacenti anche nei casi in cui il mismatch tra il modello utilizzato e la realtà era particolarmente accentuato. Questo é un grande punto a favore in quanto quando si stima un ambiente sconosciuto, creare un modello preciso risulta estremamente difficile, se non impossibile.
- Il modo asimmetrico con cui si ricavano (3.45) porta ad avere sempre la probabilità di stima delle celle vuote ad essere molto piú precisa. Questo é un altro grande punto a favore in quanto, quando si suddivide un ambiente indoor, le celle piene risultano essere sempre un numero molto piú grande delle celle vuote.
- Come é logico aspettarsi, aumentando il numero di elementi nell'antenna, ottenendo lobi sempre piú stretti, la precisione di stima migliora notevolmente.

Questo algoritmo risulta essere un altro passo in avanti per sostituire i dispositivi laser con la tecnologia ad mmW, che permetterà di integrare tale tecnologia anche in piccoli dispositivi elettronici come gli smartphone. Partendo da tale algoritmo i passi successivi sono:

1. Estendere l'algoritmo allo SLAM, e quindi alla stima simultanea della mappa e della traiettoria del robot;

2. Integrare le informazioni raccolte con il radar con contenuti visuali, ad es. da fotocamere, per emulare il personal radar e migliorare l'accuratezza nella ricostruzione dell'ambiente.
3. Simulare piú di un personal radar per mappa e valutare l'improvement ottenuto quando piú dispositivi collaborano tra di loro;
4. Migliorare il modello di radar cross-section adottato.

## Capitolo 6

# Appendice 1 - Grafici rimanenti

### 6.1 Grafici per le restanti mappe analizzate

Di seguito saranno rilegati:

- grafici di analisi delle mappe degli scenari 2 e 3 simili alle (Fig. 4.8),  
...,(Fig. 4.20)
- grafici di confronto sotto diverse condizioni, simili alla (Fig. 4.21)

### 6.1.1 Grafici della mappa stimata relativa al secondo scenario

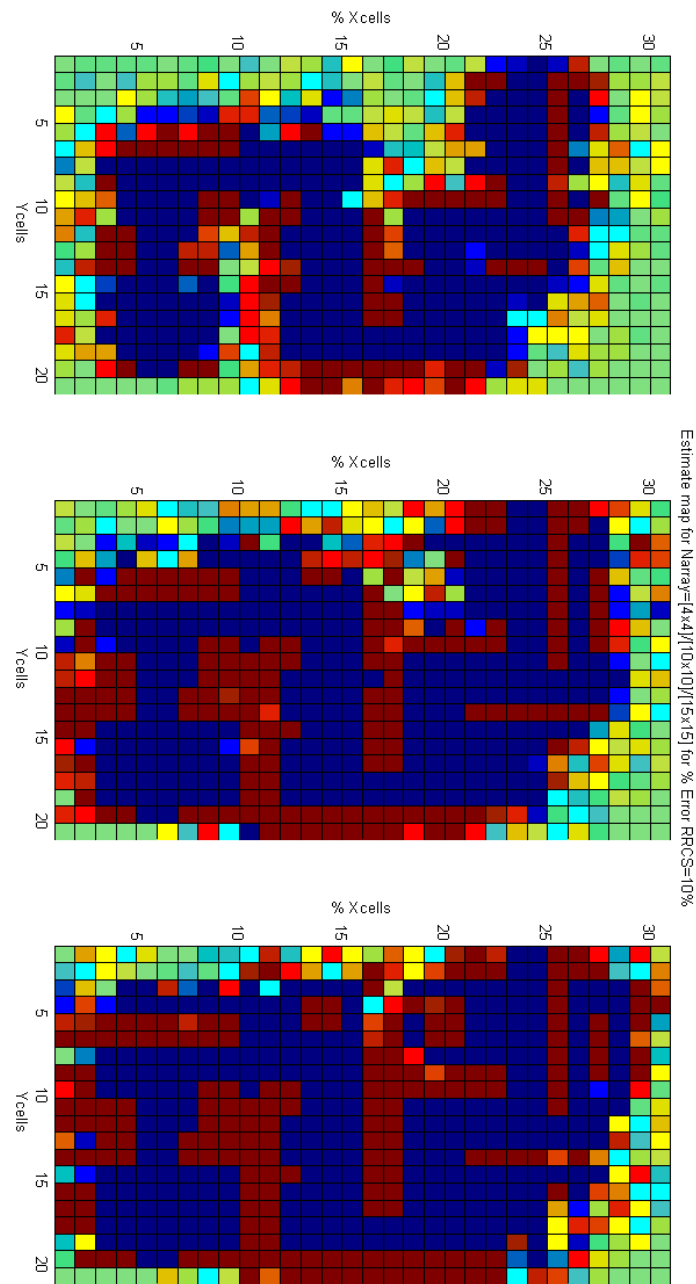


Figura 6.1: Mappa stimata per  $\sigma_e^2 = 10\%$  RRCS e al variare del numero di antenne (da sx a dx: 16, 100, 225)



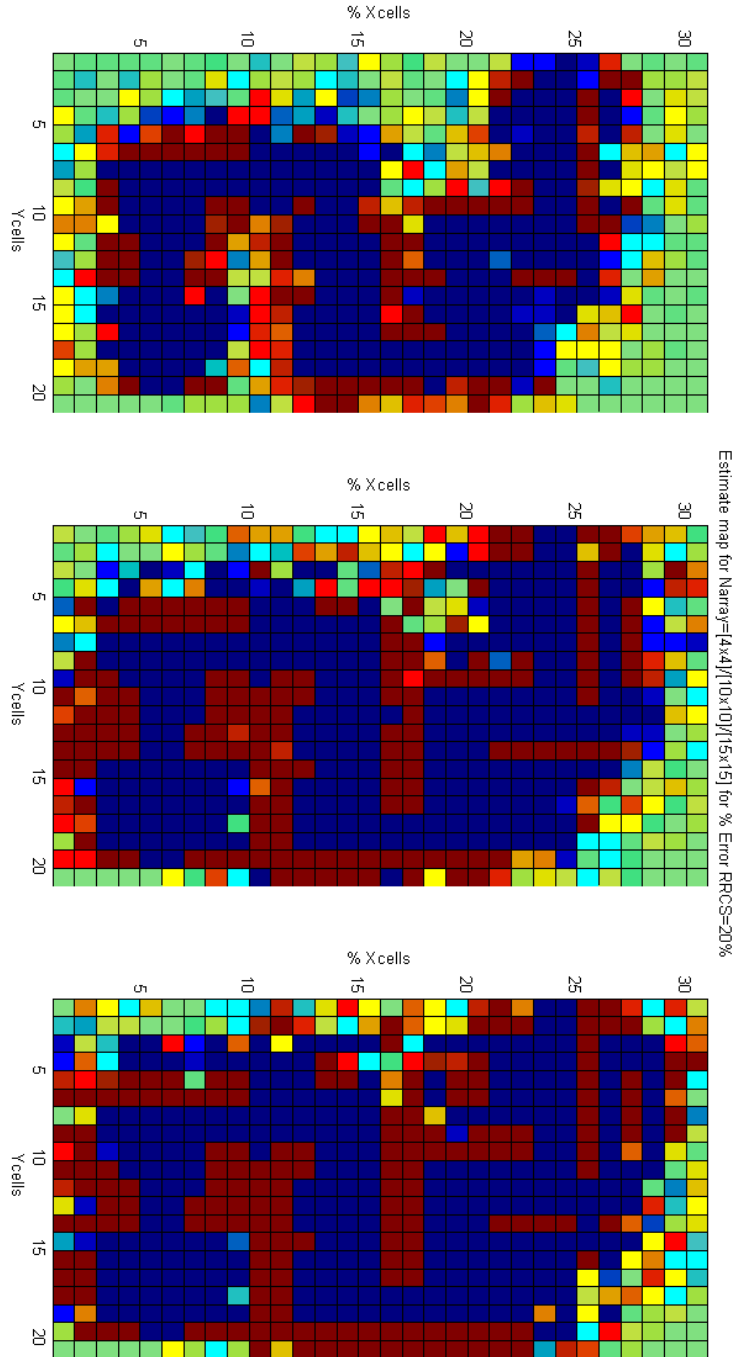


Figura 6.2: Mappa stimata per  $\sigma_e^2 = 20\%$  RRCS e al variare del numero di antenne (da sx a dx: 16, 100, 225)

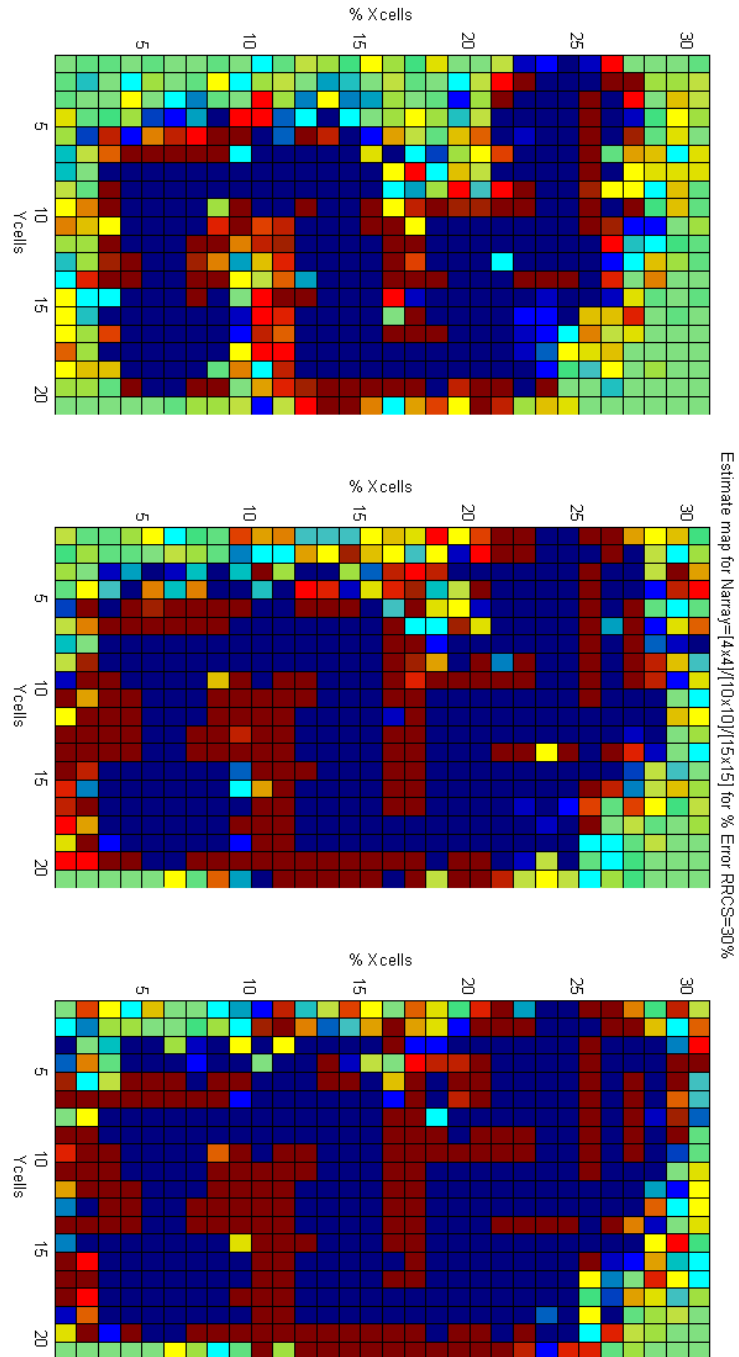


Figura 6.3: Mappa stimata per  $\sigma_e^2 = 30\%$  RRCS e al variare del numero di antenne (da sx a dx: 16, 100, 225)

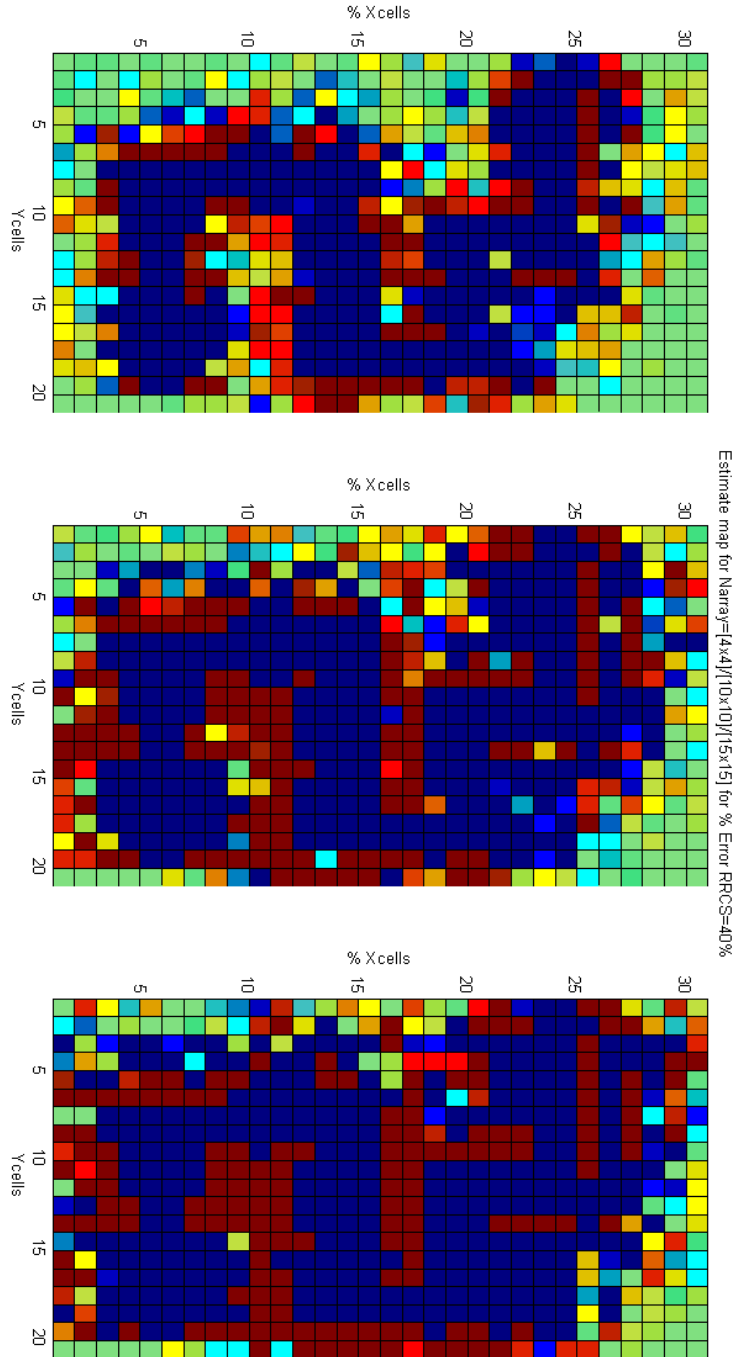


Figura 6.4: Mappa stimata per  $\sigma_e^2 = 40\%$  RRCS e al variare del numero di antenne (da sx a dx: 16, 100, 225)

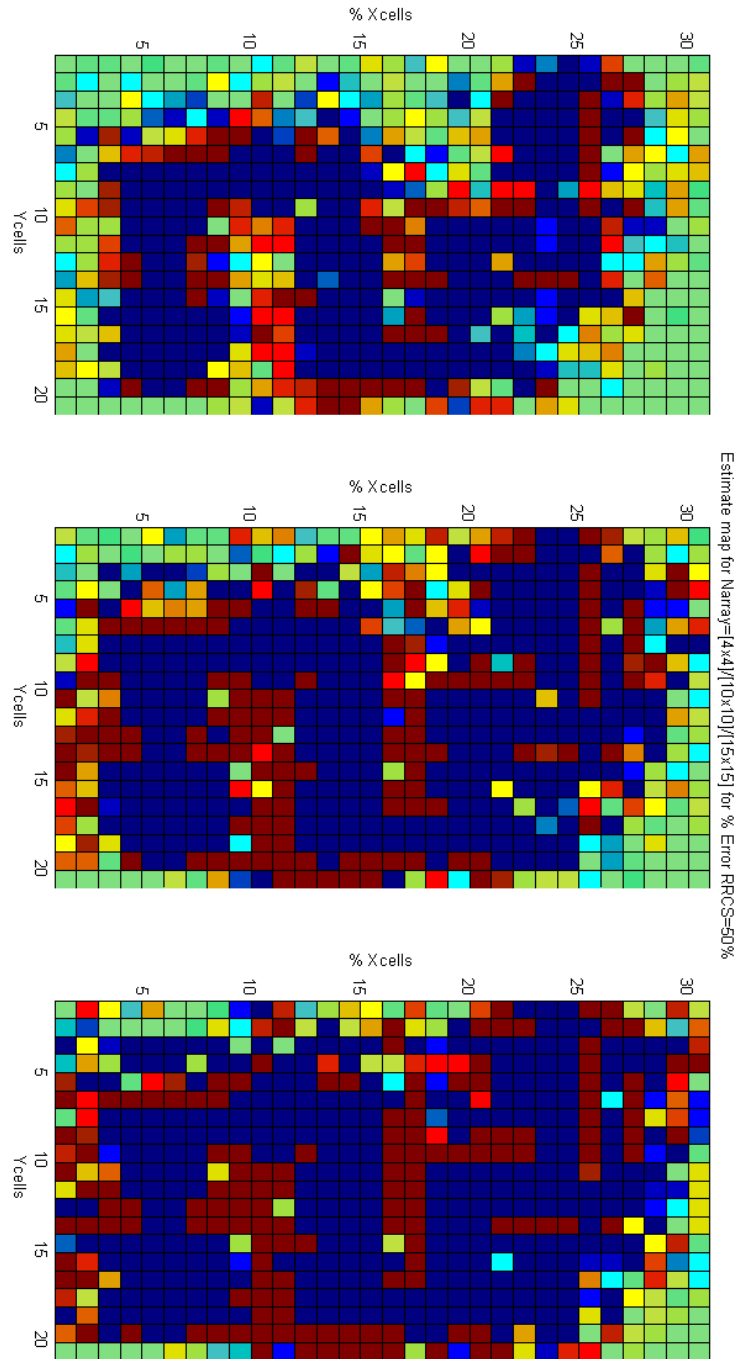


Figura 6.5: Mappa stimata per  $\sigma_e^2 = 50\%$  RRCS e al variare del numero di antenne (da sx a dx: 16, 100, 225)

