



Projecte Fi de Carrera

**Enginyeria de Telecomunicació**

---

**Diseño, Implementación y Validación de un  
Sistema de Localización Híbrido basado en una red  
de sensores inalámbricos (WSN).**

Daniel Egea Roca

---

Director: José López Vicario

*Departament de Telecomunicació i Enginyeria de Sistemes*

**Escola d'Enginyeria  
Universitat Autònoma de Barcelona (UAB)**

Setembre 2012





El sotasignant, José López Vicario, Professor de l'Escola d'Enginyeria de la Universitat Autònoma de Barcelona (UAB),

CERTIFICA:

Que el projecte presentat en aquesta memòria de Projecte Fi de Carrera ha estat realitzat sota la seva direcció per l'alumne *Daniel Egea Roca*.

I, perquè consti a tots els efectes, signa el present certificat.

Bellaterra, *5 de Septiembre del 2012*.

Signatura: *José López Vicario*

*En primer lugar, doy las gracias a mis padres y hermanos por ser como son y brindarme el apoyo y confianza que han permitido superar los momentos más difíciles en el desarrollo del proyecto, que no han sido pocos. También quiero agradecer especialmente a todos mis amigos que han estado siempre a mi lado dando ánimos y apoyo para hacer más llevadero el trabajo, sin los que habría sido imposible llegar hasta este punto. Quiero demostrar mi máxima gratitud a todas esas personas que han estado a mi lado cuando más lo he necesitado así como todas las personas que me han acompañado en el transcurso de la carrera por esas largas horas de estudio y trabajo y por todo lo aprendido de ellos en este tiempo.*

*En segundo lugar, mi más sincera gratitud al Dr. José López Vicario por confiar en mí en todo momento, dedicarme su tiempo, compartir conmigo su conocimiento y sobretodo por sus consejos y apoyo en los momentos de desesperación.*

*Por último, agradecer la colaboración de Albert Bel y Alejandro Correa en la resolución de todas y cada una de las dudas en la programación de las notas.*

“La formulación de un problema es más importante que su solución.”

**-Albert Einstein**

# ÍNDICE

ÍNDICE DE FIGURAS.....	9
ÍNDICE DE TABLAS .....	11
ABREBIACIONES.....	13
1    Introducción.....	16
1.1    Objetivos.....	18
1.2    Estructura de la memoria.....	19
2    Fundamentos teóricos.....	22
2.1    Redes de sensores inalámbricos (WSN).....	22
2.1.1    Arquitectura de motas y Hardware utilizado.....	23
2.1.2    Sistemas operativos y lenguajes de programación para WSN.....	26
2.1.3    Aplicaciones de las WSN.....	29
2.2    Conceptos de localización y posicionamiento.....	31
2.2.1    GPS (Global Positioning System).....	33
2.2.2    Sistemas de localización basados en medidas de RSS.....	35
2.2.3    Cálculo de la posición o localización.....	39
3    Técnicas de localización híbridas.....	44
3.1    Ponderación.....	48
3.1.1    Ponderación modificada.....	49
3.2    BLUHE, <i>Best Linear Unbiased Hybrid Estimator</i> .....	51
3.2.1    Modelo de señal utilizado.....	51
3.2.2    Determinación del BLUE híbrido (BLUHE).....	53
4    Aspectos de implementación del sistema.....	58
4.1    Programación de las motas de la WSN.....	60
4.1.1    Motas anclas.....	61
4.1.2    Motas desconocidas.....	62
4.2    Programación del <i>gateway</i> .....	64
4.3    Programación en MATLAB.....	67
4.3.1    Función GPS.m.....	68
4.3.2    Función triangulacion.m.....	68
4.3.3    Archivo principal (Main).....	69
4.3.4    Fase de entrenamiento.....	71

5	Resultados Híbridos.....	76
5.1	Parámetros de propagación y escenarios de medida. ....	78
5.1.1	Parámetros en línea recta.....	78
5.1.2	Parámetros en escenarios de localización. ....	82
5.1.3	Tabla de parámetros de propagación.....	87
5.2	Resultados del posicionamiento. ....	88
5.2.1	Escenarios exteriores.....	89
5.2.2	Escenarios interiores. ....	98
5.3	Conclusiones de los resultados híbridos.....	99
6	Estimación del error instantáneo.....	104
6.1	Error del posicionamiento GPS.....	105
6.2	Error del posicionamiento RSS. ....	108
6.3	Resultados experimentales. ....	112
6.3.1	Estimaciones del error instantáneo RSS. ....	113
6.3.2	Resultados híbridos con estimaciones de error. ....	118
6.3.3	Conclusiones de resultados experimentales. ....	122
7	Conclusiones.....	126
8	Bibliografía. ....	134
A	ANEXO 1: Programación en NesC. ....	140
A.1	Programación de las motas anclas.....	140
A.1.1	Configuración.....	140
A.1.2	Módulo. ....	141
A.1.3	Estructura del mensaje. ....	143
A.2	Programación de motas a localizar.....	144
A.2.1	Componente. ....	144
A.2.2	Módulo. ....	146
A.2.3	Estructura del mensaje. ....	168
B	ANEXO 2: Programación en MATLAB. ....	172
B.1	GPS.m.....	172
B.2	triangulación.m.....	175
B.3	filtroMsg.m.....	179
B.4	hibrido.m.....	181
B.5	parametros.m.....	187

B.6	resultados.....	189
B.7	Archivo Main.....	191
B.8	Pruebas.m .....	196
C	ANEXO 3: Medidas adicionales.....	200
C.1	Parámetros de propagación y posicionamiento. ....	200
C.1.1	Escenario Interior. ....	201
C.1.2	Escenario de localización interior. ....	202
C.1.3	Escenario de localización interior 2 (EI2).....	203
C.1.4	Escenario de localización exterior.....	204
C.2	Estimación de error RSS y localización híbrida.....	206
C.2.1	Escenario de localización 3 (E3).....	206
C.2.2	Localización híbrida con error GPS instantáneo.....	207



# ÍNDICE DE FIGURAS

<b>Figura 1.</b>	Arquitectura básica de una WSN [3].....	23
<b>Figura 2.</b>	Arquitectura básica de las motas utilizadas en las WSN [5].....	24
<b>Figura 3.</b>	Mota Iris de Crossbow. ....	24
<b>Figura 4.</b>	Fotografía de la <i>sensor-board</i> utilizada (MTS420/400CC) [8]. ....	25
<b>Figura 5.</b>	<i>Gateway</i> utilizado (MIB520). ....	25
<b>Figura 6.</b>	Ejemplo código NesC.....	28
<b>Figura 7.</b>	Monitorización del hábitat en <i>Great Duck</i> . ....	29
<b>Figura 8.</b>	Control de transporte y logística de mercancías [20].....	30
<b>Figura 9.</b>	Localización utilizando círculos [21'].....	32
<b>Figura 10.</b>	Constelación de satélites GPS [24]. ....	33
<b>Figura 11.</b>	Sistema de coordenadas ECEF.....	34
<b>Figura 12.</b>	Efecto del PL, <i>shadowing</i> y <i>multipath</i> [26].....	36
<b>Figura 13.</b>	Resultados experimentales, parametrización y regresión lineal.....	39
<b>Figura 14.</b>	Escenario de localización [32]. ....	39
<b>Figura 15.</b>	Efecto en localización de errores en la medida de distancia [32']... ..	41
<b>Figura 16.</b>	Diagrama de flujo del sistema de posicionamiento híbrido. ....	46
<b>Figura 17.</b>	Relaciones entre ME_GPS y el parámetro $\mu$ .....	48
<b>Figura 18.</b>	Relaciones entre $\beta$ y $\mu$ . ....	50
<b>Figura 19.</b>	Escenario híbrido.....	52
<b>Figura 20.</b>	Significado geométrico del error cometido en el posicionamiento... ..	53
<b>Figura 21.</b>	Diagrama del sistema híbrido implementado.....	59
<b>Figura 22.</b>	Estructura de empaquetamiento de mensaje TinyOS.....	61
<b>Figura 23.</b>	Formato de los mensajes GGA. ....	63
<b>Figura 24.</b>	Información útil (mensaje GGA) de las motas desconocidas. ....	63
<b>Figura 25.</b>	Esquema de la aplicación programada en las motas desconocidas... ..	64
<b>Figura 26.</b>	Captura de la ventana de XSniffer. ....	65
<b>Figura 27.</b>	Captura de mensajes de la WSN utilizada. ....	66
<b>Figura 28.</b>	Correspondencia sistema coordenadas local y UTM. ....	71
<b>Figura 29.</b>	Procedimiento para tomar las medidas.....	77
<b>Figura 30.</b>	Medidas en escenario interior y parámetros resultantes.....	79
<b>Figura 31.</b>	Medidas en escenario exterior y parámetros resultantes.....	80
<b>Figura 32.</b>	Efecto suelo que produce la absorción de la señal [39]. ....	81
<b>Figura 33.</b>	Escenario de localización 1. ....	83
<b>Figura 34.</b>	Medidas en escenario 1 y parámetros resultantes. ....	83
<b>Figura 35.</b>	Escenario de localización 2. ....	84
<b>Figura 36.</b>	Medidas en escenario 2 y parámetros resultantes. ....	85
<b>Figura 37.</b>	Escenario de localización 3. ....	86
<b>Figura 38.</b>	Medidas en escenario 3 y parámetros resultantes. ....	86
<b>Figura 39.</b>	Expresiones utilizadas. ....	89
<b>Figura 40.</b>	Estimaciones RSS y GPS en escenario de localización 1. ....	90