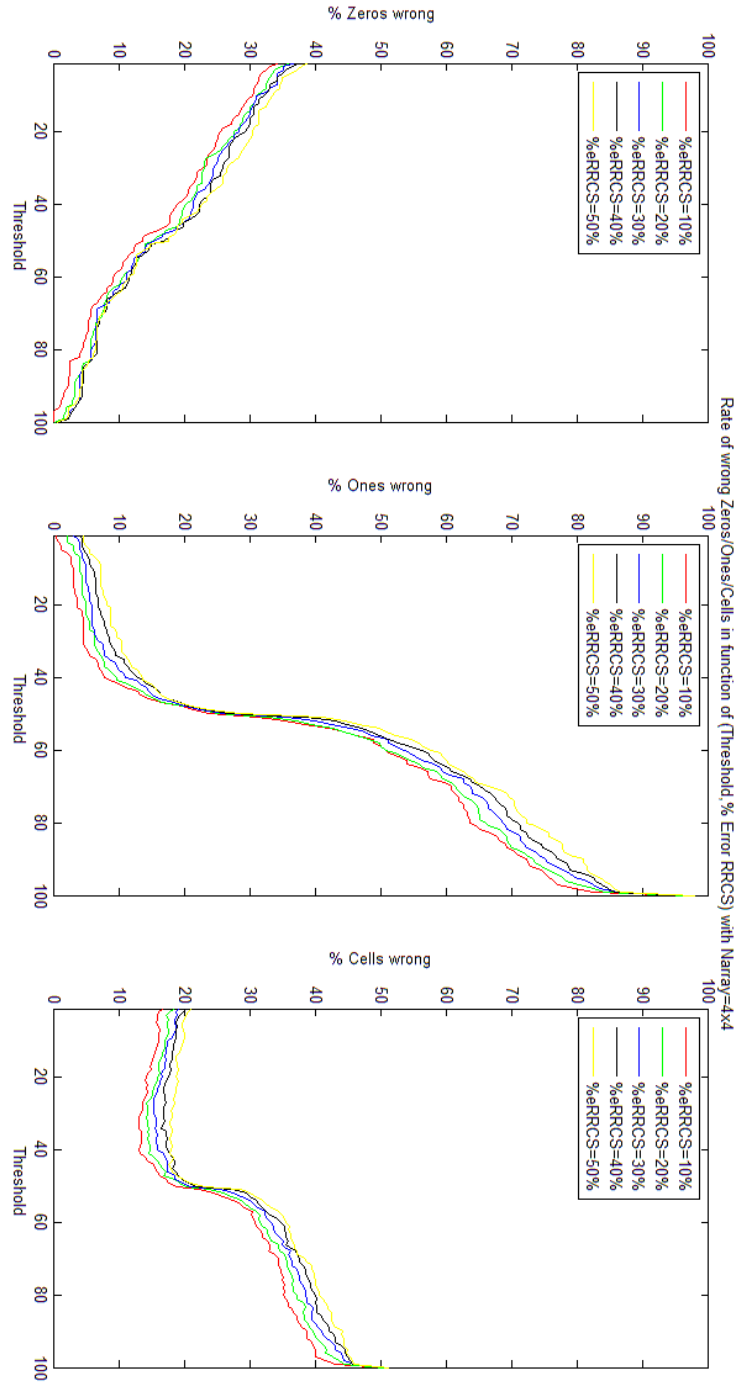


Figura 6.6: % di celle errate in funzione della soglia per  $N_{\text{array}} = 16$  al variare di  $\sigma_c^2$

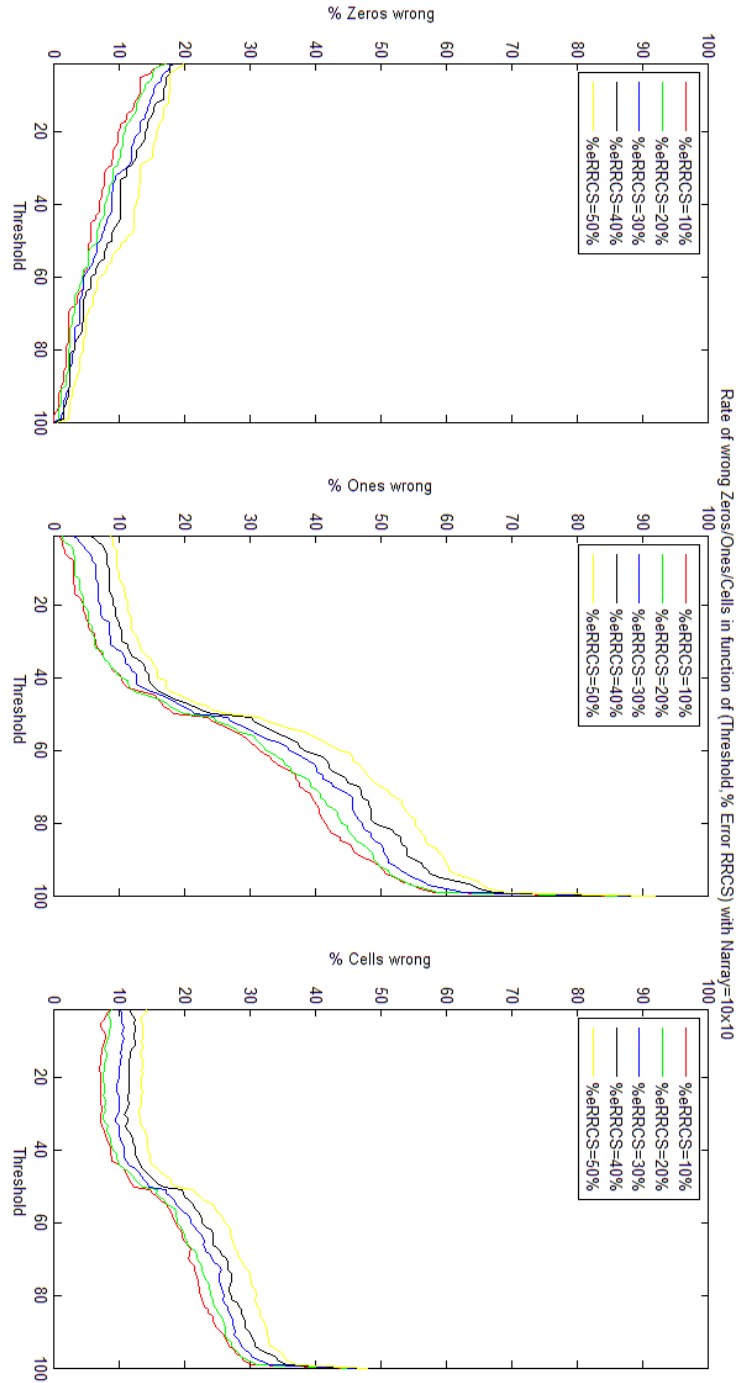


Figura 6.7: % di celle errate in funzione della soglia per  $N_{\text{array}} = 100$  al variare di  $\sigma_c^2$

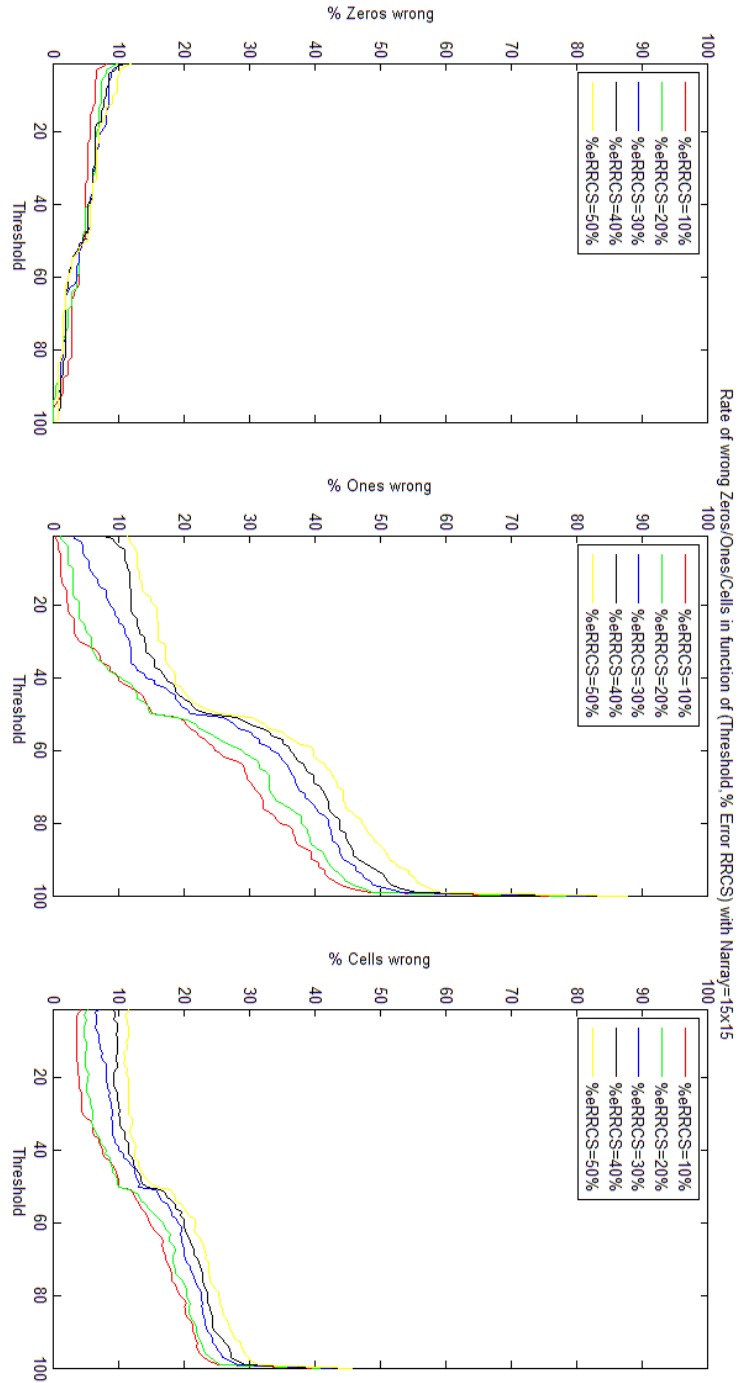


Figura 6.8: % di celle errate in funzione della soglia per  $N_{\text{array}} = 225$  al variare di  $\sigma_e^2$

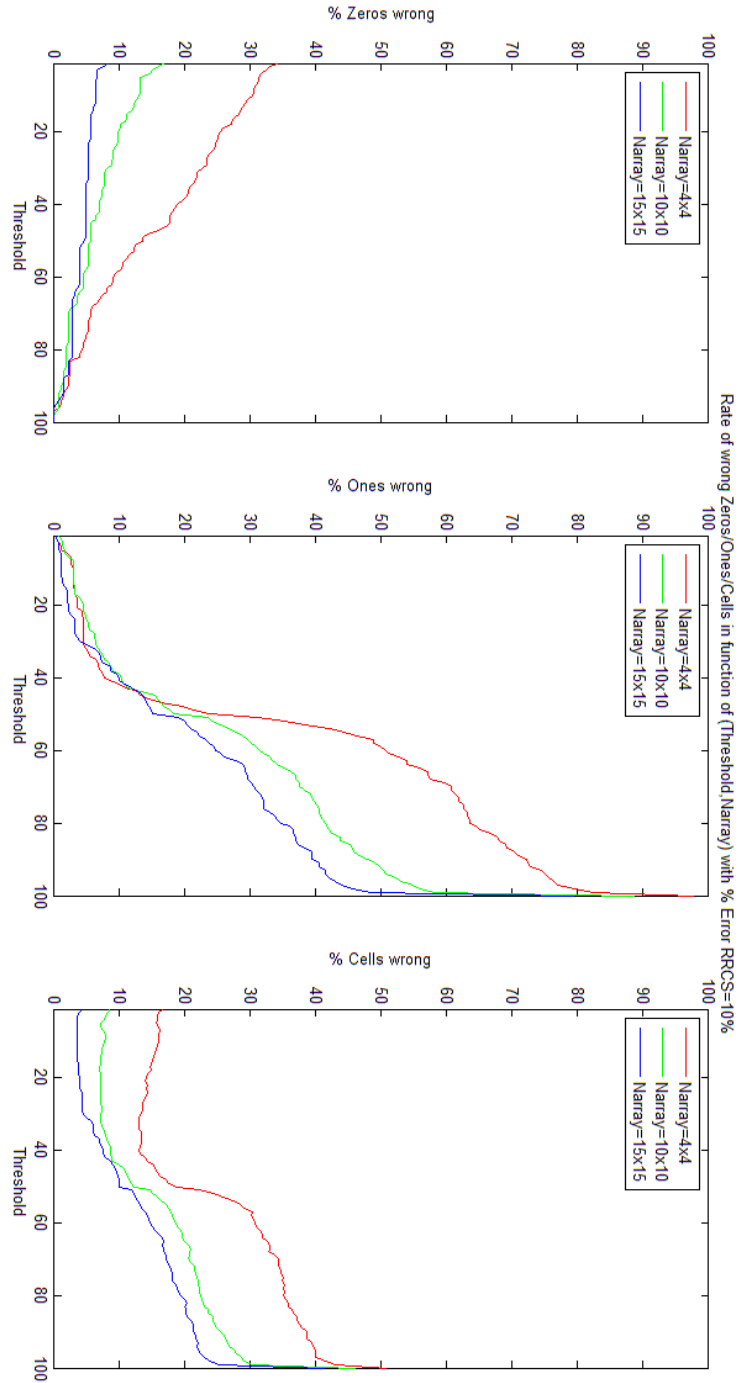


Figura 6.9: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 10\%RRCs$  al variare di  $N_{array}$



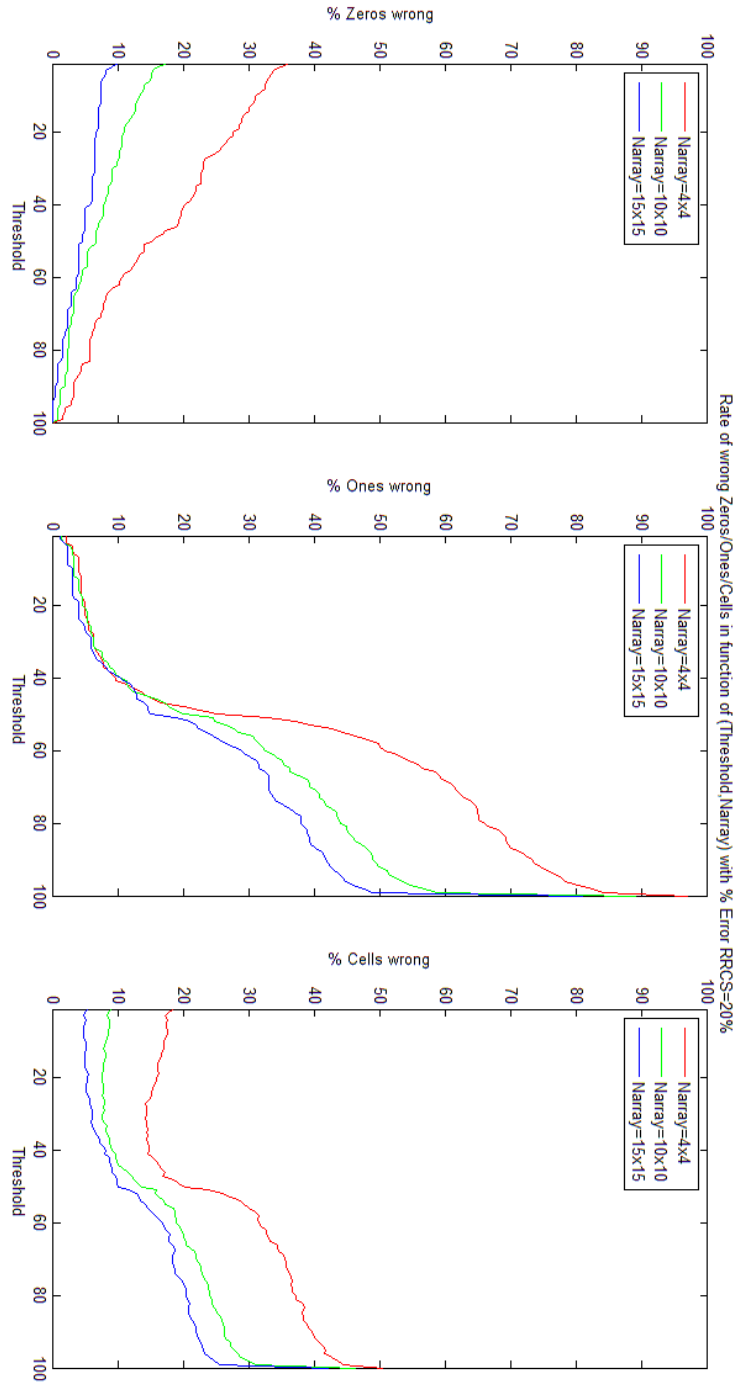


Figura 6.10: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 20\%RRCs$  al variare di  $N_{array}$

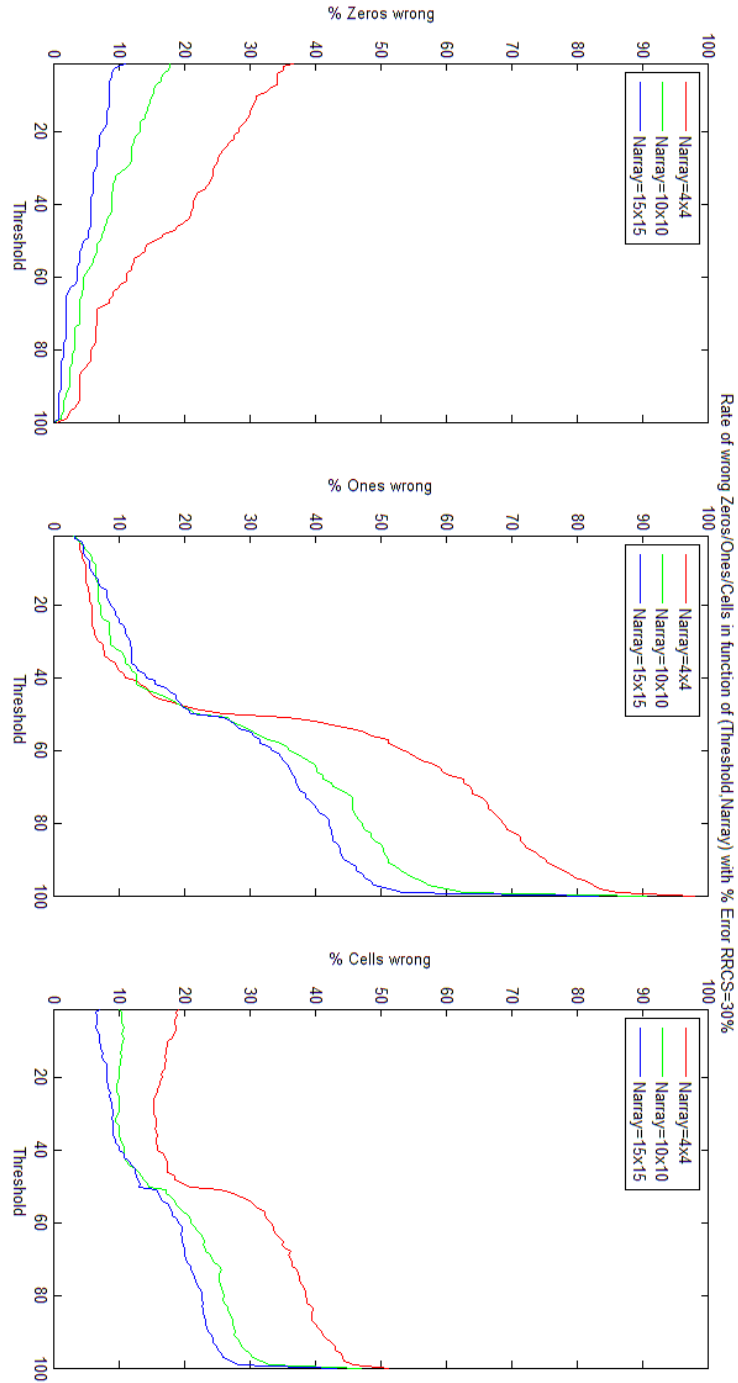


Figura 6.11: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 30\%RRCs$  al variare di  $N_{array}$

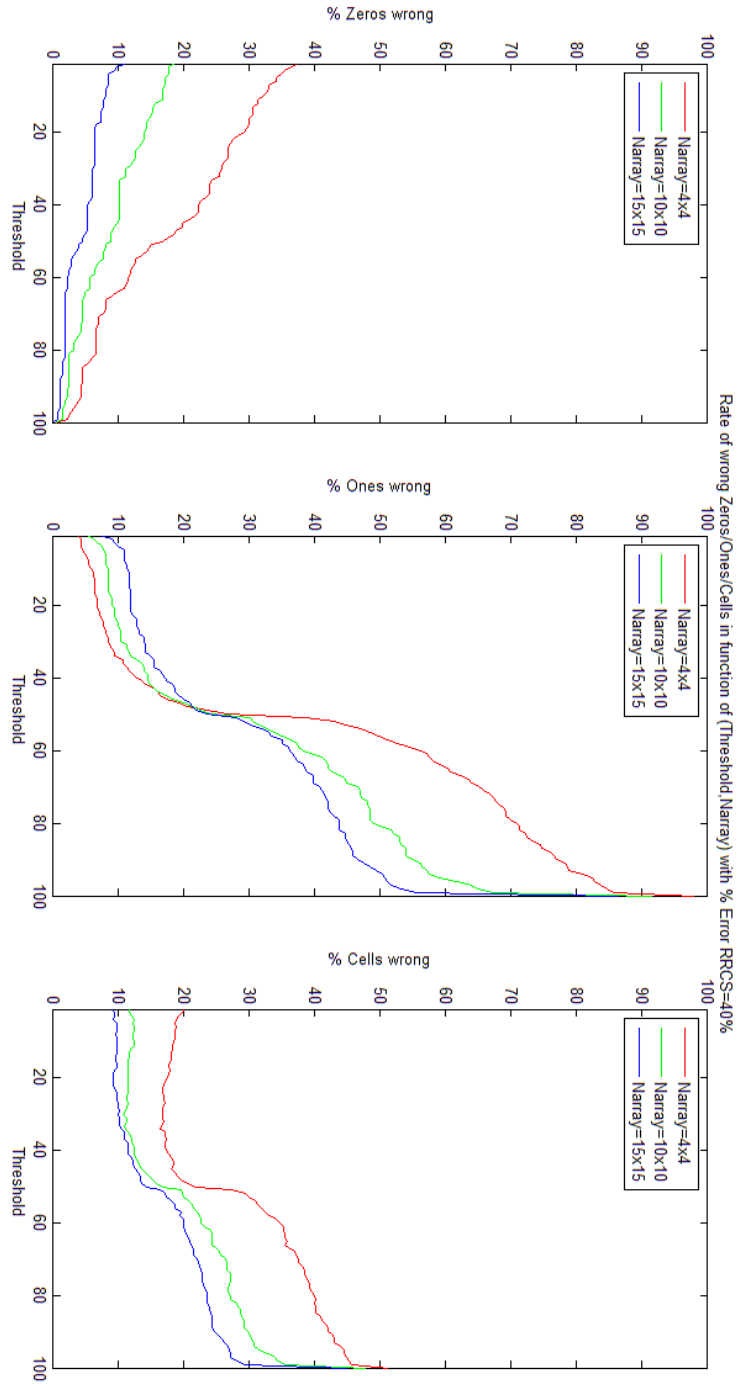


Figura 6.12: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 40\%RRCs$  al variare di  $N_{array}$

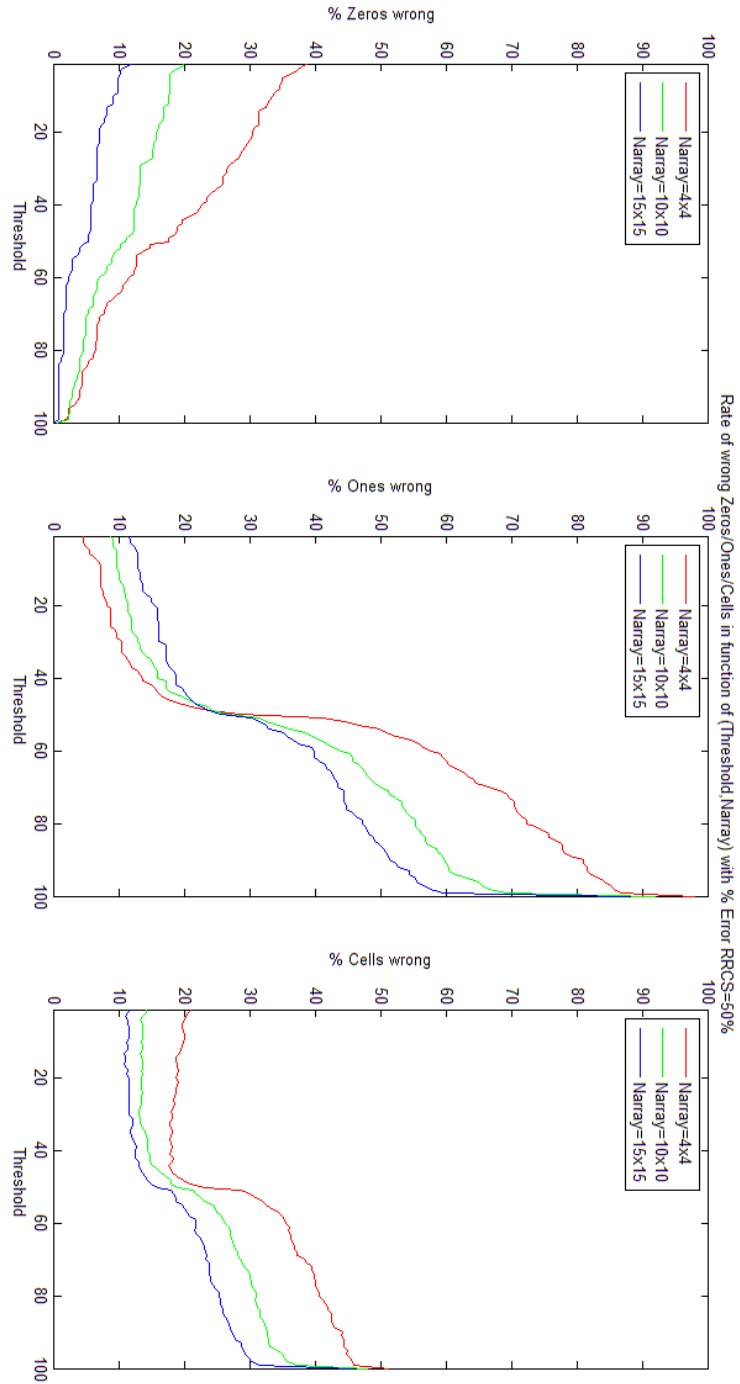


Figura 6.13: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 50\%RRCS$  al variare di  $N_{array}$

6.1.2 Grafici della mappa stimata relativa al terzo scenario

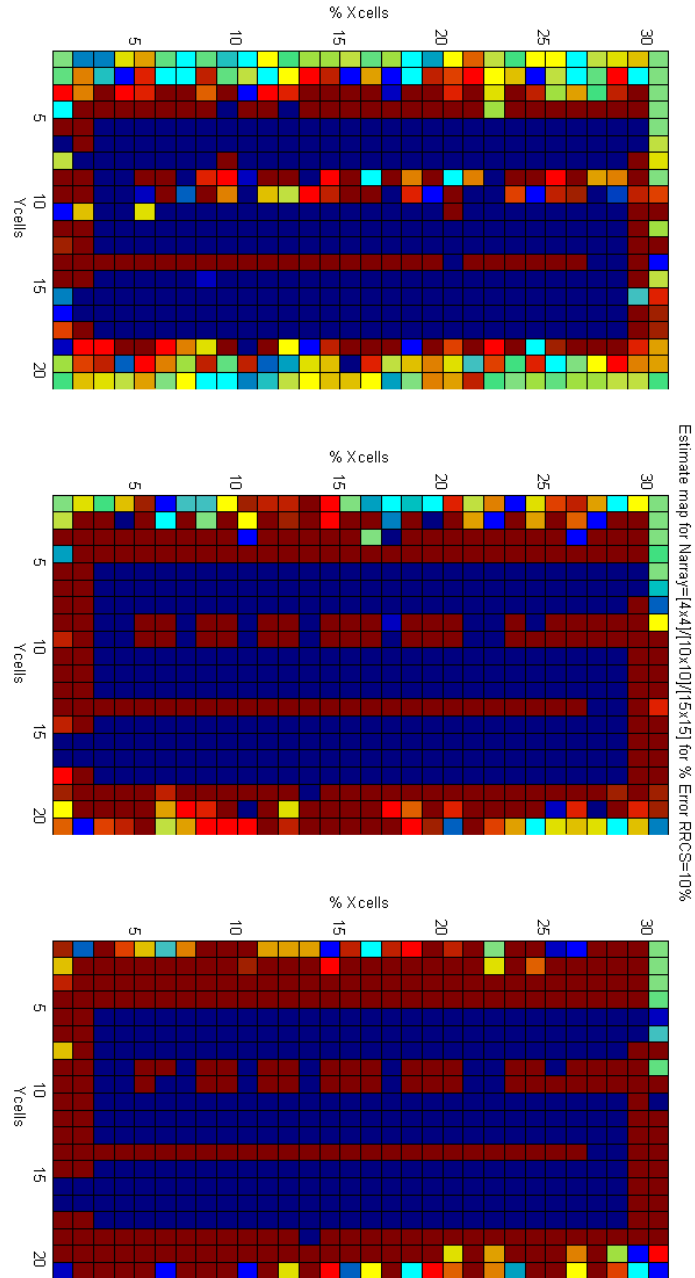


Figura 6.14: Mappa stimata per  $\sigma_e^2 = 10\%$  RRCS e al variare del numero di antenne (da sx a dx: 16, 100, 225)

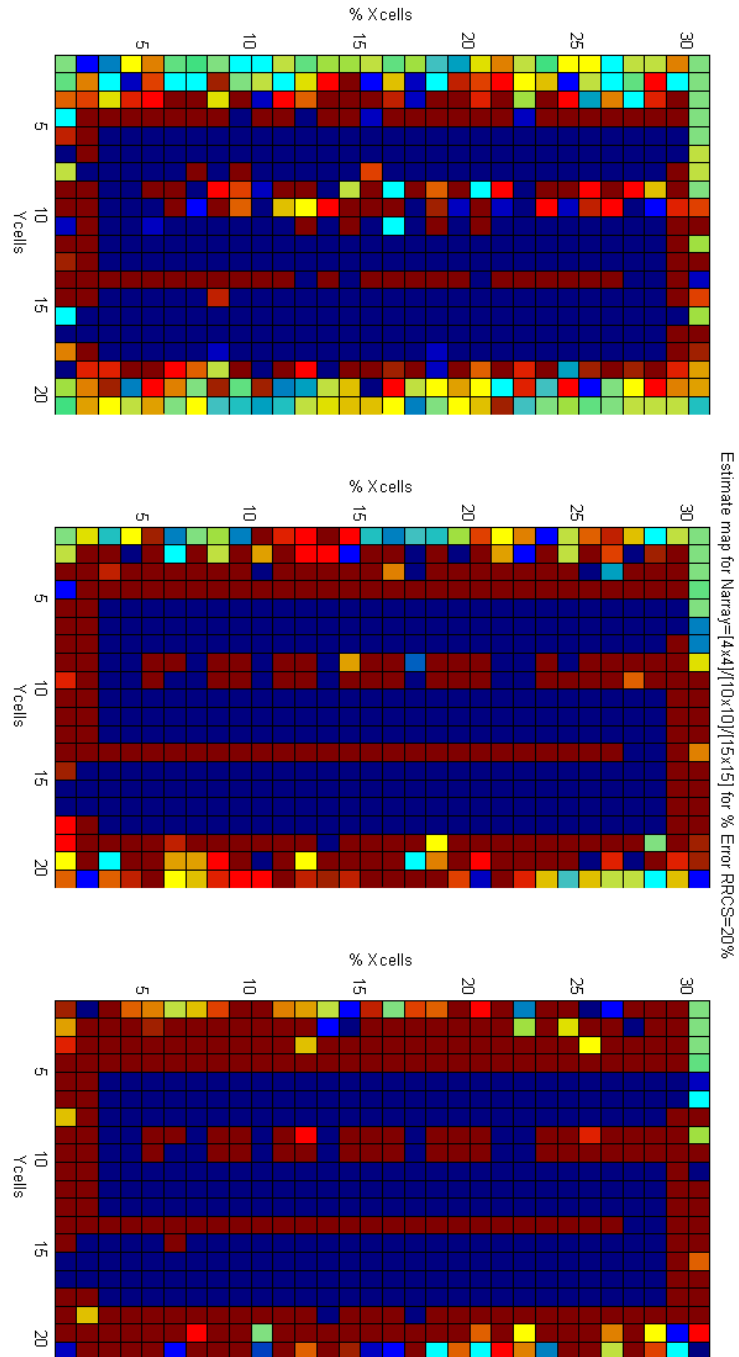


Figura 6.15: Mappa stimata per  $\sigma_e^2 = 20\%$  RRCS e al variare del numero di antenne (da sx a dx: 16, 100, 225)

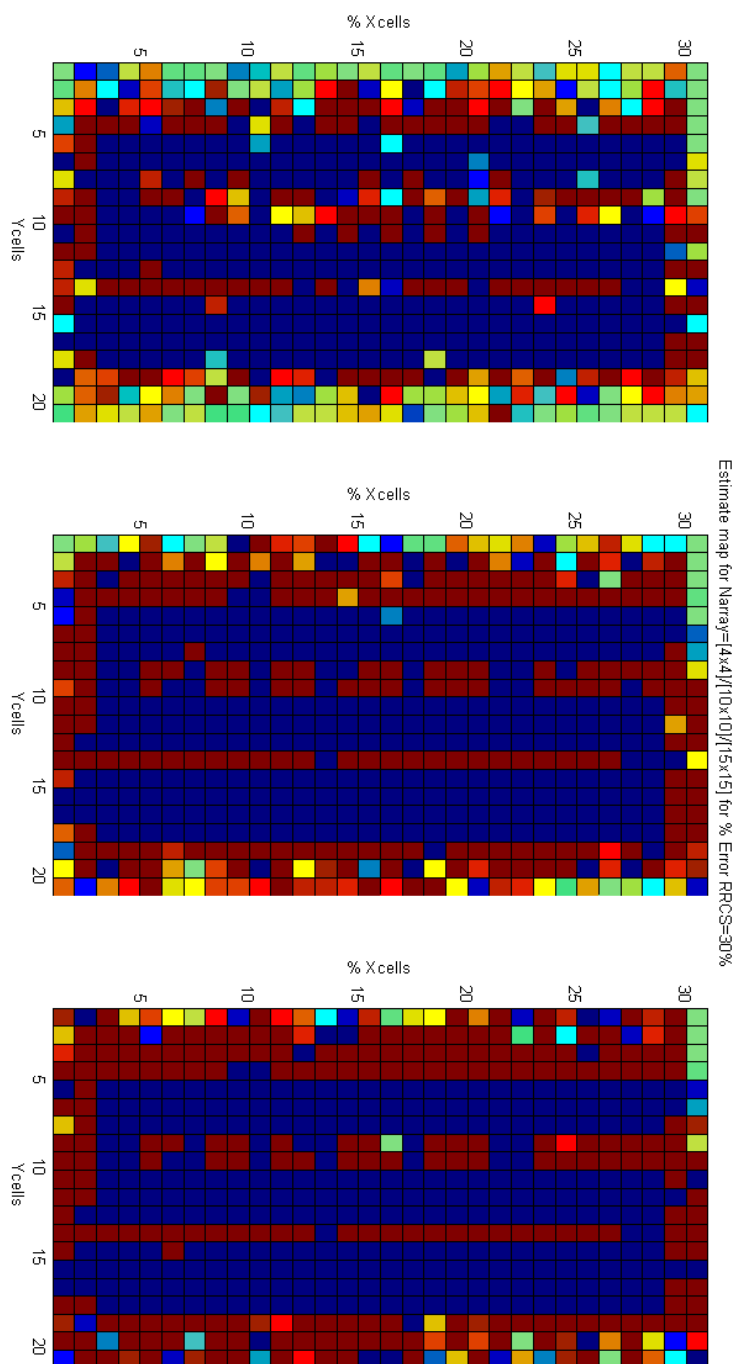


Figura 6.16: Mappa stimata per  $\sigma_e^2 = 30\%$  RRCS e al variare del numero di antenne (da sx a dx: 16, 100, 225)

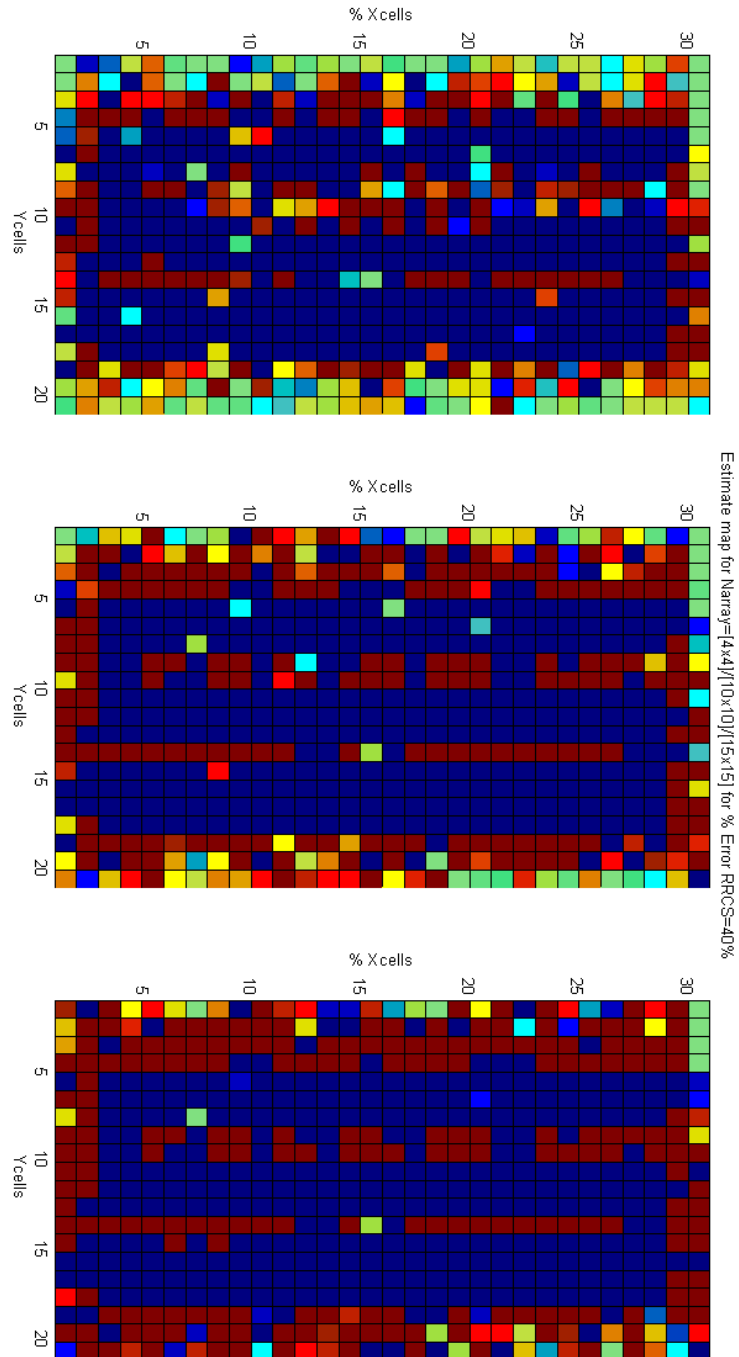


Figura 6.17: Mappa stimata per  $\sigma_e^2 = 40\%$  RRCS e al variare del numero di antenne (da sx a dx: 16, 100, 225)