<b>Figura 41.</b>	Estimaciones híbridas con parámetros promedios e instantáneos.....	92
<b>Figura 42.</b>	Posicionamiento dentro del rango del error GPS. ....	101
<b>Figura 43.</b>	Ejemplo de UERE extraído de [42].....	106
<b>Figura 44.</b>	GDOP: (a) geometría con bajo DOP y (b) con alto DOP [43].....	107
<b>Figura 45.</b>	Significado geométrico de las distancias $d_{pi}$ . ....	110
<b>Figura 46.</b>	Factor $r$ necesario y parámetros obtenidos para calcularlo. ....	117
<b>Figura 47.</b>	Medidas en escenario interior obteniendo Parámetros 7.....	201
<b>Figura 48.</b>	Escenario de localización interior. ....	202
<b>Figura 49.</b>	Medidas en escenario interior y Parámetros 8. ....	202
<b>Figura 50.</b>	Escenario de localización EI2. ....	203
<b>Figura 51.</b>	Escenario de localización exterior.....	204
<b>Figura 52.</b>	Medidas en escenario de localización exterior.....	205

## ÍNDICE DE TABLAS

<b>Tabla 1.</b>	Resultados experimentales de ejemplo. ....	38
<b>Tabla 2.</b>	Medidas de potencia en escenario interior. ....	79
<b>Tabla 3.</b>	Medidas de potencia en escenario exterior. ....	80
<b>Tabla 4.</b>	Coordenadas del escenario de localización 1. ....	82
<b>Tabla 5.</b>	Coordenadas de motas anclas en el escenario 2. ....	84
<b>Tabla 6.</b>	Posiciones de medida del escenario 2. ....	84
<b>Tabla 7.</b>	Coordenadas de motas anclas en el escenario 3. ....	85
<b>Tabla 8.</b>	Posiciones de medida del escenario 3. ....	85
<b>Tabla 9.</b>	Tabla de parámetros de propagación utilizados en el PFC. ....	87
<b>Tabla 10.</b>	Errores medios de posicionamiento según los parámetros (E1). ....	89
<b>Tabla 11.</b>	Coordenadas GPS. ....	91
<b>Tabla 12.</b>	Coordenadas RSS. ....	91
<b>Tabla 13.</b>	Error de posicionamiento para cada técnica de localización (E1). ....	92
<b>Tabla 14.</b>	Errores medios de posicionamiento según los parámetros (E2). ....	94
<b>Tabla 15.</b>	Coordenadas GPS y RSS en escenario de localización 2. ....	94
<b>Tabla 16.</b>	Error de posicionamiento para cada técnica de localización (E2). ....	95
<b>Tabla 17.</b>	Errores medios de posicionamiento según los parámetros (E3). ....	96
<b>Tabla 18.</b>	Coordenadas GPS y RSS en escenario de localización 3. ....	97
<b>Tabla 19.</b>	Error de posicionamiento para cada técnica de localización (E3). ....	97
<b>Tabla 20.</b>	Distancias y motas anclas del posicionamiento en E1. ....	113
<b>Tabla 21.</b>	Factor $F$ obtenido en E1. ....	113
<b>Tabla 22.</b>	Factor $r$ necesario y obtenido en las posiciones de E1. ....	114
<b>Tabla 23.</b>	Error RSS estimado en E1. ....	115
<b>Tabla 24.</b>	Distancias y motas anclas del posicionamiento en E2. ....	116
<b>Tabla 25.</b>	Factor $F$ obtenido en E2. ....	117
<b>Tabla 26.</b>	Factor $r$ necesario y obtenido en las posiciones de E2. ....	117
<b>Tabla 27.</b>	Error RSS estimado en E2. ....	118
<b>Tabla 28.</b>	Error de posicionamiento con los errores estimados (E1). ....	119
<b>Tabla 29.</b>	Errores para las técnicas promedio e instantáneas teóricas (E1). ....	120
<b>Tabla 30.</b>	Error de posicionamiento con los errores estimados (E2). ....	121
<b>Tabla 31.</b>	Errores para las técnicas promedio e instantáneas teóricas (E2). ....	121
<b>Tabla 32.</b>	Medidas de potencia en escenario interior. ....	201
<b>Tabla 33.</b>	Coordenadas del escenario de localización interior. ....	202
<b>Tabla 34.</b>	Coordenadas RSS estimadas. ....	203
<b>Tabla 35.</b>	Posiciones de medida del escenario de localización EI2. ....	203
<b>Tabla 36.</b>	Coordenadas de motas anclas en EI2. ....	203
<b>Tabla 37.</b>	Coordenadas RSS estimadas. ....	204
<b>Tabla 38.</b>	Coordenadas del escenario de localización exterior. ....	204
<b>Tabla 39.</b>	Posicionamiento RSS en escenario de localización exterior. ....	205
<b>Tabla 40.</b>	Posicionamiento GPS en escenario de localización exterior. ....	205

<b>Tabla 41.</b>	Error de posicionamiento para cada técnica de localización. ....	206
<b>Tabla 42.</b>	Error RSS estimado en E3. ....	206
<b>Tabla 43.</b>	Error de posicionamiento con los errores estimados (E3). ....	207
<b>Tabla 44.</b>	Errores promedios utilizando error GPS instantáneo.....	207
<b>Tabla 45.</b>	Comparación de errores promedios según las medidas utilizadas.....	208

## ABREVIACIONES

$\alpha$ : Vector columna  $\alpha$ .

$A$ : Matriz  $A$ .

$\alpha^T/A^T$ : Vector/Matriz transpuesto.

$\alpha^H/A^H$ : Vector/Matriz hermítico (transpuesto conjugado).

AWGN: *Additive White Gaussian Noise*/Ruido Blanco Gaussiano.

BLUE: *Best Linear Unbiased Estimator*/Mejor estimador lineal insesgado.

BLUHE: *Best Linear Unbiased Hybrid Estimator*/BLUE híbrido.

BS: *Base Station*/ Estación Base.

DARPA: *Defense Advanced Research Projects Agency*.

dB: Decibelios.

DOA: *Direction Of Arrival*.

DSN: *Distributed Sensor Network*.