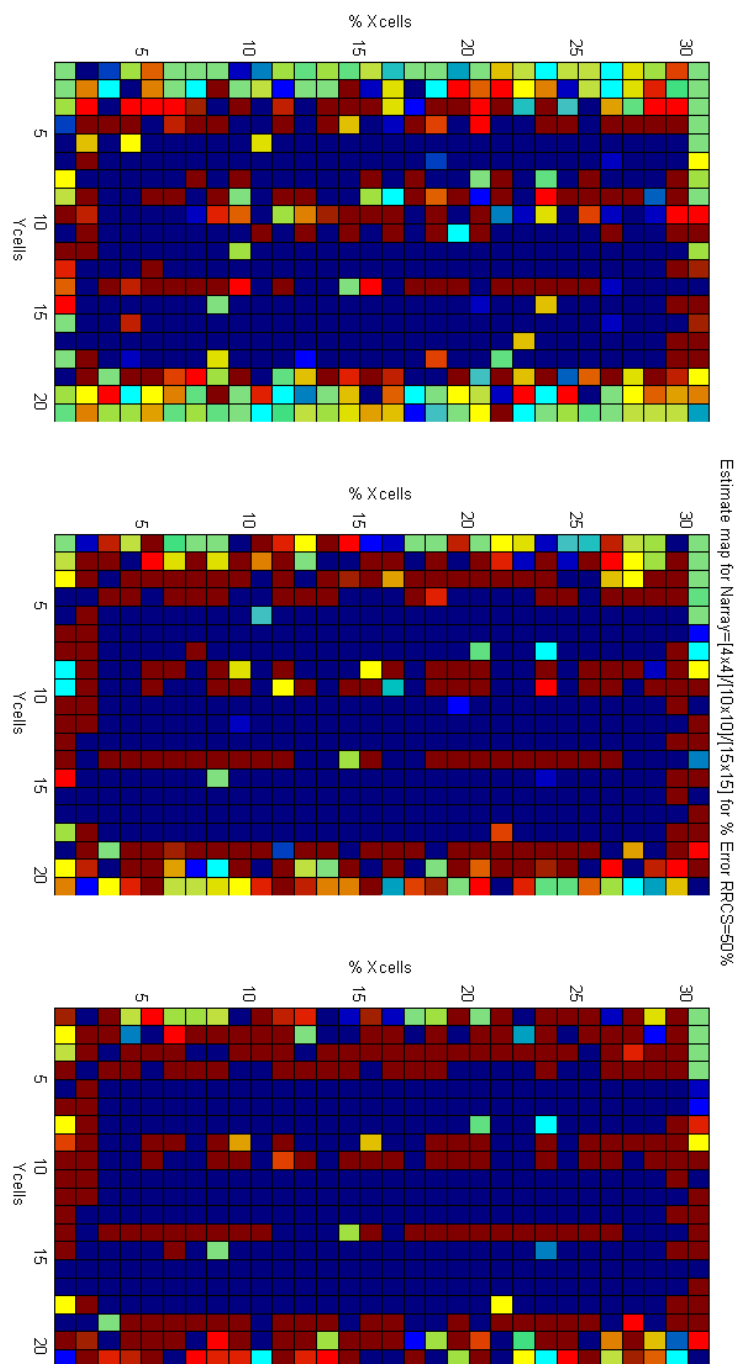


Figura 6.18: Mappa stimata per  $\sigma_e^2 = 50\%$  RRCS e al variare del numero di antenne (da sx a dx: 16, 100, 225)

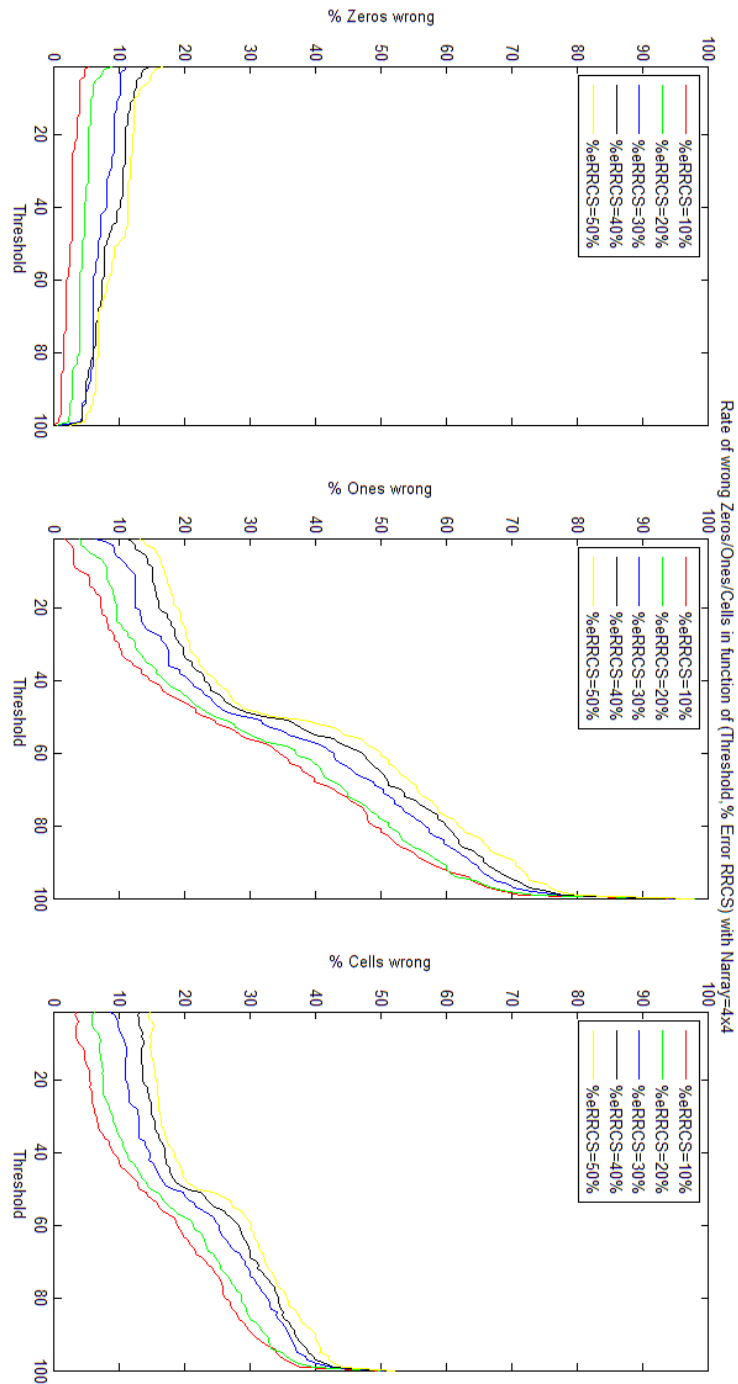


Figura 6.19: % di celle errate in funzione della soglia per  $N_{\text{array}} = 16$  al variare di  $\sigma_c^2$

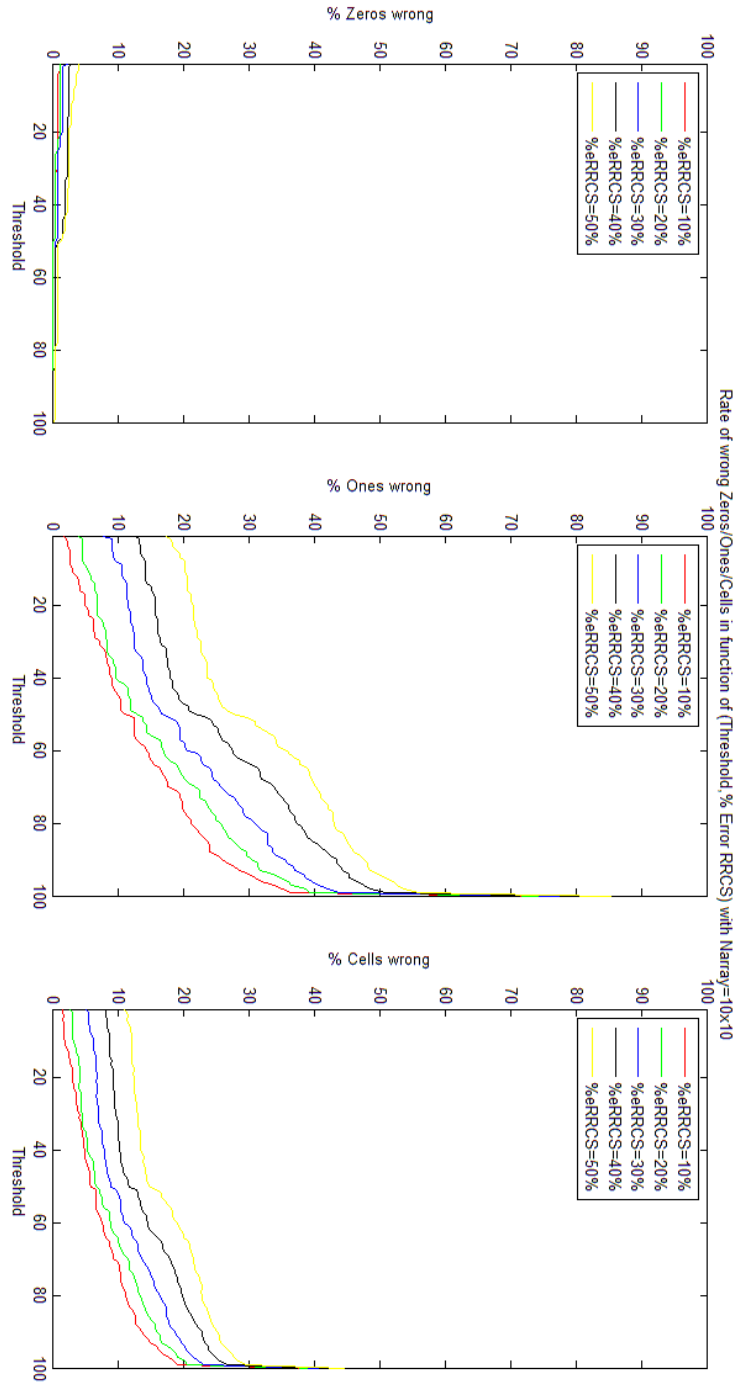


Figura 6.20: % di celle errate in funzione della soglia per  $N_{array} = 100$  al variare di  $\sigma_c^2$

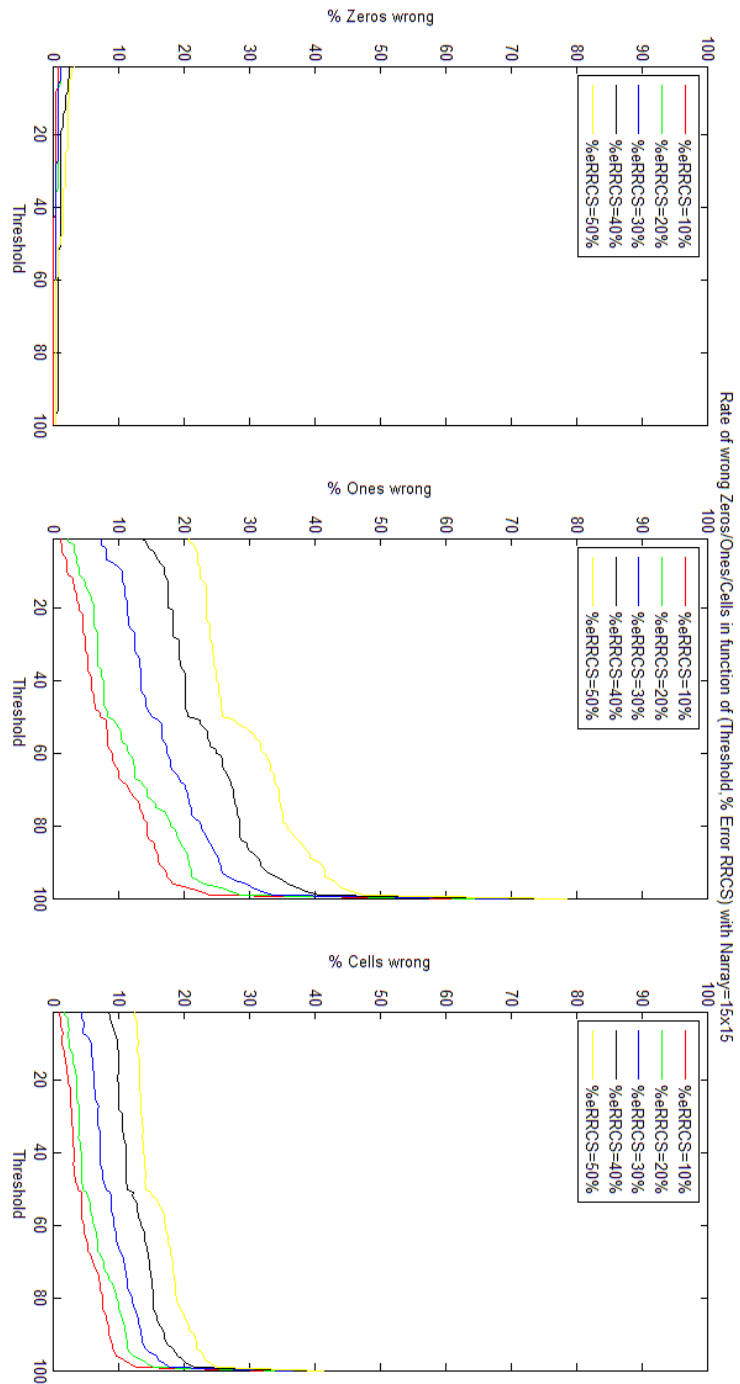


Figura 6.21: % di celle errate in funzione della soglia per  $N_{\text{array}} = 225$  al variare di  $\sigma_c^2$

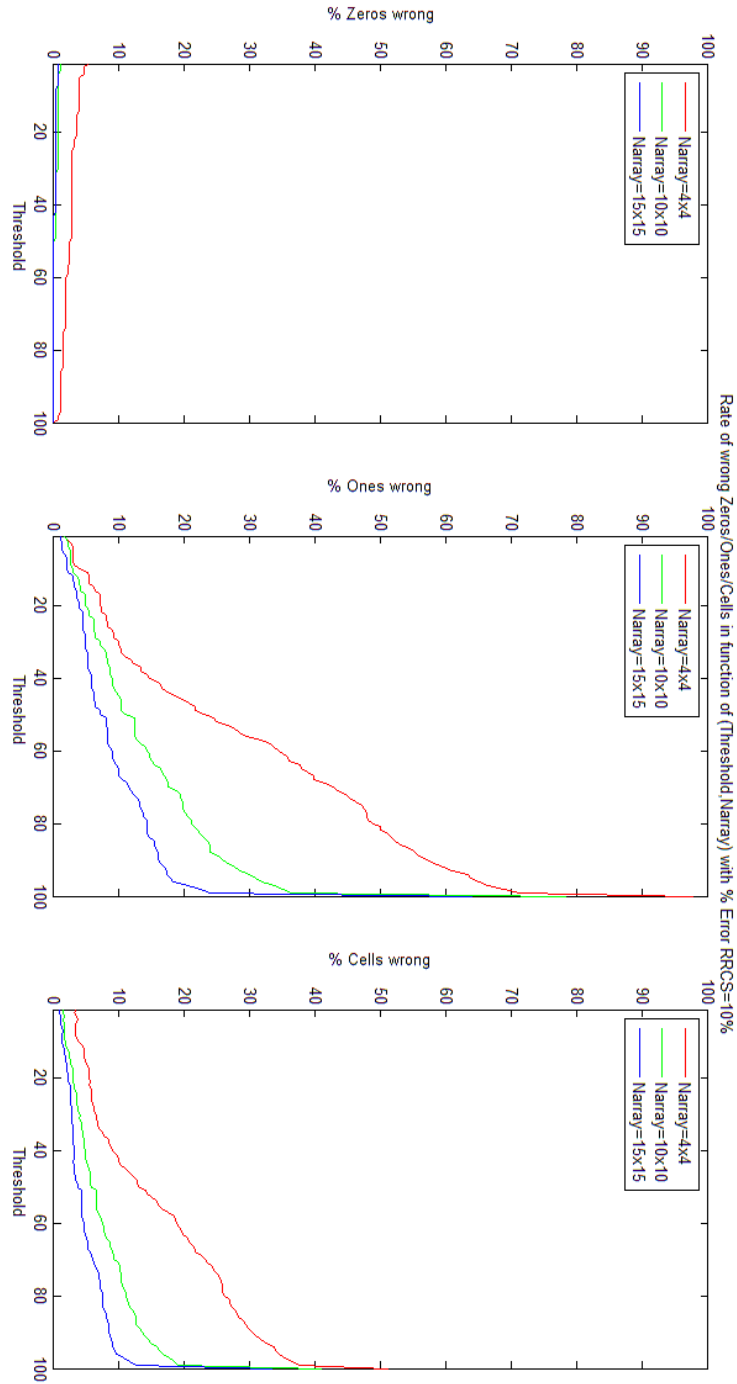


Figura 6.22: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 10\%RRCS$  al variare di  $N_{array}$

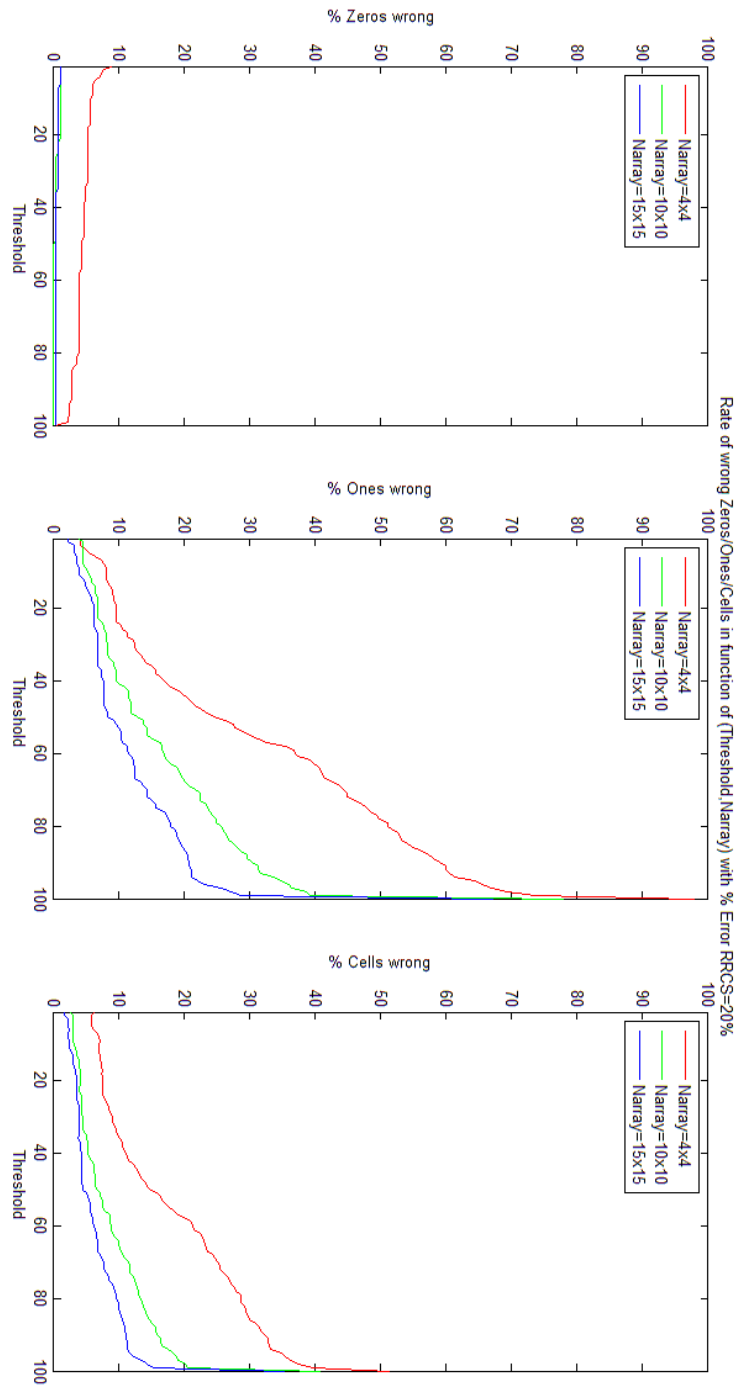


Figura 6.23: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 20\%RRCs$  al variare di  $N_{array}$