ECEF: *Earth Centered-Earth Fixed*.

GDOP: *Geometric Dilution Of Precision*.

GGA: *Global Positioning System Fix Data*.

GPS: *Global Positioning System*.

LAN: *Local Area Network*/Red de área local.

LED: *Light-Emitting Diode*.

LS: *Least Squares*.

ME: *Mean Error*/ Error Medio.

MIB: *Mote Interface Board*/Placa interfaz de motas.

MS: *Mobile Station*/Estación móvil.

NesC: *Network Embedded Systems C*.

Pag.: Página.

PFC: Proyecto Final de Carrera.

PG: *Path Gain*/ Ganancia de canal.

PL: *Path Loss*/ Pérdidas de canal.

RAM: *Random Access Memory*.

RSS: *Receive Signal Strength*/ Potencia Recibida.

TDOA: *Time Difference Of Arrival*.

TOA: *Time Of Arrival*.

TOF: *Time Of Flight*.

UERE: *User-Equivalent Range Error*.

WGS-84: *World Geodetic System 84*.

WSN: *Wireless Sensor Network*/Red de sensores inalámbricos.

.csv: Archivo de valores separados por coma.

[et al.]: “y otros”.

[x]: Referencia bibliográfica x, siendo x un número.

[x’]: Referencia bibliográfica x’ relacionada de alguna manera con la referencia [x].

(n): Referencia a la ecuación n, siendo n un número.



# 1 Introducción.

En la actualidad se encuentran muchos sistemas y dispositivos electrónicos que utilizan sensores, sin embargo la mayor parte de estos sensores no disponen de capacidad de procesar y analizar los datos que detectan, limitándose básicamente a medir variables físicas y enviarlas a un procesador central. El desarrollo de los transceptores en los últimos años ha permitido la llegada de una nueva generación de sensores inteligentes, capaces de organizarse e interconectarse de forma inalámbrica entre ellos. Es así como surgen las denominadas redes de sensores inalámbricos (*Wireless Sensor Networks*, WSN), que consisten en redes formadas por un gran número de estos sensores inteligentes (motas) que intercambian información entre sí sin necesidad de cables.

A partir de la aparición de estos nuevos sensores, varios laboratorios de investigación tanto en el sector científico como en el industrial han mostrado un gran interés por este tipo de redes debido a que pueden ser desplegadas en entornos de difícil acceso y sin infraestructura de red previa. En este tipo de redes existe un amplio rango de aplicaciones entre ellas la automoción, aplicaciones militares y médicas, predicción medioambiental, entre otras como el posicionamiento tanto en exterior como interior de uno de los elementos de la WSN.

Además del gran interés científico e industrial y el gran abanico de aplicaciones que presentan las WSN, en diversos informes se augura que este tipo de redes conllevará a una revolución tecnológica similar a la que tuvo la aparición de Internet, también se habla de redes de vigilancia global del planeta con capacidad de registrar información de personas, realizar seguimientos de individuos y mercancías concretas, monitorizar el tráfico, tiempo, etc. Para todo esto habrá que esperar todavía unos años, no obstante sí que han surgido múltiples iniciativas y proyectos de investigación de enorme interés y aplicabilidad práctica como es el caso del proyecto DSN (*Distributed Sensor Network*) llevado a cabo por la institución dependiente del Departamento de Defensa Estadounidense DARPA (*Defense Advanced Research Projects Agency*) [1],[2].

Se ha comentado la posibilidad de utilizar las WSN para la localización tanto en entornos exteriores como interiores. El problema de la localización en exteriores ya hace tiempo que está resuelta con los sistemas de localización como el GPS (*Global Positioning System*), sin embargo el problema de la localización en interiores ha sido objeto de estudio e investigación durante los últimos años ya que hasta ahora ninguna de las soluciones propuestas ha conseguido el éxito obtenido por los sistemas utilizados en exteriores. En estos años de estudio e investigación del posicionamiento en interiores han surgido varias soluciones utilizando WSN proporcionando resultados similares o incluso mejores que los resultados de GPS en exteriores, estas técnicas se basan en la estimación de distancias a partir de la RSS (*Receive Signal Strength*) y posterior estimación de la posición a partir de estas distancias.



La capacidad de posicionamiento en interiores que brindan las WSN permite la creación de un sistema de localización completo o híbrido que sea capaz de proporcionar un servicio aceptable tanto en entornos interiores como entornos exteriores. Estos sistemas híbridos permiten el desarrollo de innumerables aplicaciones relacionadas con el posicionamiento de objetos o personas en tiempo real en entornos mixtos, donde en cualquier instante de tiempo el objeto o persona a posicionar puede estar en un entorno exterior o interior. Alguno de los principales servicios están relacionados con:

- El control de accesos a complejos deportivos, industriales, de ocio, etc., a partir de la identificación de los usuarios.
- La gestión de las instalaciones, permitiendo el ahorro energético y el guiado de los usuarios por las instalaciones.
- Servicios de emergencia o alertas y servicios de estadísticas.

No obstante, las aplicaciones de mayor interés comercial en los últimos años provienen del posicionamiento contextual, que permite el desarrollo de aplicaciones que reaccionan ante los cambios de contexto de los usuarios, lo que incluye su posición y los recursos cercanos. De esta manera es posible crear aplicaciones comerciales como:

- La publicidad a la carta según la posición del usuario en un centro comercial.
- El guiado en aeropuertos, museos o parques temáticos proporcionando información sobre los establecimientos o puntos de interés cercanos.
- Ayuda a la movilidad de gente discapacitada en este tipo de establecimientos.

El objetivo principal de estas aplicaciones es el de proporcionar información al usuario relacionada con los eventos/recursos disponibles alrededor suyo. La información proporcionada puede estar relacionada con:

- Productos accesibles desde la posición actual, así como la indicación de productos relacionados o complementarios.
- Eventos cercanos a la posición del usuario, así como información relacionada como el tiempo de espera para acceder, horario del evento, etc.
- Recorrido a seguir para llegar a un cierto destino, así como recorridos alternativos que pasan por lugares de interés del usuario.

En el presente proyecto se plantea un sistema de posicionamiento híbrido en el sentido de proporcionar posicionamiento tanto en entornos exteriores como en entornos interiores y en el sentido de combinar dos técnicas de posicionamiento para obtener una técnica de posicionamiento conjunta que proporcione unas mejores prestaciones. En el caso del PFC que se describe en esta memoria se ha realizado un sistema híbrido a partir de una WSN que combina técnicas de posicionamiento GPS y RSS, aun así la solución híbrida que se propone puede ser utilizada para la combinación de cualquier otro tipo de técnicas y sin la necesidad de una WSN.

## 1.1 Objetivos.

Este PFC tiene como objetivo el diseño, implementación y validación de un sistema de posicionamiento híbrido basado en una red de sensores inalámbricos (WSN) que permita la combinación de los resultados GPS con el posicionamiento a partir de RSS para conseguir un posicionamiento aceptable tanto en lugares exteriores como interiores. Para llegar a tal objetivo se tiene que:

- Obtener medidas RSS en una WSN para la posterior estimación de distancias que nos permitan estimar la posición deseada.
- Obtener medidas GPS en una WSN para la posterior transformación y translación de coordenadas que nos permita situar las coordenadas GPS en un sistema de coordenadas común al del punto anterior.
- Proporcionar una solución de posicionamiento híbrido que permita:
  - Posicionamiento exterior e interior con resultados aceptables.
  - Siempre que se pueda, la combinación de medidas GPS y RSS para la obtención de resultados más precisos.
- Estudio de errores en la estimación de distancia y su posible propagación en los errores de las posiciones estimadas con el fin de proponer una técnica de estimación del error de posicionamiento instantáneo (en cada una de las posiciones) para poder ser usadas en las técnicas de localización híbridas o en otros sistemas de posicionamiento presentes en la actualidad y poder saber si las posiciones obtenidas son fiables o no según su precisión o error.
- Implementación del *software* en MATLAB que permita la validación del sistema de posicionamiento híbrido.
- Pruebas del sistema de posicionamiento en distintos escenarios a partir de medidas experimentales que son procesadas con el *software* previamente generado.

## 1.2 Estructura de la memoria.

La presente memoria describe los conceptos necesarios para entender el desarrollo del proyecto así como el proceso llevado a cabo en el PFC para la obtención de los objetivos mencionados. La estructura de la memoria está estructurada en siete capítulos, tal y como se indica a continuación:

- En el **Capítulo 1** se han presentado las motivaciones principales de este proyecto, que son las redes de sensores inalámbricos y el posicionamiento/localización de personas u objetos, además se han definido los objetivos a conseguir con la realización del PFC así como la organización de la memoria del mismo.
- En el **Capítulo 2** se introducen los conceptos teóricos necesarios para el diseño del posicionamiento híbrido, entre los que se encuentran:
  - Conceptos relacionados con las WSN.
  - Conceptos de posicionamiento general que conducen a las técnicas de localización utilizadas y estudiadas en el proyecto:
    - **GPS.**
    - **RSS.**
- En el **Capítulo 3** se proponen técnicas de posicionamiento híbridas que permitan la combinación de varias técnicas de localización y brinden la posibilidad de obtener un sistema de localización aceptable tanto en entornos interiores como exteriores.
- En el **Capítulo 4** se comentan los aspectos de implementación a tener en cuenta, relacionados con:
  - La programación de las motas para la obtención de medidas RSS y GPS.
  - La programación de las aplicaciones que proporcionen el posicionamiento RSS y relacionen su sistema de coordenadas con el de GPS para poder utilizar las técnicas híbridas.
  - Programas de apoyo utilizados para la adquisición de los datos de la WSN para su posterior procesado.
  - La implantación del sistema según la aplicación que se quiera obtener.
- En el **Capítulo 5** se muestran, comparan y comentan los resultados experimentales del posicionamiento tanto GPS, RSS, como híbrido, que permiten la validación del sistema diseñado e implementado en esta memoria.

- En el **Capítulo 6** se realiza un estudio teórico de los errores de posicionamiento de las técnicas utilizadas a lo largo del PFC, con el objetivo de proponer técnicas de estimación de errores en la posición estimada para ser usados en las estimaciones híbridas y conseguir así una mejora de prestaciones. Una vez propuesta alguna técnica de estimación de error de posicionamiento se utiliza en las técnicas híbridas y se comprueban los resultados obtenidos.
- Por último, en el **Capítulo 7** se reflexiona sobre el trabajo realizado y los resultados obtenidos para extraer unas conclusiones y líneas futuras del PFC.

Para completar el contenido de los siete capítulos se incluyen cuatro **ANEXOS** en los que se adjunta información adicional relacionada con alguno de los capítulos que componen la memoria, estos **ANEXOS** son los siguientes:

- **ANEXO 1:** Incluye los códigos en NesC utilizados para configurar las motas de la red de sensores de tal manera que se obtengan las capacidades deseadas para el funcionamiento del sistema de localización híbrido.
- **ANEXO 2:** Recoge los programas creados en lenguaje MATLAB para procesar los datos obtenidos en la WSN y poder obtener de ellos el posicionamiento híbrido deseado a partir del cual será posible validar el sistema implementado.
- **ANEXO 3:** Incluye las medidas experimentales llevadas a cabo a lo largo del proyecto que no se muestran en los capítulos de la memoria, así como medidas experimentales relacionadas con las estimaciones de errores de posicionamiento y su utilización en las técnicas híbridas que se han realizado en el transcurso del PFC y no se incluyen en la memoria.



## 2 Fundamentos teóricos.

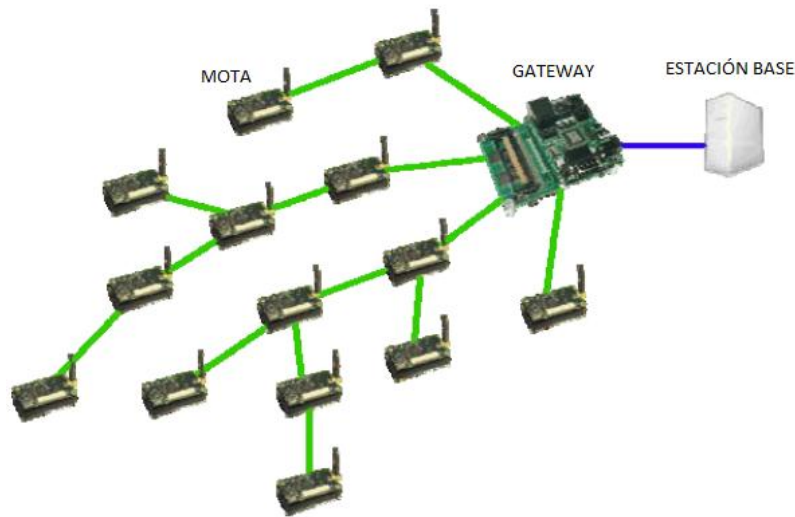
En este capítulo se tratan los conceptos fundamentales para poder llevar a cabo el diseño, la implementación y evaluación del sistema de posicionamiento híbrido a obtener con el desarrollo del presente PFC. El capítulo se divide en dos temas principales, el primero relacionado con las redes de sensores y el segundo con los conceptos relacionados con el posicionamiento y localización.

Para empezar, se define con detalle las redes de sensores inalámbricos y su arquitectura básica así como la arquitectura de los nodos constituyentes de la red, los sistemas operativos y lenguajes de programación disponibles para ponerlos en marcha y principales aplicaciones. Además de la información general sobre las WSN se detalla el *hardware* concreto necesario para formar la red de sensores utilizada en el proyecto que llevará a cabo el sistema de posicionamiento híbrido.

A continuación, se presenta el problema general de la estimación de la posición o localización así como su solución a partir de la estimación de distancias entre nodos de la WSN (en el caso del PFC), comentando las posibles técnicas utilizadas para estas estimaciones. Una vez presentado el problema, se analiza por un lado el sistema de posicionamiento global GPS y por otro se estudia con detalle los efectos físicos influyentes en la potencia recibida, conduciendo a la parametrización del escenario que permitirá la modelización de la potencia recibida o RSS a partir de la cual se estimarán las distancias que permiten calcular la posición de nodos desconocidos. Esta parametrización es la idea utilizada en el PFC con el objetivo de obtener unos resultados precisos en el posicionamiento, tanto en escenarios interiores como exteriores. Para acabar, se analiza la manera de obtener la posición de un terminal a partir de las distancias estimadas.

### 2.1 Redes de sensores inalámbricos (WSN).

Los avances en las comunicaciones inalámbricas, en la micro-fabricación e integración de circuitos, y de los microprocesadores empotrados (*embedded microprocessors*) han permitido una nueva generación de redes de sensores de larga escala disponible para aplicaciones comerciales, científicas y militares. Esta nueva generación de redes de sensores ha aparecido gracias al desarrollo de dispositivos sensores de tamaño reducido, bajo coste y consumo mínimo capaces de procesar información de manera local y de comunicarse inalámbricamente, trabajando de forma cooperativa. Una red de sensores inalámbricos (WSN) consiste en muchos de estos nuevos dispositivos sensores, denominados nodos o motas, distribuidos geográficamente en un lugar concreto para cumplir una función determinada de forma cooperativa.