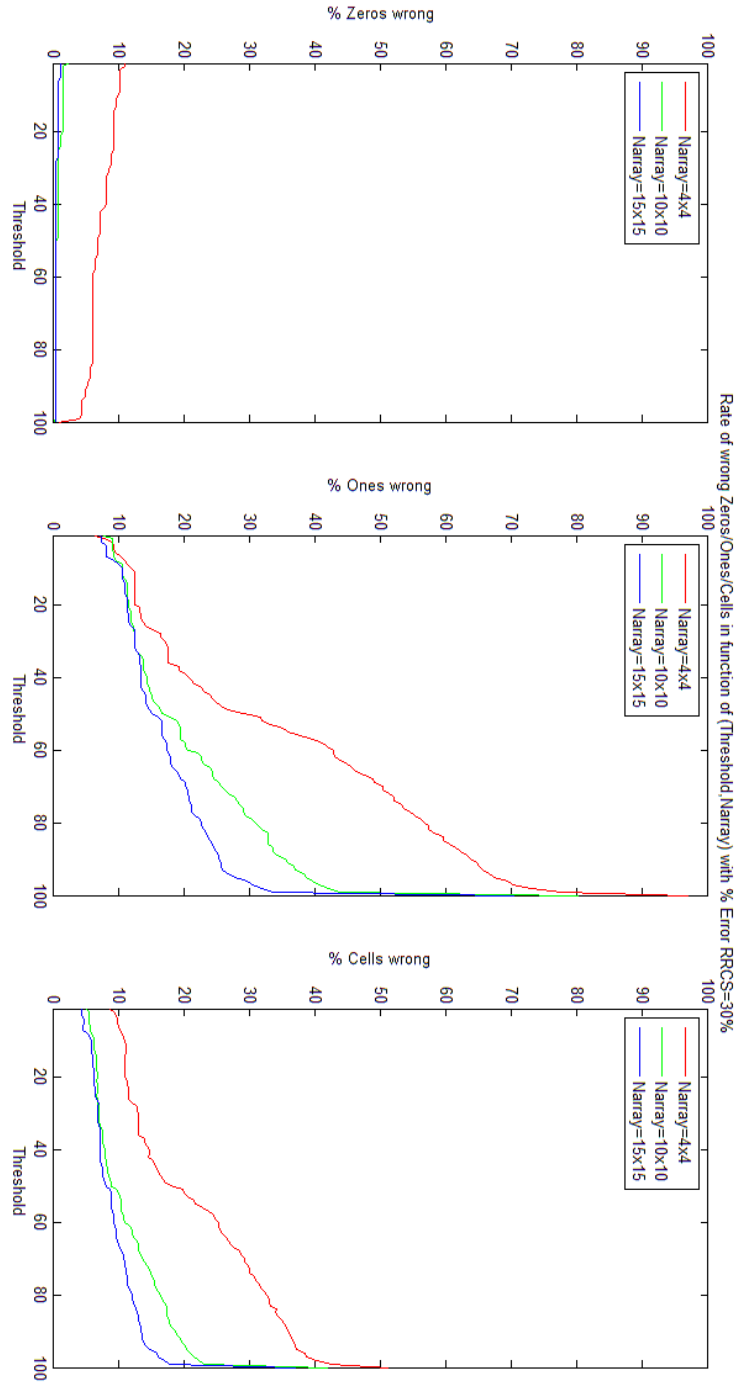


Figura 6.24: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 30\%RRCs$  al variare di  $N_{array}$

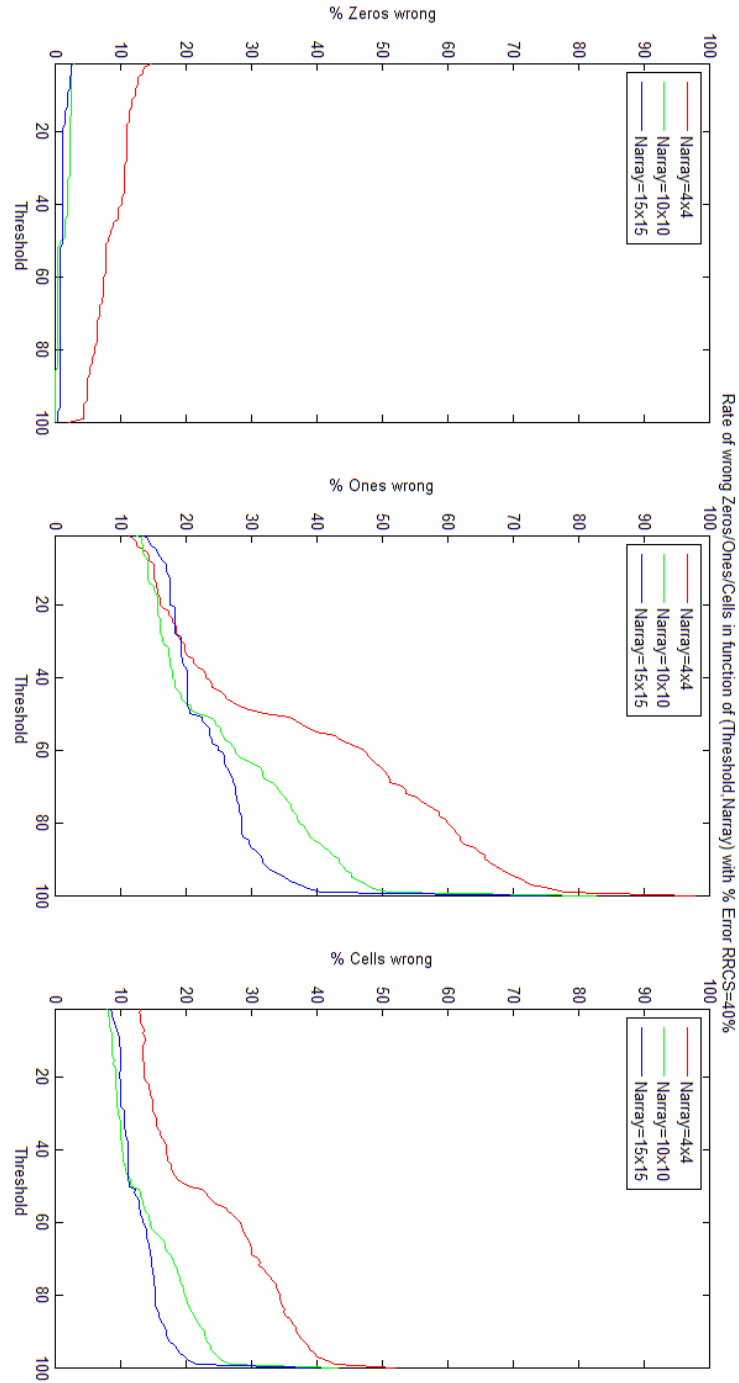


Figura 6.25: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 40\%RRCs$  al variare di  $N_{array}$



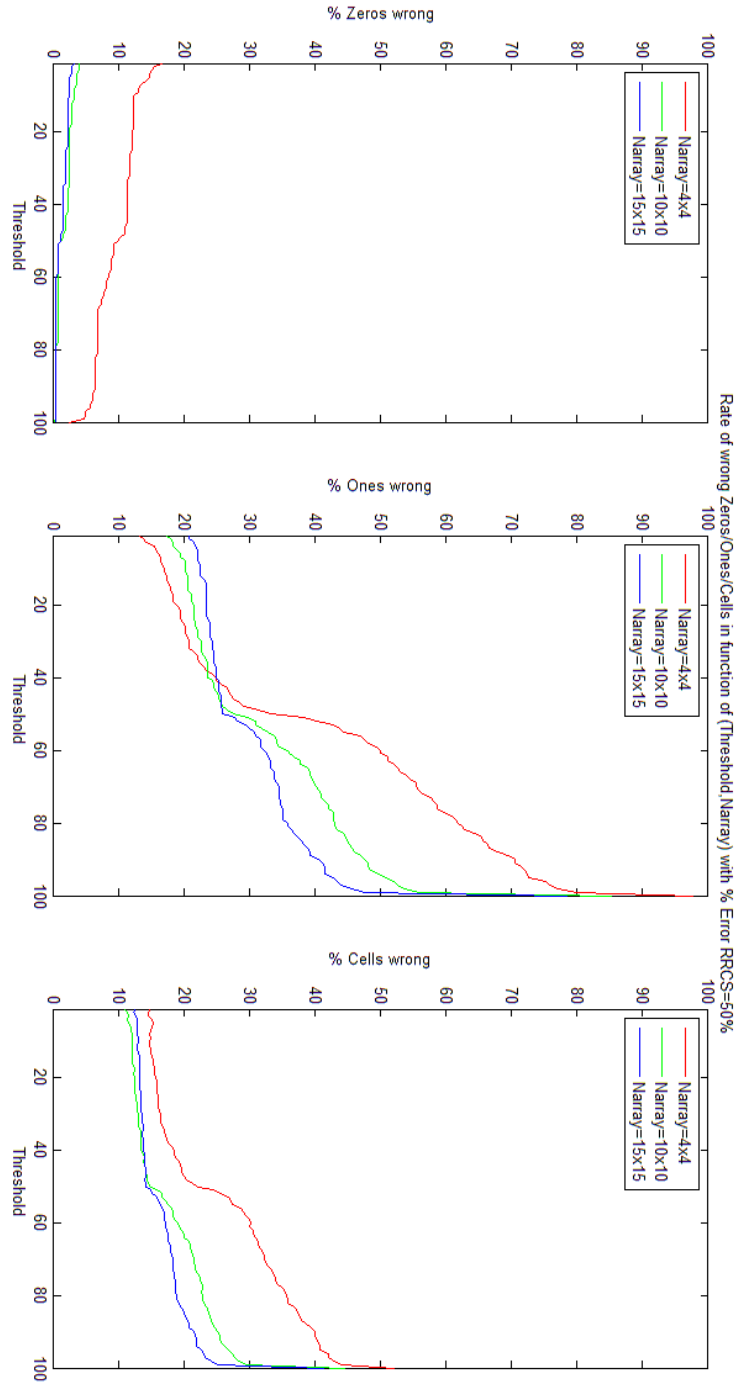


Figura 6.26: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 50\%RRCs$  al variare di  $N_{array}$

## 6.2 Grafici per i restanti confronti effettuati

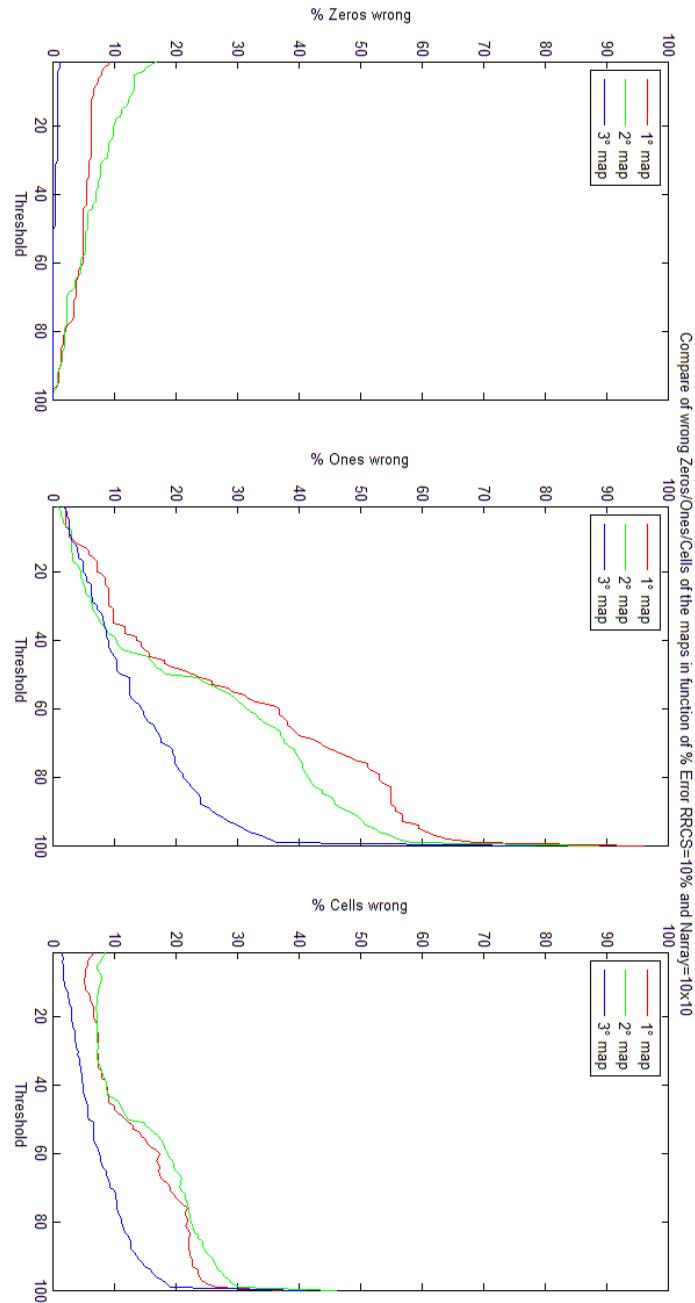


Figura 6.27: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 10\%RRCS$  e  $N_{array} = 100$

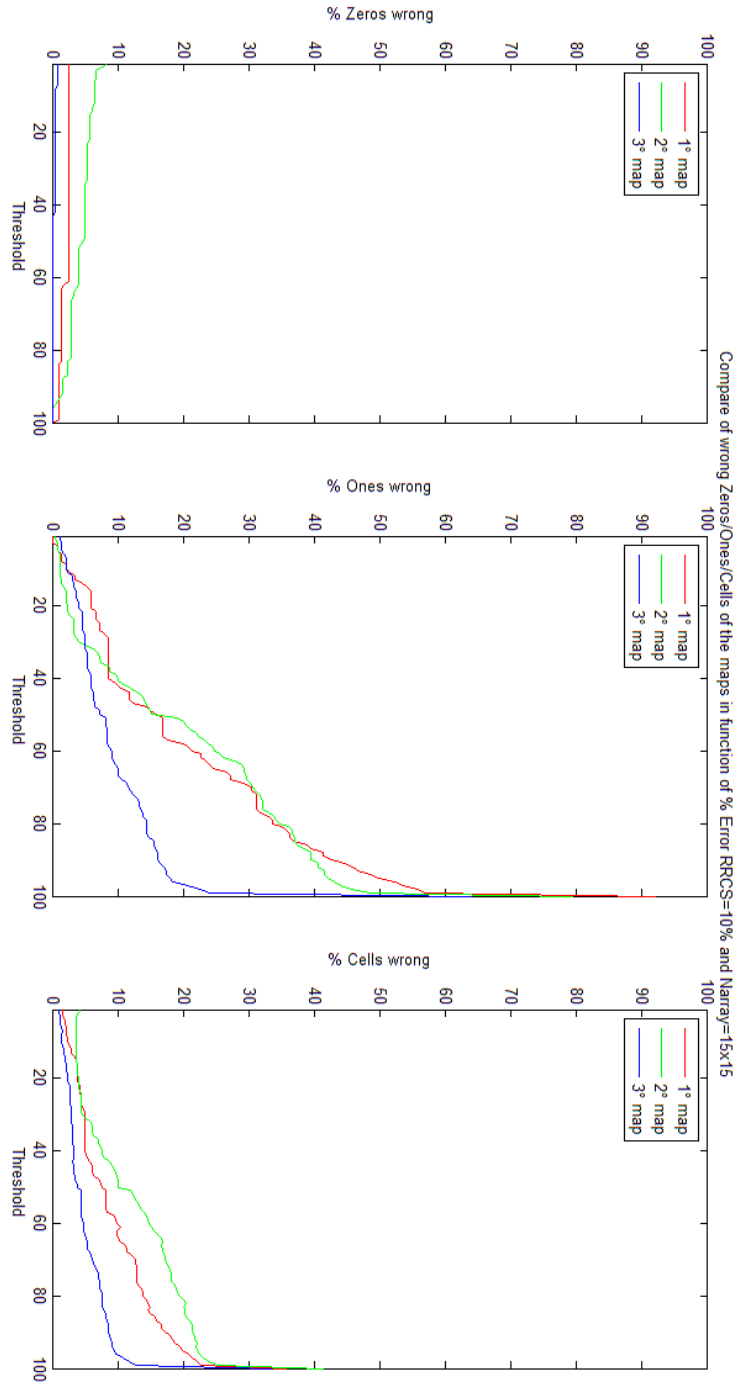


Figura 6.28: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 10\%RRCS$  e  $N_{array} = 225$

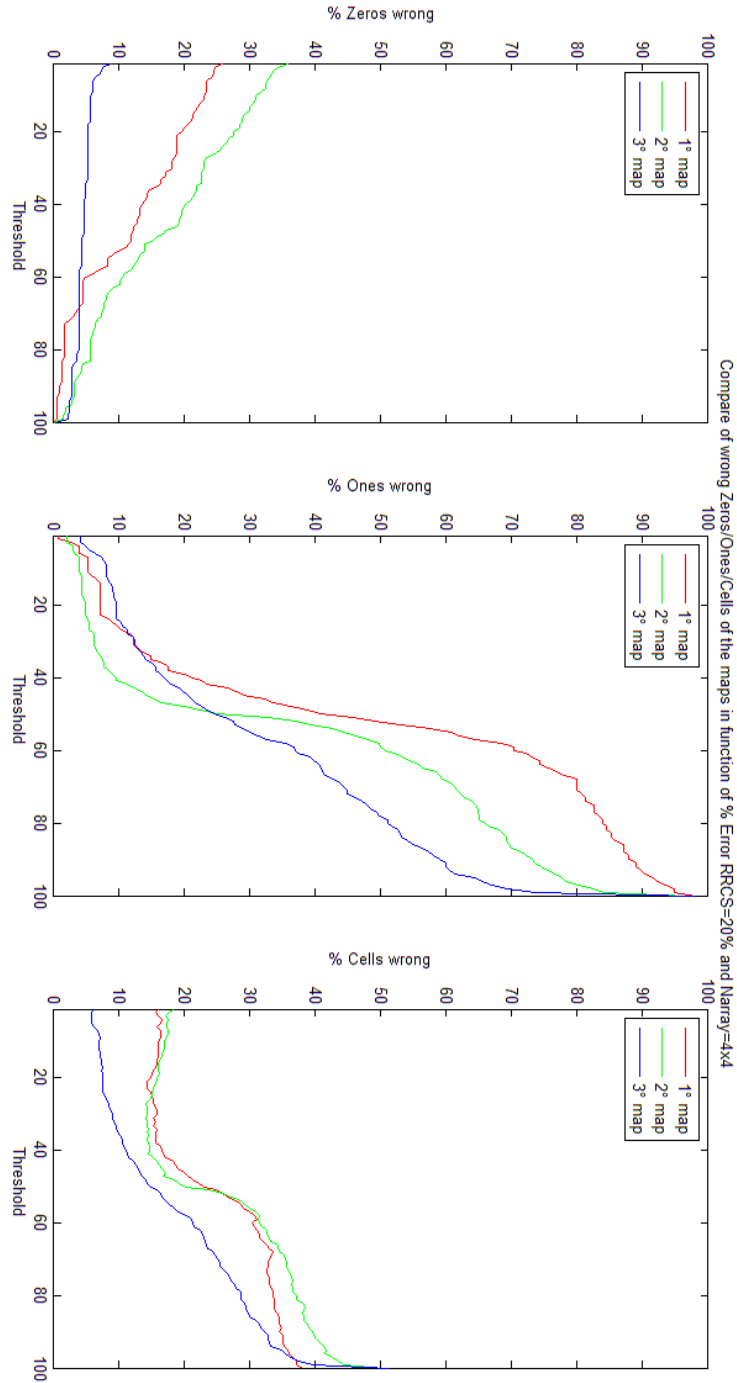


Figura 6.29: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 20\%RRCS$  e  $N_{array} = 16$