**Figura 1.** Arquitectura básica de una WSN [3].

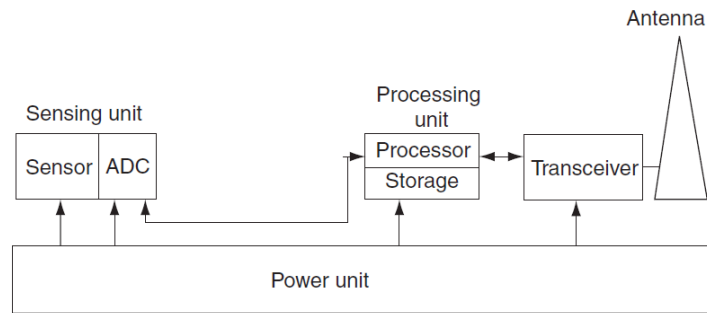
Normalmente, cada nodo está equipado con uno o más dispositivos transductores (temperatura, humedad, presión, aceleración,...) y se comunican entre ellos de forma inalámbrica dentro de su rango de cobertura. Además, en una WSN, los nodos pueden ser los encargados de encaminar los datos hacia la estación base que se encuentra conectada a un ordenador que puede comunicarse hacia el exterior a través de Internet o una red de área local (LAN). De esta manera se puede representar la arquitectura básica de una WSN como el esquema mostrado en la **Figura 1**, en la que se distinguen tres elementos principales:

- **Mota:** Son los sensores inalámbricos o nodos que captan la información deseada y la procesan para transmitirla inalámbricamente hacia el destinatario.
- **Gateway:** Elemento encargado de interconectar la red de sensores con otros dispositivos o redes de datos que nos permitan monitorizar, adquirir y procesar la información que proviene de la WSN.
- **Estación base:** Agente encargado de recolectar los datos que llegan al *gateway* y permitir al usuario monitorizar, adquirir o procesar la información.

### **2.1.1** *Arquitectura de motas y Hardware utilizado.*

Actualmente, existe una gran variedad de motas, tal y como se puede comprobar en la página oficial del fabricante Crossbow [4], cada una con sus particularidades pero todas se basan en el esquema básico representado en la **Figura 2**, donde se aprecian 4 componentes importantes:

1. **Fuente de energía:** Elemento encargado de suministrar la energía apropiada para alimentar las motas durante horas, meses o años, dependiendo de la aplicación. Normalmente la fuente de energía proviene de un sistema de alimentación autónomo, es decir, baterías.
2. **Microcontrolador:** Proporciona la funcionalidad de cómputo y almacenamiento necesarios para tratar la información que proviene de los sensores y la posterior transmisión por la red.



**Figura 2.** Arquitectura básica de las motas utilizadas en las WSN [5].

3. **Unidad de sensores:** Elemento encargado de medir alguna magnitud física y convertirla en una señal conveniente para ser procesada por el microcontrolador.
4. **Transceptor:** Este elemento permite la transmisión y recepción de datos, conectando el nodo con la red de sensores.

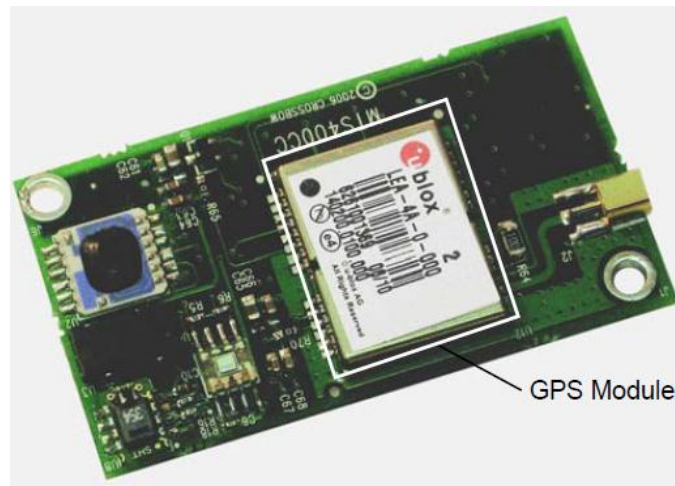
Esta arquitectura básica puede ser ampliada según el modelo de mota con varios elementos como pueden ser generadores de energía (placas solares) o sistemas de localización o posicionamiento. En el caso particular del presente PFC se ha utilizado el modelo de mota Iris (2.4 GHz) de Crossbow cuyo *datasheet* puede ser consultado en [6] y cuya apariencia física se presenta en la **Figura 3**.

La mota de la **Figura 3** dispone de todos los elementos principales del esquema en la **Figura 2** excepto la unidad de sensores. Es habitual que los fabricantes proporcionen por separado distintas unidades de sensores de tal manera que el usuario pueda obtener la unidad de sensor que más le convenga según su aplicación. Si se observa la **Figura 3** se distingue un conector de expansión para conectar estas unidades de sensores, denominadas en la industria como *sensor-boards*. Concretamente la *sensor-board* utilizada ha sido la MTS420/400CC ya que dispone de un receptor GPS (uBlox LEA-4A) y la conexión de su antena (**Figura 4**). Para más información sobre la *sensor-board* utilizada y el receptor GPS disponible consultar [7,8] y [9] respectivamente.



**Figura 3.** Mota Iris de Crossbow.





**Figura 4.** Fotografía de la *sensor-board* utilizada (MTS420/400CC) [8].

Se ha visto que dentro de la arquitectura de una red de sensores inalámbricos además de las motas, que se encargan de captar la información deseada y propagarla a través de la red, es necesario otro elemento que permita la interconexión de la WSN con otros dispositivos capaces de representar, almacenar o procesar la información adquirida. Este elemento es el *gateway* y en el caso del PFC se ha utilizado una de las placas de interfaz de motas o MIB (*Mote Interface Board*) suministradas también por Crossbow, concretamente la MIB520 (**Figura 5**). Para poder utilizar esta MIB como puente entre las motas que forman la WSN y la estación base se necesita tener conectada una mota como las de la **Figura 3**, correctamente programada, y estar conectada por su conexión USB a un ordenador, que será el dispositivo utilizado como estación base. Además, esta MIB es la placa de programación que permite instalar las aplicaciones programadas en el ordenador en las motas utilizadas para la WSN.

Para completar la arquitectura básica de una WSN falta por definir la estación base utilizada, que ya se ha comentado en el párrafo anterior que se utiliza un ordenador para poder visualizar, almacenar y procesar los datos obtenidos de la WSN. En concreto Crossbow proporciona un *software* (XSniffer) que permite visualizar los datos recibidos por la mota conectada en el *gateway* (MIB520) y crear unos ficheros que pueden ser procesados por el *software* MATLAB.



**Figura 5.** *Gateway* utilizado (MIB520).

### ***2.1.2 Sistemas operativos y lenguajes de programación para WSN.***

Hasta ahora se ha podido observar que las necesidades de las motas que forman las redes de sensores inalámbricos son totalmente distintas a las que pueda tener cualquier otro dispositivo de procesamiento de datos como por ejemplo un ordenador de propósito general o un teléfono móvil, por esta razón las motas de las WSN tienen sus propios sistemas operativos. Estos sistemas operativos propios no suelen ser tan complejos como los de propósito general debido a:

- Los requisitos especiales de las aplicaciones de estas redes.
- Las restricciones de recursos presentes en las plataformas *hardware* de este tipo de redes.

Uno de los primeros sistemas operativos diseñado específicamente para WSN es TinyOS que se basa en un modelo de programación concurrente y controlado por eventos. Tanto TinyOS como sus aplicaciones están escritas en un lenguaje de programación denominado NesC, que es una extensión del lenguaje de programación C que permite la reducción del tamaño del código y la RAM (*Random Acces Memory*) utilizada, características necesaria dadas las restricciones de memoria presente en estas redes.

A continuación se procede a la explicación detallada del sistema operativo TinyOS, así como el lenguaje de programación NesC. La explicación se centra en este sistema operativo y lenguaje de programación en concreto porque han sido los utilizados para la programación de las motas que forman la WSN, sin embargo existen otros sistemas operativos y lenguajes de programación disponibles descritos en [10] y [11] respectivamente.

#### **- *TinyOS.***

La principal diferencia de TinyOS con la mayoría de sistemas operativos es su cuidadoso diseño para trabajar con operaciones de muy bajo consumo de potencia. TinyOS permite la creación de aplicaciones de una forma relativamente fácil, por el hecho de proporcionar un conjunto de componentes reutilizables para formar componentes mayores que formen las aplicaciones finales. Entre estos componentes se encuentran componentes de:

- Toma de medidas o sensado.
- Comunicación.
- Almacenamiento.
- Temporizadores.
- Etc.

El diseño del *Kernel* (núcleo) de TinyOS se basa en una estructura de dos niveles de planificación:

- **Eventos:** Pensados para procesos pequeños que tienen el objetivo de informar sobre la aparición de un suceso importante con el que se debe interrumpir cualquier función ejecutándose en ese momento. Por ejemplo, la expiración de un temporizador, recepción de mensajes, etc.
- **Tareas:** Pensadas para cantidades grandes de procesamiento que no son críticas en el tiempo, de tal manera que se ejecutan de forma asíncrona siempre y cuando la CPU no tenga ningún **evento** que ejecutar. El inicio y final de las tareas son funciones separadas que pueden ser activadas a partir de un evento.

Este diseño de núcleo permite que los eventos, que tienen poca cantidad de procesamiento, puedan ejecutarse inmediatamente mientras se interrumpen las tareas, que tienen una mayor carga computacional, consiguiendo un alto rendimiento en aplicaciones de concurrencia intensiva, donde la CPU del dispositivo se utiliza de manera eficiente consumiendo una energía mínima.

TinyOS es un sistema operativo de distribución libre que se puede obtener fácilmente en la página Web oficial de TinyOS [12], en la que además de encontrar las últimas versiones del sistema operativo se puede consultar una gran variedad de tutoriales.

#### - *NesC.*

Todo y que TinyOS permita la creación de programas para las motas de forma modular, es decir, utilizando la combinación de los componentes proporcionados por TinyOS, la programación de las motas puede llegar a ser compleja debido a la limitada capacidad de cálculo y la escasa cantidad de recursos tanto energéticos como de memoria de las motas. Por otra parte, en la actualidad, no se encuentran disponibles entornos de depuración de código para los microcontroladores utilizados en las motas, con lo que dificulta la fase de localización y corrección de errores, que se debe realizar a partir de los LEDs (*Light-Emitting Diode*) disponibles en las motas y apoyándose en la transmisión de mensajes hacia la estación base y posterior visualización de los datos.

NesC (*Network Embedded Systems C*) es un lenguaje de programación con una sintaxis similar a C y está optimizado para las limitaciones de memoria y capacidad de cálculo presentes en las redes de sensores inalámbricos. NesC está orientado a componentes y especialmente diseñado para programar motas bajo el sistema operativo TinyOS.

Las aplicaciones en NesC se forman a partir de uno o más componentes entrelazados entre ellos. Estos componentes son componentes proporcionados por el sistema operativo TinyOS o componentes desarrollados por el propio programador (aplicaciones).

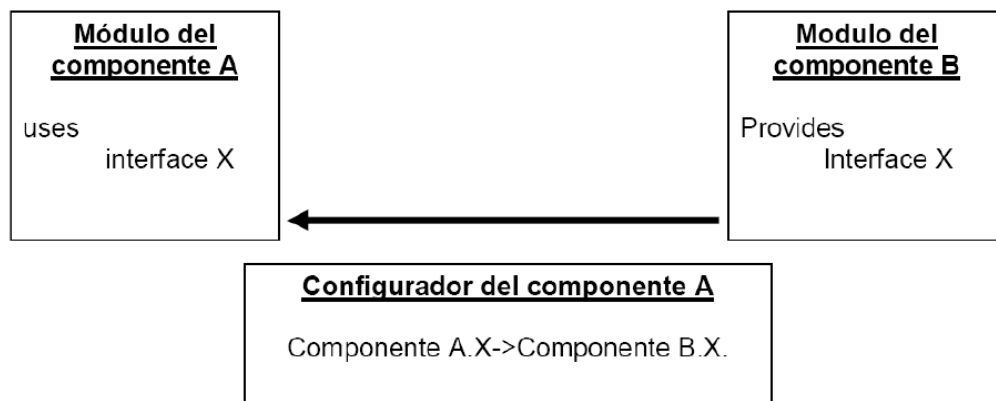
Cada uno de estos componentes proporciona interfaces para que los demás se puedan comunicar él y utiliza las interfaces proporcionadas por los demás componentes para comunicarse con ellos. Estas interfaces son bidireccionales y se puede realizar más de una instancia a la misma interfaz, por ejemplo es el caso en que un componente necesite dos temporizadores distintos, éste tendrá que realizar dos instancias a la interfaz que proporciona el temporizador. Gracias al uso de interfaces se consigue:

- Unión de componentes estática, permitiendo una mayor robustez de diseño.
- Mejor análisis del programa (mejor depuración).
- Aumento de la eficiencia en tiempos de ejecución (mayor concurrencia).

Las implementaciones de los componentes en NesC se realiza mediante:

- **Módulos:** Proporcionan el código de la aplicación, que es donde se implementan los eventos de una o varias interfaces formadas por la agrupación de comandos y otros eventos, a partir de los cuales se consigue dar respuesta a algún suceso determinado como puede ser la recepción del aviso de cierto componente.
- **Configuraciones:** Se utilizan para la unión de componentes, conectando las interfaces utilizadas con las interfaces proporcionadas de los distintos componentes que forman la aplicación. De esta forma, las configuraciones se encargan de ensamblar los distintos componentes constituyentes de la aplicación en función de las interfaces que utilizan y proporcionan.

En la **Figura 6** se muestra un ejemplo de código NesC que utiliza dos componentes, A y B, que se comunican a partir de la interfaz X, que proporciona el módulo del componente B y utiliza el módulo del componente A. En este caso se asume que el componente A representa la aplicación a programar ya que es el que utiliza la interfaz proporcionada por otro componente. De esta manera, para reflejar la unión entre el componente A y el componente B a través de la interfaz X, se debe incluir en la configuración del componente A el tipo de unión, tal y como se indica en la **Figura 6**.



**Figura 6.** Ejemplo código NesC.

La mayoría de información sobre TinyOS y NesC utilizada a lo largo del PFC, además de la página oficial de TinyOS, ha sido extraída de [13] donde se introduce y detalla de forma clara la programación en TinyOS y NesC.

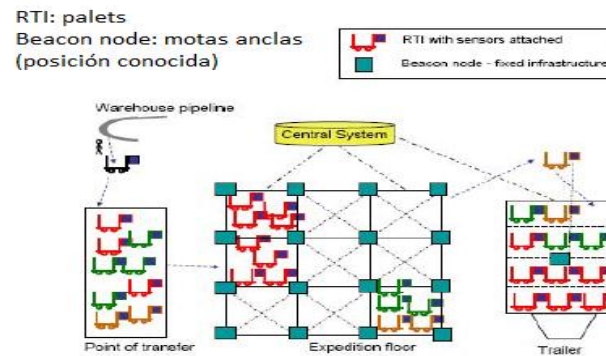
### 2.1.3 Aplicaciones de las WSN.

Para terminar de completar el capítulo teórico dedicado a las redes de sensores inalámbricos se presentan ciertas aplicaciones que actualmente se están o se han llevado a cabo. El rango potencial de las aplicaciones de las WSN es ilimitado ya que la convergencia de las tecnologías de la información y comunicaciones inalámbricas, gracias a la miniaturización, han convertido a las WSN en un área con una capacidad de crecimiento elevadísima. Entre la inmensa cantidad de aplicaciones posibles [14,15] se encuentran las siguientes:

- **Monitorización del entorno:** Las motas de una WSN se pueden utilizar para la monitorización de condiciones y movimientos de animales o plantas salvajes situados en hábitats hostiles, en los que se desea la mínima alteración del medio y los habitantes. De hecho, la monitorización del entorno es una de las primeras aplicaciones de las redes de sensores inalámbricos, siendo dos ejemplos de esta aplicación:
  - Monitorización de condiciones y movimientos de animales en la isla *Great Duck* (noreste de los Estados Unidos) [16].
  - Monitorización de un glaciar en Noruega con el objetivo de entender la dinámica del glaciar con respecto al cambio climático [17].