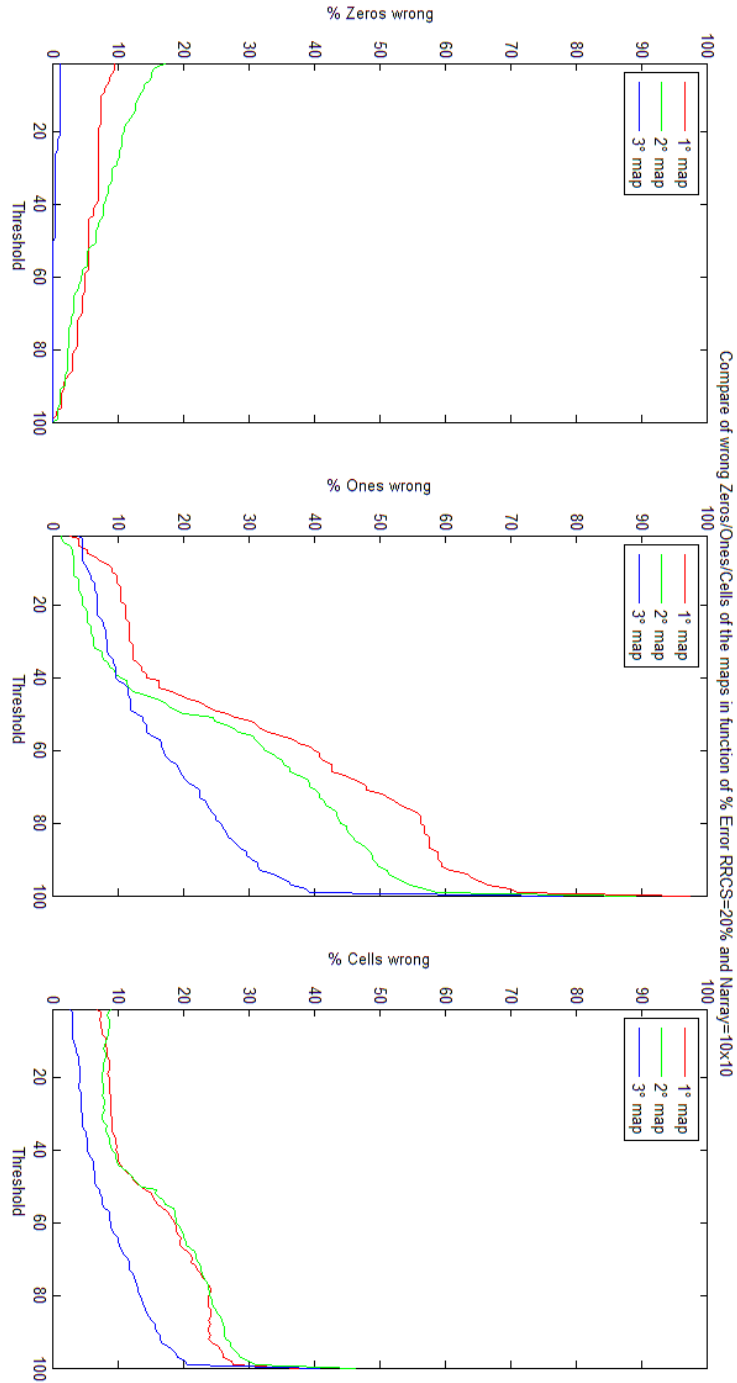


Figura 6.30: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 20\%RRCS$  e  $N_{array} = 100$

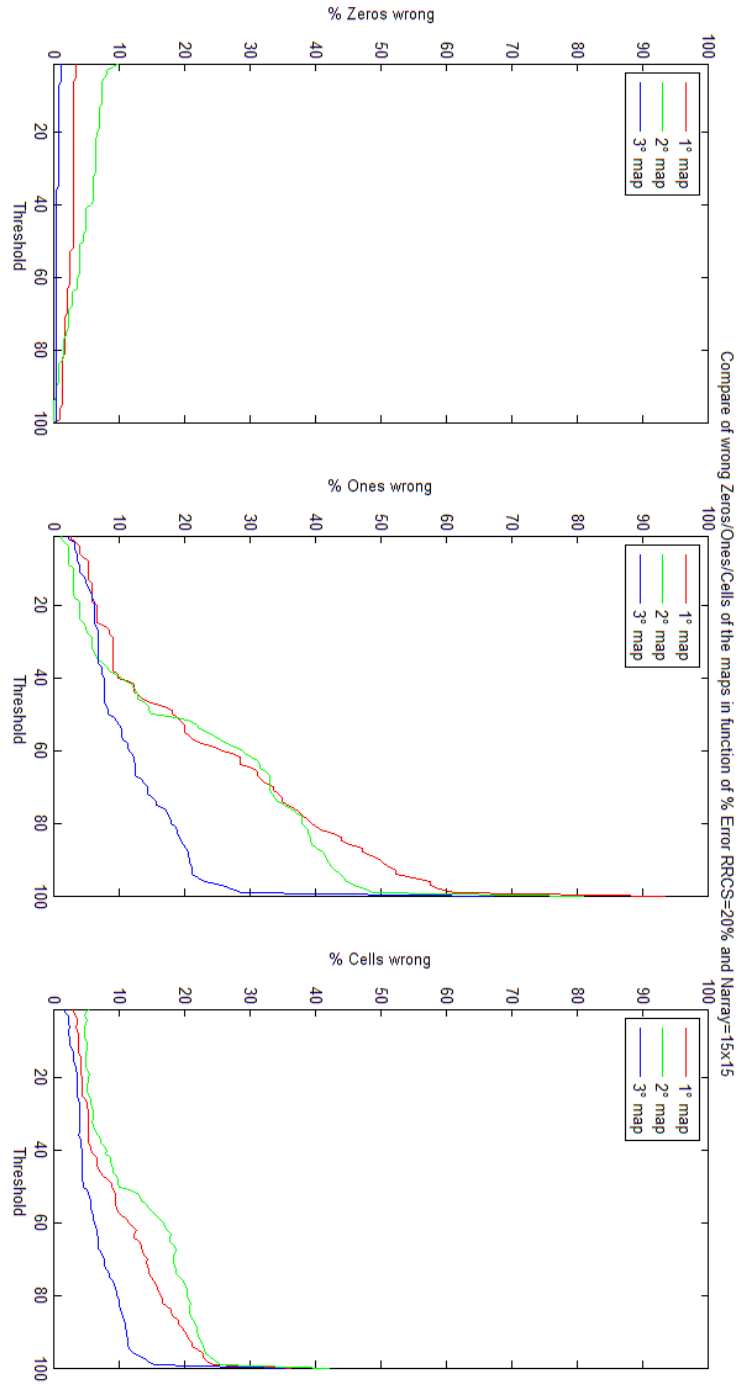


Figura 6.31: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 20\%RRCS$  e  $N_{array} = 225$

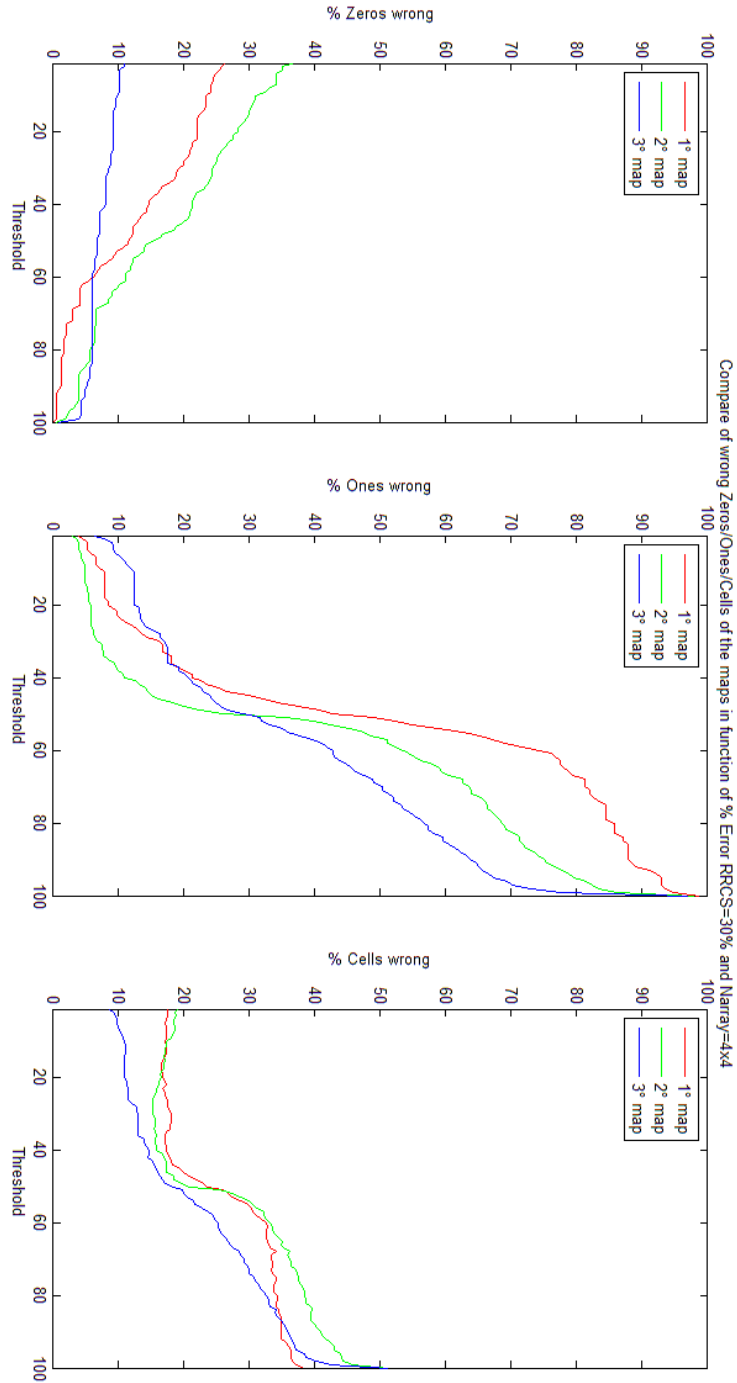


Figura 6.32: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 30\%RRCS$  e  $N_{array} = 16$

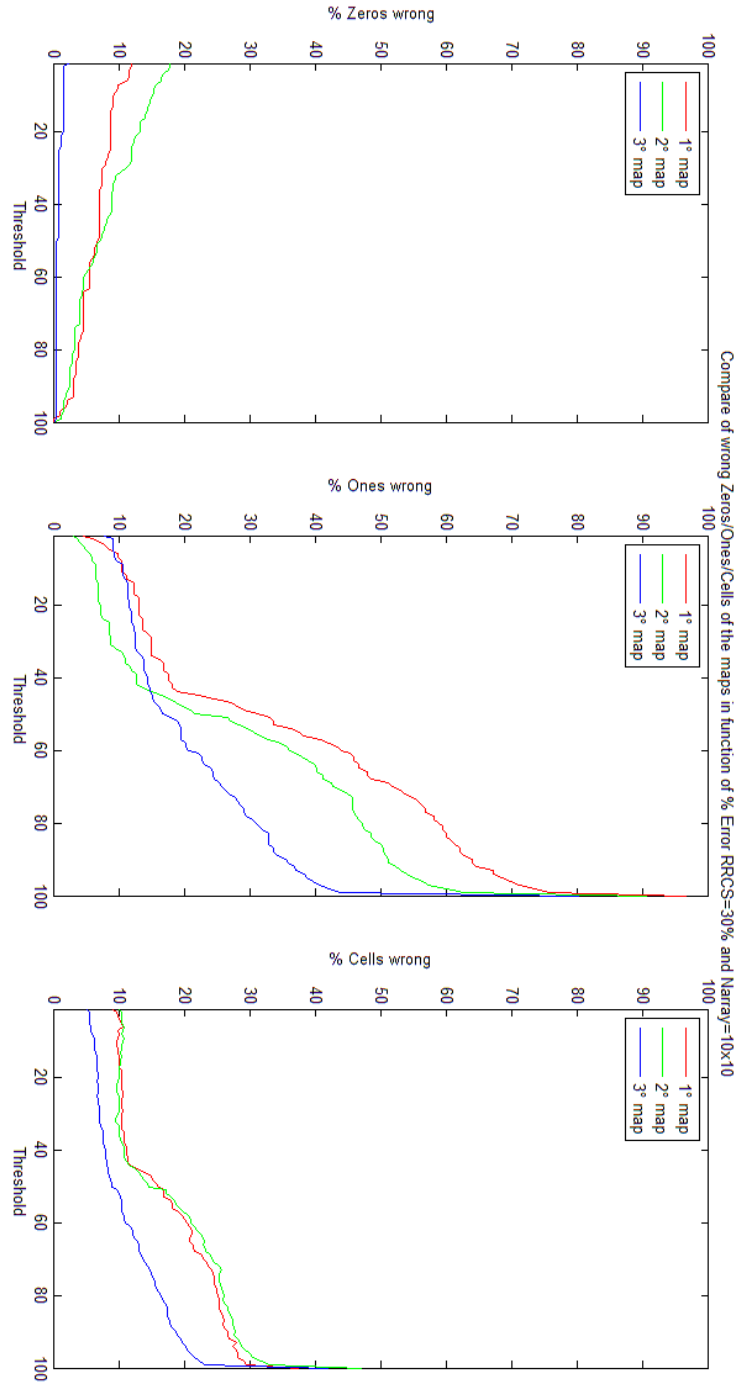


Figura 6.33: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 30\%$ RRCS e  $N_{array} = 100$



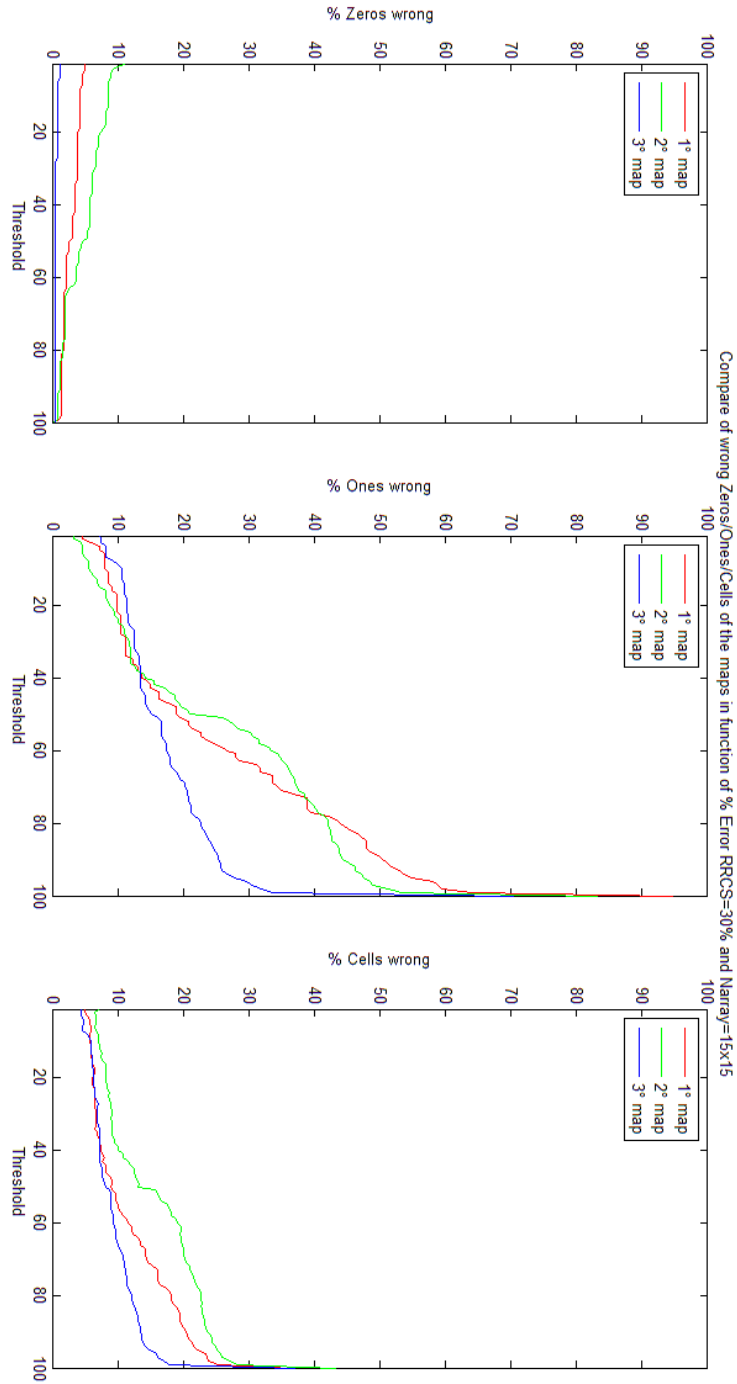


Figura 6.34: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 30\%RRCS$  e  $N_{array} = 225$

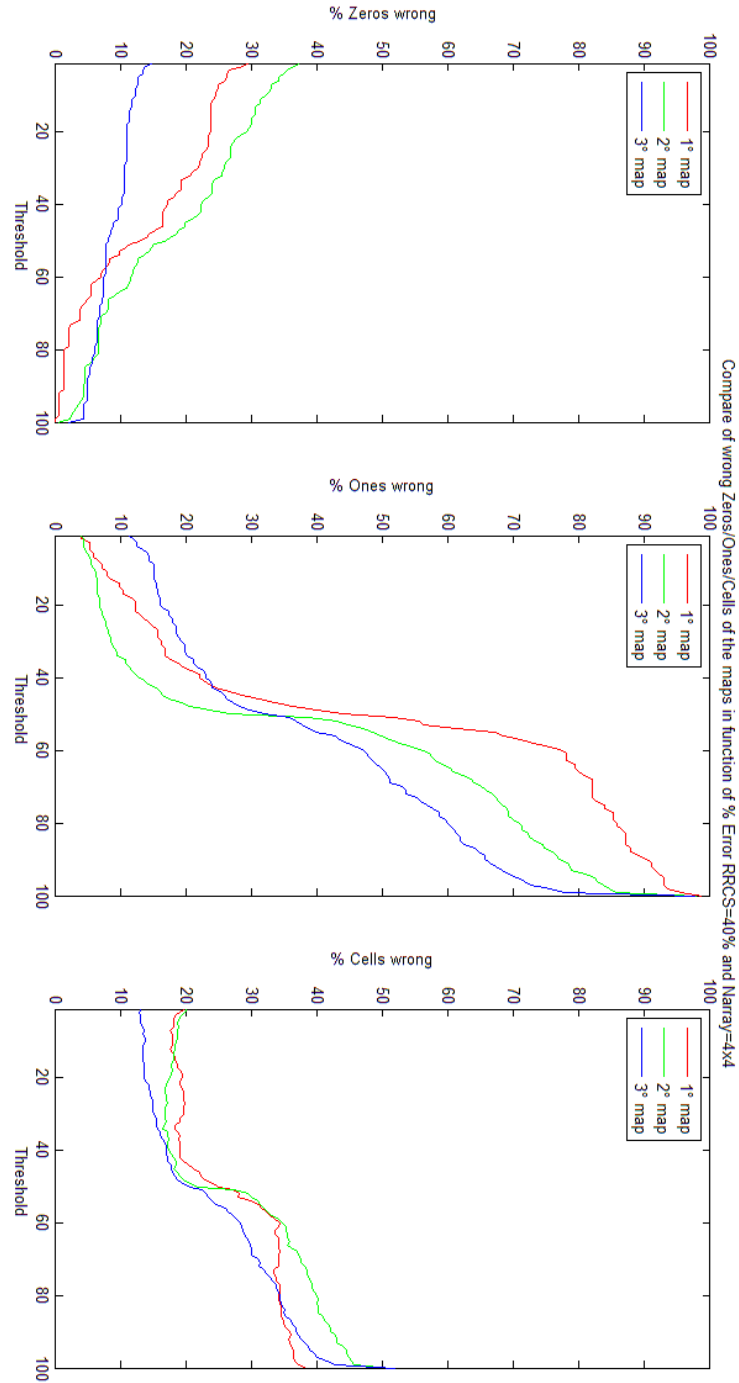


Figura 6.35: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 30\%$ RRCS e  $N_{\text{array}} = 16$

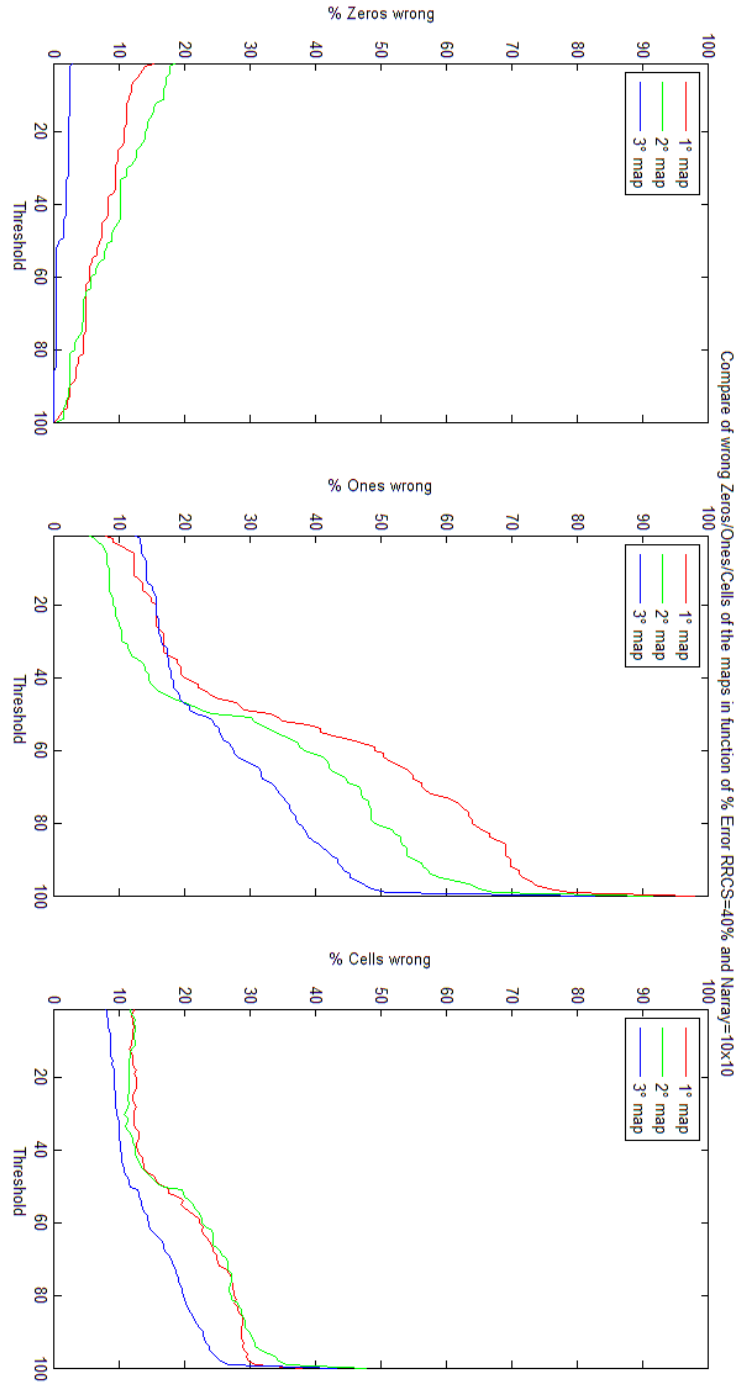


Figura 6.36: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 40\%RRCS$  e  $N_{array} = 100$

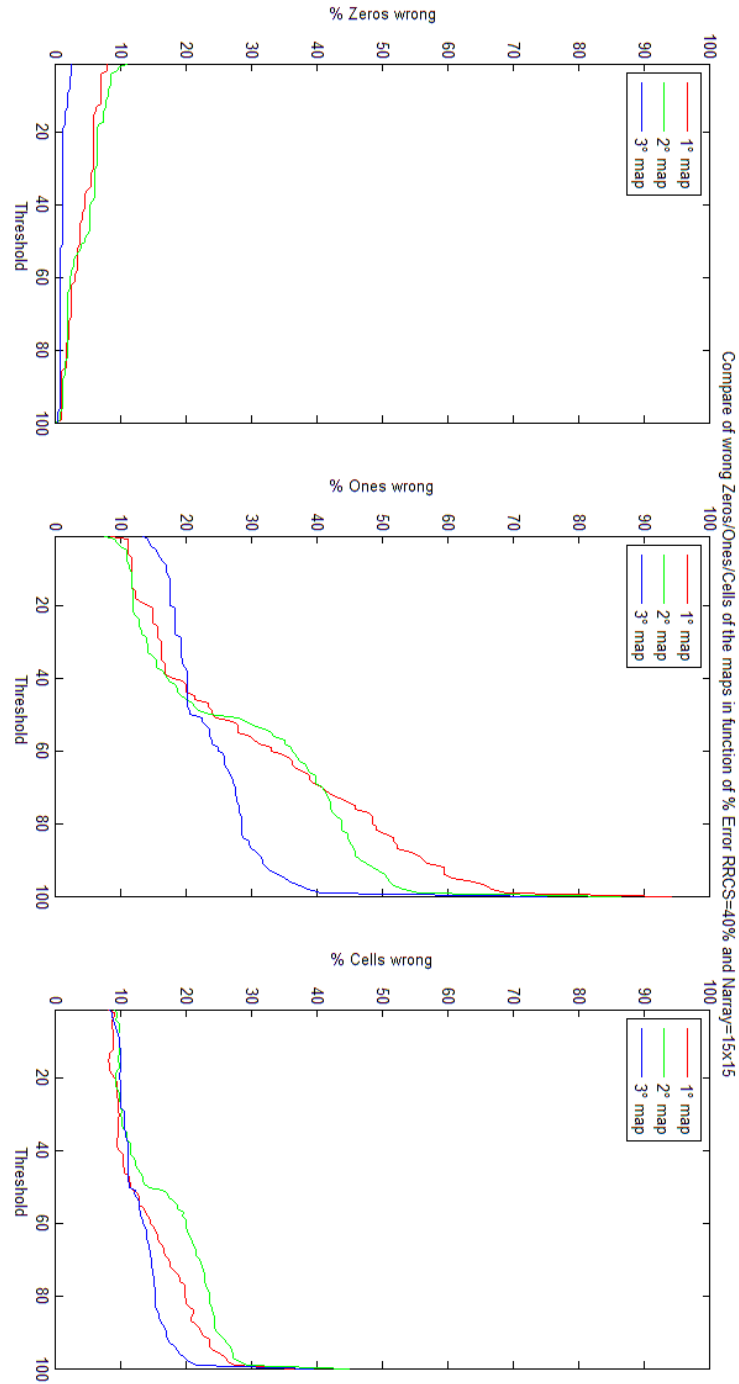


Figura 6.37: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 40\%RRCS$  e  $N_{array} = 225$

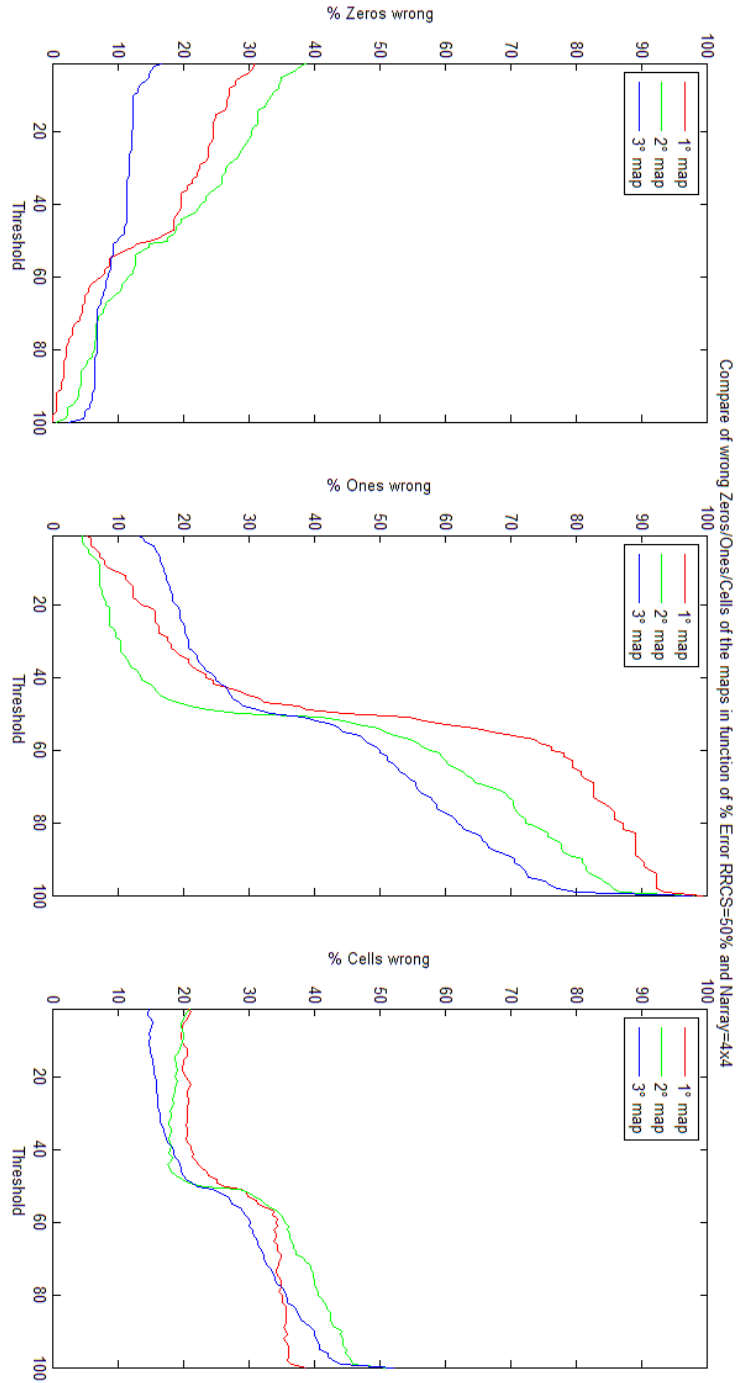


Figura 6.38: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 50\%RRCS$  e  $N_{array} = 16$

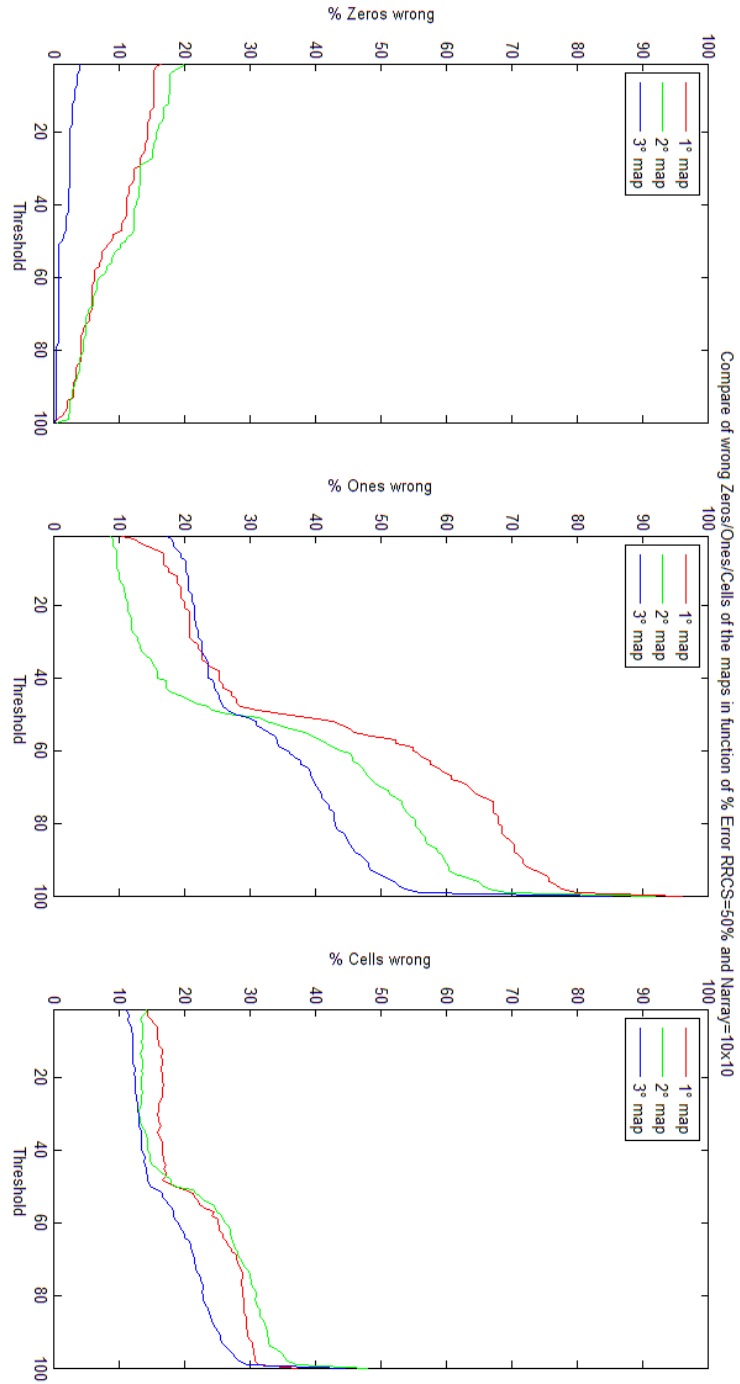


Figura 6.39: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 50\%RRCS$  e  $N_{array} = 100$

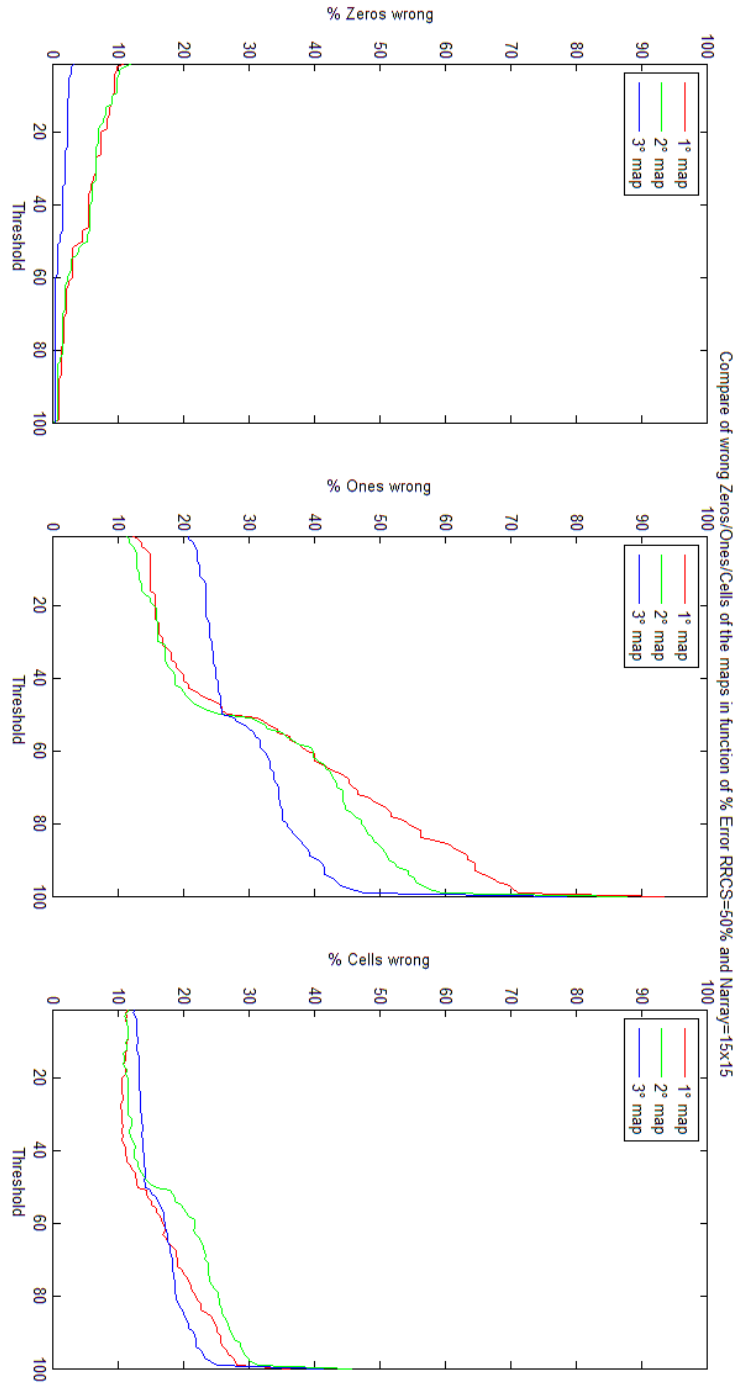


Figura 6.40: Percentuale di celle sbagliate in funzione della soglia con  $\sigma_e^2 = 50\%RRCS$  e  $N_{array} = 225$