**Figura 7.** Monitorización del hábitat en *Great Duck*.



**Figura 8.** Control de transporte y logística de mercancías [20].

- **Medicina:** El cuidado tanto de los ancianos como los enfermos crónicos puede verse beneficiado en gran parte con el uso de las WSN, de tal manera que se podría monitorizar los signos vitales del paciente manteniéndolo conectado constante y remotamente con las consultas de los doctores. No sólo esto sino que se podría instalar una WSN en los hogares con el fin de alertar a los doctores cuando un paciente cae o básicamente para entender el movimiento y comportamiento de ciertas personas. Un ejemplo de aplicación llevada a cabo es:
  - El control del entorno de ancianos con alzhéimer realizado por Intel y la Universidad de Washington [18].
- **Control de almacenes/procesos industriales (Figura 8):** Tanto en entornos industriales como en edificios dedicados al almacenaje y transporte de objetos existen zonas de difícil acceso, por ello resulta más cómodo y económico utilizar una WSN que no una red de sensores cableada para la monitorización de procesos, seguimiento de objetos o equipamiento industrial. Ejemplos de estas aplicaciones se encuentran las siguientes:
  - Control de un tanque de petróleo en una plataforma petrolífera de BP a partir de datos sobre la vibración [19].
  - Control de transporte y logística de mercancías [20].
- **Apoyo militar:** El apoyo militar fue el primer propósito por el cual se empezó a investigar en el área de las WSN. Tener conocimiento en tiempo real del campo de batalla es esencial para el control, las comunicaciones y la toma de decisiones militares. La red de sensores puede ser desarrollada y desplegada rápidamente sin la necesidad de realizar una infraestructura, como puede ser el caso del radar, y pasar a recoger información inmediatamente. Un ejemplo es:
  - La vigilancia de fronteras desarrollando una infraestructura tecnológica que mediante el uso de varios sensores y aparatos de detección se lograba, a través del satélite Ku-band, enviar la información a un centro remoto de operaciones. Este desarrollo fue llevado a cabo por Boeing Co. Contratado en 2006 por el *Department of Homeland Security* de los Estados Unidos.

Es importante destacar que todo y que no se haya visto una aplicación directa de posicionamiento como puede ser un navegador como GPS en todas las aplicaciones comentadas es necesaria alguna técnica de posicionamiento que permita obtener la localización de algún elemento, con lo que en la mayoría de las aplicaciones relacionadas con las WSN es necesario algún tipo de técnica de localización. De esta manera queda presente la gran utilidad que puede tener el sistema de posicionamiento híbrido a desarrollar en el PFC, dado el gran número de aplicaciones actuales y que están en pleno desarrollo relacionadas con el sistema.

## 2.2 Conceptos de localización y posicionamiento.

La tarea básica de un sistema de comunicación inalámbrico, como es el caso de las WSN, es la de transmitir la información originada en un terminal hacia otro u otros terminales. No obstante, utilizando ciertas características de la señal transmitida se le puede dar otro uso a estos sistemas como puede ser la estimación de lo lejos que está un terminal de otro o la localización de un terminal concreto.

La medida de distancias y la localización son dos conceptos estrechamente relacionados. Por un lado, la medida de distancias puede considerarse como la determinación del radio de un círculo o una esfera, mientras que la localización determina un punto en el espacio descrito por la relación de varios círculos o esferas. De esta manera, las distancias se pueden utilizar para calcular localizaciones o viceversa.

Existen tres propiedades básicas que permiten la medida de distancias a partir del análisis de ciertas características físicas de las señales de radio:

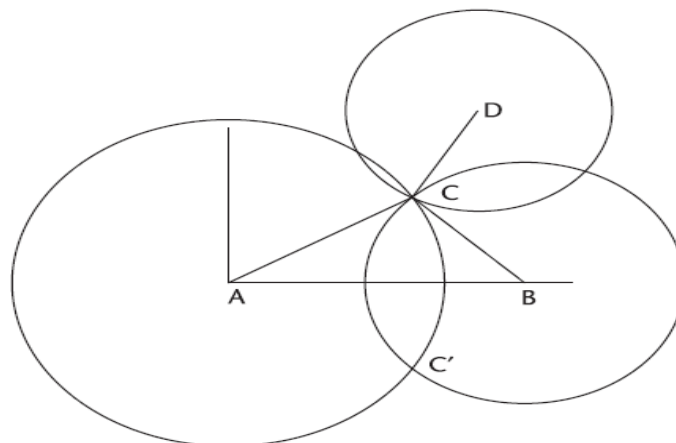
1. **Potencia recibida (RSS):** La densidad de potencia de una onda electromagnética es proporcional a la potencia transmitida e inversamente proporcional al cuadrado de la distancia respecto la fuente de transmisión. Esta ley física junto la combinación vectorial de las ondas que llegan al receptor por diferentes caminos son la base de las estimaciones de distancias basadas en medidas de RSS.
2. **Tiempo de vuelo (TOF, *Time Of Flight*):** La distancia entre un transmisor y un receptor es igual al tiempo de vuelo o tiempo de propagación de la onda electromagnética multiplicado por la velocidad de propagación, que en el caso inalámbrico es igual a la velocidad de la luz. La distancia entre el transmisor y el receptor se puede calcular a partir de la medida del tiempo de llegada (TOA, *Time Of Arrival*) de la señal al receptor siempre y cuando se conozca el tiempo o instante de transmisión, si este instante no es conocido la distancia se calcula a partir de las diferencias de los instantes de recepción en varios lugares (TDOA, *Time Difference Of Arrival*).

3. **Dirección de llegada (DOA, *Direction Of Arrival*):** El frente de onda de una señal transmitida es perpendicular a la dirección de propagación de la onda. Esta dirección puede ser estimada variando el diagrama de radiación espacial del transmisor o receptor mientras se observa el cambio en la potencia recibida. El ángulo o dirección de llegada se puede determinar como el punto en el diagrama donde la potencia de señal es máxima o tiene un nulo, dependiendo del punto de referencia de diseño. Para la estimación de la distancia no es suficiente con una medida DOA, al menos se necesitan dos medidas o una con el apoyo de medidas TOF o RSS.

Una vez estimadas las distancias entre el terminal a posicionar y los demás nodos de la red, cuyas posiciones son conocidas, se puede proceder al cálculo de la posición del terminal desconocido. En [21] se demuestra que para obtener la posición sin ambigüedad es necesario disponer al menos de 3 medidas de distancias y conocer la posición de los nodos respecto a los que se toman las medidas (nodos/motas anclas). Con estas tres distancias se describen tres círculos centrados en las respectivas coordenadas de los nodos ancla, que se cruzarán en un mismo punto correspondiente a la posición del terminal desconocido (**Figura 9**). En la **Figura 9** se observa un segundo punto C', que es el punto de ambigüedad que se tendría al utilizar únicamente dos medidas de distancia, incluyendo la tercera este punto ya no introduce ambigüedad.

La gran mayoría de métodos de localización derivan de las medidas descritas anteriormente, ya sean solas o combinadas, sin embargo hay dos variantes que no se basan en estas medidas:

1. **Proximidad:** Se basa en referir la detección de un terminal alrededor de un rango determinado en el que se sabe que se encuentra el terminal.
2. **Detección de patrones:** Localiza los terminales comparando varias características de las señales recibidas en los terminales con una base de datos del mismo tipo de características que han sido recogidas con anterioridad.