# Capitolo 7

## Appendice 2 - Codice MatLab

- *Diagramma di Irradiazione*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% array factor %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function a= ArrayFactor(theta,theta0,M,N,k0,dx)
    a=zeros(1,length(theta));
    for m=1:M
        for n=1:N
            a=a+exp(1i*(m-1)*dx*(k0*cos(theta)-k0*cos(theta0)));
        end
    end
end
end
```

- *Visibilità celle*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% Get visible landmarks %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [flag_ISvisible,real_angle,d]= get_visible_...
    landmarks(Xcell,Ycell,MAXRANGE,Xmobile,Ymobile,OrienMobile)
    flag_ISvisible=0;
    real_angle=0;

    dx=Xmobile-Xcell;
    dy=Ymobile-Ycell;
    phi=OrienMobile;
    d=sqrt(dx^2+dy^2);
    if d<MAXRANGE
```

```

%calcolo angolo della cella
if dx<0 && dy<0 % primo quadrante
    alfa=(180/pi)*atan2(-dy,-dx);
elseif dx>0 && dy<0 % secondo quadrante
    alfa=180-(180/pi)*atan2(-dy,dx);
elseif dx>0 && dy>0 % terzo quadrante
    alfa=180+(180/pi)*atan2(dy,dx);
elseif dx<0 && dy>0 % quarto quadrante
    alfa=360-(180/pi)*atan2(dy,-dx);
elseif dx==0 && dy<0
    alfa=90;
elseif dx==0 && dy>0
    alfa=270;
elseif dx<0 && dy==0
    alfa=0;
elseif dx>0 && dy==0
    alfa=180;
end

%calcolo visibilit\'a cella
if phi>=90&&phi<=270
    lim_destra=phi-90;
    lim_sinistra=phi+90;
    if alfa>=lim_destra && alfa<=lim_sinistra
        flag_ISvisible=1;
        real_angle=alfa-lim_destra;
    end
end
if phi==0 || phi==360
    lim_destra=270;
    lim_sinistra=90;
    if alfa>=lim_destra || alfa<=lim_sinistra
        flag_ISvisible=1;
        if alfa<=90
            real_angle=alfa+90;
        else
            real_angle=alfa-lim_destra;
        end;
    end
end
if phi>0 && phi<90
    lim_sinistra=phi+90;
    lim_destra=270+phi;
    if alfa>=lim_destra || alfa<=lim_sinistra

```

```

        flag_ISvisible=1;
        if alfa<=180
            real_angle=alfa+90-phi;
        else
            real_angle=alfa-lim_destra;
        end
    end
end
if phi>270 && phi<360
    lim_sinistra=phi-270;
    lim_destra=phi-90;
    if alfa>=lim_destra || alfa<=lim_sinistra
        flag_ISvisible=1;
        if alfa<=180
            real_angle=alfa+450-phi;
        else
            real_angle=alfa-lim_destra;
        end
    end
end
end
end
end
end

```

- *Parametri costanti*

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Nms=3; %Numero degli stati per il mobile (x,y,orientamento)
Nfs=1; %Numero di stati per le celle (pieno o vuota)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f0=60*1e9;
c=3e8;
lambda0=c/f0;
k0=2*pi/lambda0;
adx=lambda0/2;

W=1e9;
Ted=1/W;
Tf=30e-9;
Nbins=floor(Tf/Ted);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Narray=zeros(1,N_iterations_algorithm);
for i=1:N_iterations_algorithm
    Narray(1,i)=Nx(1,i)*Ny(1,i);
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:N_iterations_algorithm
    switch W % BSA
        case 2e9
            if Narray(1,i)==16 || Narray(1,i)==1
                BSA(1,i)=0.1570;
            elseif Narray(1,i)==100
                BSA(1,i)=0.0238;
            elseif Narray(1,i)==225
                BSA(1,i)=0.0105;
            end
        case 1e9
            if Narray(1,i)==16 || Narray(1,i)==1
                BSA(1,i)=0.1595;
            elseif Narray(1,i)==100
                BSA(1,i)=0.0242;
            elseif Narray(1,i)==225
                BSA(1,i)=0.0107;
            end
        case 0.5e9
            if Narray(1,i)==16 || Narray(1,i)==1
                BSA(1,i)=0.1608;
            elseif Narray(1,i)==100
                BSA(1,i)=0.0244;
            elseif Narray(1,i)==225
                BSA(1,i)=0.0108;
            end
    end
end
Dmax=zeros(1,N_iterations_algorithm);
for i=1:N_iterations_algorithm
    Dmax(1,i)=pi^2/BSA(1,i);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
kb=1.38e-23;
F=10^(6/10);
NO=kb*290*F;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dBin=c*Ted/2; % Distanza di un bin[m]
MAXRANGE=dBin*Nbins; % Massima distanza copribile [m]
Np=1; %ceil(Tobs/Tf)
Tobs=Np*Tf; % Tempo di osservazione
Tscan=Tobs*Nsteer; % Tempo di scannerizzazione
Nobs=Nsteer*Nbins; % Numero di osservazioni

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
theta_inc=0; % theta incident angle
tan_delta=0.0491;
epsilon_r=2.26;
n_2=epsilon_r-1i*epsilon_r*tan_delta;
n=sqrt(n_2);
gamma0_2=abs((1-n)/(1+n))^2;
RCS=4*gamma0_2*(Delta^2); %radar cross section
RRCS=sqrt(RCS); %root radar cross section
sigma2=N0*W; %Varianza di rumore

Pw=(sigma2*Ted*Np/Tf); % normalized noise power

RCS0=1; % reference RCS [m^2]
d0=1; % reference distance [m]

L0=EIRP*c^2*(RCS0)/((4*pi)^3*f0^2*(d0^4))*Np; %Formula di Friis

```

- *Parametri variabili*

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% Variable Parameters%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Delta=0.5; % cell side of grid [m]
SizeX=10; % layout size [m]
SizeY=10; % layout size [m]

Xcells=floor(SizeX/Delta); % number of cell on the x-axis
Ycells=floor(SizeY/Delta); % number of cell on the y-axis

Ncells=Xcells*Ycells; % number of landmarks or grid cell
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
px=7.5;
py=0;
po=90;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N_movement_step=22;
Delta_step=1;
N_iterations_algorithm=15;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Nx=[4 10 15 4 10 15 4 10 15 4 10 15 4 10 15];
Ny=[4 10 15 4 10 15 4 10 15 4 10 15 4 10 15];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
perc_var_RRCS=[10 10 10 20 20 20 30 30 30 40 40 40 50 50 50];

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Nsteer=30;
EIRP=1e-4;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

- *Programma completo*

```

clc
clear all
close all

INPUT_Parameters_variable;
INPUT_Parameters_constant;
INPUT_Parameters_noise;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% plot irradiation graphs %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
steer_direction=zeros(1,Nsteer);
for i=1:Nsteer
    steer_direction(i)=(i-1)*pi/Nsteer;
end
point=0:pi/100:pi;
plot(point*180/pi,10*log10(abs(ArrayFactor(point,pi/2,Nx,Ny,k0,adx))));...
    xlabel('Degrees []');ylabel('Gain[dB]');
figure;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% create and plot the map %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[MapLinear,Pos_Xcells,PosY_cells,zero_count,one_count]=...
    createMap(SizeX,Xcells,SizeY,Ycells,Ncells,Delta,0,RRCS);
figure;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROGRAM %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Result=zeros(N_iterations_algorithm,6);
iterations_remained=N_iterations_algorithm

for iteration=1:N_iterations_algorithm
    %%%%%%%%% initial position %%%%%%%%%
    flag=1;
    track_of_mobile=zeros(3,N_movement_step);
    track_of_mobile(1,1)=px;
    track_of_mobile(2,1)=py;
    track_of_mobile(3,1)=po;

```

```

Bel_Totale=zeros(Ncells,N_movement_step);
Bel_Totale(:,1)=0.5;
Corr_cells_story=zeros(Ncells,N_movement_step);
D_mob_cell_i=zeros(Ncells,N_movement_step+1);
G_plot=zeros(Ncells,length(steer_direction));
theta_true=zeros(Ncells,N_movement_step+1);

for i=1:Ncells
    [flag_visible,theta_true(i,1),D_mob_cell_i(i,1)]=...
        get_visible_landmarks(Pos_Xcells(i),PosY_cells(i),MAXRANGE,...
            track_of_mobile(1,1),track_of_mobile(2,1),track_of_mobile(3,1));
end

MapGridPlot_re=zeros(Xcells+1,Ycells+1);
k=1;
j=1;
for i=1:Ncells
    MapGridPlot_re(k,j)=Bel_Totale(i,1);
    k=k+1;
    if k==Ycells+1
        j=j+1;
        k=1;
    end
end

steps_remained=N_movement_step
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% start program %%%%%%%%%%%%%%%
for t=2:N_movement_step+1
    %%% compute the position of the mobile for the t-th step %%%%%%%%%%%
    [track_of_mobile(1,t),track_of_mobile(2,t),track_of_mobile(3,t),flag]=
        mobile_Movement(track_of_mobile(1,t-1),track_of_mobile(2,t-1),...
            Delta_step,flag);
    Ebs=Pw*Tf*ones(Nobs,1);
    RSS_per_cell=zeros(Ncells,Nobs);
    RSS_per_cell_zero=zeros(Ncells,Nobs);
    s=zeros(1,Ncells);
    G=zeros(Ncells,Nobs);
    %%%%%%%%% compute Ebs %%%%%%%%%%%%%%%
    for i=1:Ncells
        [flag_visible,theta_true(i,t),D_mob_cell_i(i,t)]=...
            get_visible_landmarks(Pos_Xcells(i),PosY_cells(i),MAXRANGE,...
                track_of_mobile(1,t),track_of_mobile(2,t),track_of_mobile(3,t)
        if flag_visible==1

```

```

theta_true_rad=theta_true(i,t)*pi/180;
s(i)=floor(D_mob_cell_i(i,t)/dBin+1);
Corr_cells_story(i,t)=s(i);
for b=1:length(steer_direction)
    m=(b-1)*Nbins+s(i);
    G(i,m)=Dmax(1,iteration)*abs(ArrayFactor(theta_true_rad,...
        steer_direction(b),Nx(1,iteration),Ny(1,iteration),...
        k0,adx))^2/Narray(1,iteration)^2;
    G_plot(i,b)=G(i,m);
    if MapLinear(i)==RRCS
        Ebs(m)=Ebs(m)+L0*Tf*(RRCS^2)*G(i,m)^2/...
            (Dmax(1,iteration)*D_mob_cell_i(i,t)^4);
    else
        Ebs(m)=Ebs(m)+0;
    end
end
else
    for b=1:length(steer_direction)
        G_plot(i,b)=NaN;
    end
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% compute RSS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:Ncells
    if s(i)>0
        for b=1:length(steer_direction)
            m=(b-1)*Nbins+s(i);
            RRCS_rand=RRCS+(perc_var_RRCS(1,iteration)/...
                100*RRCS_noise(i))*RRCS;
            RSS_per_cell(i,m)=L0*Tf*(RRCS_rand^2)*G(i,m)^2/...
                (Dmax(1,iteration)*D_mob_cell_i(i,t)^4);
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% compute Rk and invRk %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Rk=zeros(Nobs,Nobs);
for m_index=1:Nobs
    Rk(m_index,m_index)=(N0*(Np*W*Ted*N0+2*Ebs(m_index)));
end
invRk=Rk\eye(Nobs);
if Narray(iteration)==16
    Ebs_real=Ebs+Vbs_4x4(:,t-1);
elseif Narray(iteration)==100
    Ebs_real=Ebs+Vbs_10x10(:,t-1);

```



```

elseif Narray(iteration)==225
    Ebs_real=Ebs+Vbs_15x15(:,t-1);
end
end
%%% cicle for compute the belief %%%%%%%%%%%
for i=1:Ncells

    %%%%%%%%% if the i-th cell is in range: %%%%%%%%%
    if s(i)>0
        [row,col,vol] = find(s==s(i));
        col_minus_i=col(col~=i);
        Ncellscomb=sum(vol)-1;
        Ncombos=2^Ncellscomb;
        Dcombos=0:Ncombos-1;
        combos_pwr=str2double(num2cell(dec2bin(Dcombos)));
        [index1,index2]=find(combos_pwr~=0);
        combos=combos_pwr;

        Inn_Matrix_zero=zeros(Ncombos,Nobs);
        Zbs_sum_zero=zeros(Nobs,Ncombos);
        Zbs_zero=zeros(Ncells,Ncombos);

        Inn_Matrix_one=zeros(Ncombos,Nobs);
        Zbs_sum_one=zeros(Nobs,Ncombos);
        Zbs_one=zeros(Ncells,Ncombos);

        p_not_mi=zeros(Ncombos,1);
        p_mi=zeros(Ncombos,1);
        for b=1:length(steer_direction)
            m=(b-1)*Nbins+s(i);

            %%%%%%%%% case of multiple neighbors %%%%%%%%%
            if Ncombos>1
                for i_row=1:length(index1)
                    combos_pwr(index1(i_row),index2(i_row))=...
                        RSS_per_cell(col_minus_i(index2(i_row)),m);
                end
                index_i=find(col==i);
                %%%%%%%%% compute Zbs_sum_zero %%%%%%%%%
                combos_pwr_zero=[zeros(size(combos_pwr,1),1) ...
                    combos_pwr];
                for j=1:index_i-1
                    combos_pwr_zero(:,[j,j+1])=...
                        combos_pwr_zero(:,[j+1,j]);
                end
            end
        end
    end
end

```

```

        combos_pwr_zero=combos_pwr_zero.';
        Zbs_zero(col,:)=combos_pwr_zero;
        Zbs_sum_zero(m,:)=sum(Zbs_zero,1)+Pw*Tf;
        %%%%% compute Zbs_sum_one %%%%%%%%%%%%%%
        combos_pwr_one=[RSS_per_cell(i,m)*...
            ones(size(combos_pwr,1),1) combos_pwr];
        for j=1:index_i-1
            combos_pwr_one(:,[j,j+1])=...
                combos_pwr_one(:,[j+1,j]);
        end
        combos_pwr_one=combos_pwr_one.';
        Zbs_one(col,:)=combos_pwr_one;
        Zbs_sum_one(m,:)=sum(Zbs_one,1)+Pw*Tf;

        %%%%% case of absence of neighbors %%%%%%%%%%%%%%
        else
            %%%%% compute Zbs_sum_zero %%%%%%%%%%%%%%
            Zbs_sum_zero(m)=Pw*Tf;
            %%%%% compute Zbs_sum_one %%%%%%%%%%%%%%
            Zbs_sum_one(m)=RSS_per_cell(i,m)+Pw*Tf;
        end
    end

    %%%%% compute p(mi) p(not_mi) %%%%%%%%%%%%%%
    Ebs_mat= repmat(Ebs_real,1,Ncombos);
    for j_combos=1:Ncombos
        for b=1:length(steer_direction)
            m=(b-1)*Nbins+s(i);
            Inn_Matrix_zero(j_combos,m)=...
                (Ebs_mat(m,j_combos)-Zbs_sum_zero(m,j_combos));
            Inn_Matrix_one(j_combos,m)=...
                (Ebs_mat(m,j_combos)-Zbs_sum_one(m,j_combos));
        end
    end

    for j_combos=1:Ncombos
        p_not_mi(j_combos,1)=exp(-Inn_Matrix_zero(j_combos,:))*...
            invRk*(Inn_Matrix_zero(j_combos,:).');
        p_mi(j_combos,1)=exp(-Inn_Matrix_one(j_combos,:))*...
            invRk*(Inn_Matrix_one(j_combos,:).');
    end

    Bel_mj_prec=ones(Ncombos,1);
    if Ncombos>1
        for j=1:Ncombos
            for k=1:Ncellscomb

```

```

        if combos(j,k)==0
            Bel_mj_prec(j,1)=Bel_mj_prec(j,1)*...
                (1-Bel_Totale(col_minus_i(k),t-1));
        elseif combos(j,k)==1
            Bel_mj_prec(j,1)=Bel_mj_prec(j,1)*...
                Bel_Totale(col_minus_i(k),t-1);
        end
    end
end
end

%%%%% compute belief of i-th cell for the (t-1)-th step %%
Bel_mi_prec=Bel_Totale(i,t-1);
%%%%% compute belief of i-th cell for the t-th step %%%
num=0;
for j=1:Ncombos
    num=num+p_not_mi(j)*(1-Bel_mi_prec)*Bel_mj_prec(j);
end
den=0;
for j=1:Ncombos
    den=den+p_mi(j)*Bel_mi_prec*Bel_mj_prec(j);
end
Bel_Totale(i,t)=(1+(num/den))^-1;
if isnan(Bel_Totale(i,t))
    Bel_Totale(i,t)=Bel_Totale(i,t-1);
end
else
%%%%% if the i-th cell isn't in range: %%%
Bel_Totale(i,t)=Bel_Totale(i,t-1);
end
end

%%%%% plot irradiation diagram + map on-line %%%
for b=1:length(steer_direction)
    GainGridPlot_re=zeros(Xcells+1,Ycells+1);
    k=1;
    j=1;
    for i=1:Ncells
        GainGridPlot_re(k,j)=G_plot(i,b);
        k=k+1;
        if k==Ycells+1
            j=j+1;
            k=1;
        end
    end
end

```

```

end
clf
subplot(1,2,1); pcolor(MapGridPlot_re);...
    axis([1 (Xcells+1) 1 (Ycells+1)]);hold on
subplot(1,2,1); plot((track_of_mobile(1,+)/Delta)+1,...
    (track_of_mobile(2,+)/Delta)+1,'og','MarkerFaceColor','g');...
    hold on
subplot(1,2,2); pcolor(GainGridPlot_re);...
    axis([1 (Xcells+1) 1 (Ycells+1)]); shading interp;hold on
drawnow
end
MapGridPlot_re=zeros(Xcells+1,Ycells+1);
k=1;
j=1;
for i=1:Ncells
    MapGridPlot_re(k,j)=Bel_Totale(i,t);
    k=k+1;
    if k==Ycells+1
        j=j+1;
        k=1;
    end
end
subplot(1,2,1); pcolor(MapGridPlot_re);...
    axis([1 (Xcells+1) 1 (Ycells+1)]);hold on
subplot(1,2,1); plot((track_of_mobile(1,+)/Delta)+1,...
    (track_of_mobile(2,+)/Delta)+1,'og','MarkerFaceColor','g');...
    hold on
drawnow

steps_remained=steps_remained-1
end
Belief{1,iteration}=Bel_Totale;
iterations_remained=iterations_remained-1
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% pulizia workspace %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clearvars -except Belief N_iterations_algorithm N_movement_step Ncells...
    MapLinear one_count zero_count

```

# Bibliografia

- [1] L. Liao D. Shchulz G. Borriello D. Fox, J. Hightower. Bayes filtering for location estimation. pages 1–6, 2003.
- [2] Davide Dardari Francesco Guidi, Anna Guerra. Personal mobile radars with millimeter-wave massive arrays for indoor mapping. pages 5–7, 2016.
- [3] Tim Bailey Hugh Durrant-Whyte. Simultaneous localization and mapping: Part i. pages 2–6, 2006.
- [4] S. Clark H.F. Durrant-Whyte M. Csorba M.W.M. Gamini Dissanayake, P. Newman. A solution to the simultaneous localization and map building (slam) problem. pages 1–4, 2001.
- [5] Simo Sarkka. *Bayesian Filtering and Smoothing*. 2013.
- [6] Peter FM Smulders and LM Correia. Characterisation of propagation in 60 ghz radio channels. *Electronics & communication engineering journal*, 9(2):73–80, 1997.
- [7] Sebastian Thrun. Robotic mapping: A survey. pages 19–20, 2002.