**Figura 9.** Localización utilizando círculos [21'].

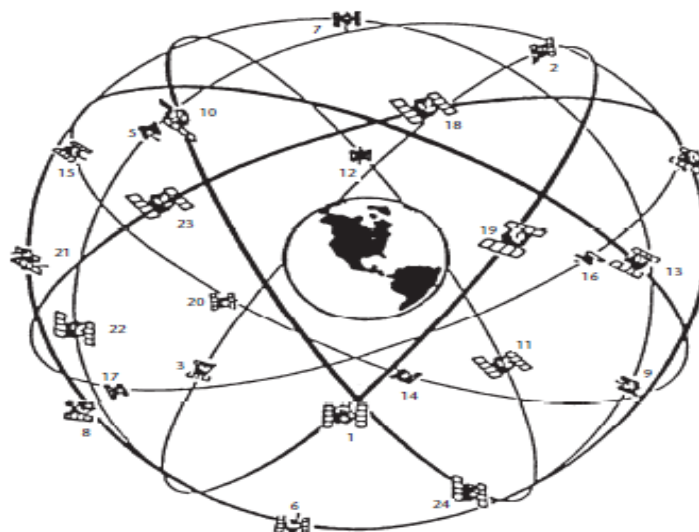


Estas variantes pueden servir como apoyo a las anteriores en el caso que no se dispongan medidas de distancias suficientes para proceder al cálculo de la posición que determina la localización del terminal. A continuación, se describe el sistema de posicionamiento global GPS para mostrar un ejemplo de sistema que utiliza medidas TOF para la localización y posteriormente, se detalla el procedimiento seguido para obtener la localización a partir de medidas RSS. Para más detalle sobre las medidas TOA/TDOA y DOA consultar [22] y [23] respectivamente.

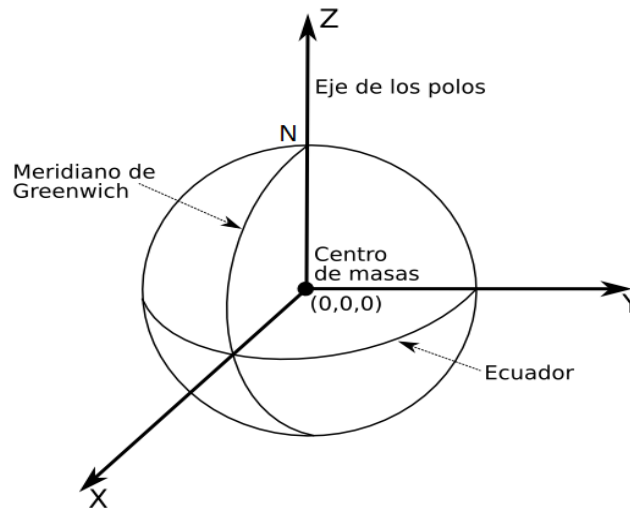
### 2.2.1 GPS (*Global Positioning System*).

GPS es sin duda el desarrollo más importante en las tecnologías de localización de la última década del siglo XX, siendo un sistema que utiliza medidas TOF/TOA para la determinación de la posición del terminal a localizar. GPS determina la posición y la velocidad en tres dimensiones relativas a coordenadas universales, siendo un sistema unilateral en el sentido que es el receptor quien calcula su propia posición analizando las señales que le llegan de distintos transmisores. Ya se ha comentado que para la obtención de la posición son necesarias tres medidas de distancia, en el caso que el receptor GPS no esté sincronizado con los relojes de los transmisores, que es la mayoría de casos, se necesitará otro transmisor para obtener una corrección del error del reloj del receptor (bias de reloj).

En GPS los transmisores se sitúan en satélites en órbita alrededor de la Tierra. Concretamente se utiliza una constelación activa de 24 satélites, de tal manera que de media 8 satélites están siempre en visión directa con cualquier punto de la Tierra. La **Figura 10** es una ilustración de la constelación GPS. Es importante que el receptor GPS conozca con alta precisión la posición exacta de los satélites que tiene en visión, para ello cada satélite informa de su posición exacta en los mensajes que transmite.



**Figura 10.** Constelación de satélites GPS [24].



**Figura 11.** Sistema de coordenadas ECEF.

El sistema de coordenadas utilizado para el posicionamiento básico en GPS es el ECEF (*Earth Centered- Earth Fixed*), que es un sistema de coordenadas cartesiano con las siguientes características (**Figura 11**):

- Origen de coordenadas situado en el centro de gravedad de la Tierra.
- El eje  $x$  apunta hacia la intersección del ecuador con la línea de longitud  $0^\circ$  o meridiano de Greenwich.
- El eje  $z$  apunta hacia el polo Norte.
- El eje  $y$  es perpendicular a los anteriores ejes y apunta en dirección contraria al centro de masa (sale del centro de masa).

Este sistema de coordenadas es el utilizado para obtener unas coordenadas  $x, y, z$  del terminal a localizar para posteriormente trasladarlas a coordenadas geográficas (latitud, longitud y altura) que son más prácticas a la hora de representarlas en un mapa de referencia. La base para realizar esta transformación de coordenadas es la descripción geodésica de la Tierra denominada WGS-84 [25].

Actualmente, se pueden encontrar receptores GPS con una precisión horizontal menor de 22 m y una precisión vertical menor de 27.7 m el 95% del tiempo si se toman medidas de distancia de 4 satélites, sin embargo al tomar en consideración más de 4 satélites se puede llegar a un error menor de 10 m, que es un error más que aceptable para localizaciones en la Tierra a nivel global. Entre los factores que contribuyen a errores se tiene:

- Imprecisiones en el tiempo de satélite y posición.
- Cambios en la velocidad de propagación de las señales debido a efectos de la troposfera e ionosfera.
- El denominado *geometric dilution of precision (GDOP)* que es un factor multiplicativo en el error relacionado con la geometría de los satélites en visión con el receptor. Lo ideal es que los satélites estén lo más separados posible entre ellos.

### 2.2.2 *Sistemas de localización basados en medidas de RSS.*

GPS es un sistema basado en técnicas de estimación de distancias TOF, sin embargo en los últimos tiempos han aparecido varios sistemas de localización basados en técnicas de estimación de distancias RSS debido a su capacidad de trabajar tanto en entornos exteriores como en entornos interiores en los que GPS no funciona. Las medidas de distancia RSS se basan en la observación de la variación media de la potencia de la señal recibida en un receptor, de esta manera y conociendo la variación de la potencia recibida en función de la distancia es posible determinar la distancia entre dos terminales. Para el caso de una WSN, involucrando en la medida de las distancias varias motas, la posición de la mota desconocida se puede calcular mediante el método expuesto en la **Figura 9** (círculos).

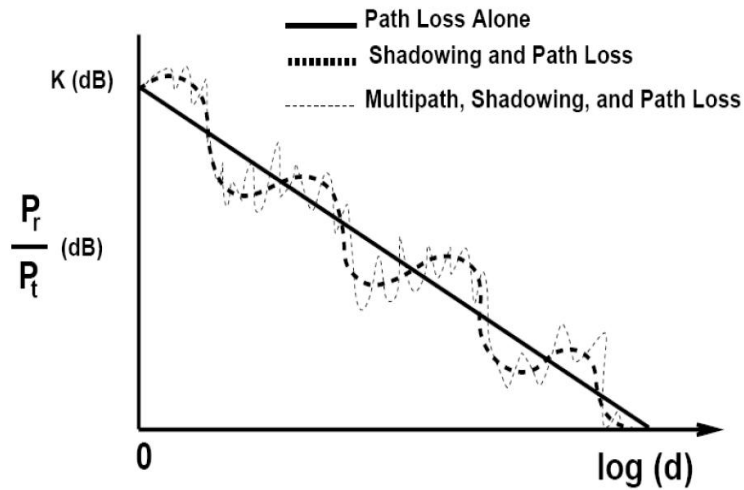
La relación entre la potencia recibida y la distancia entre transmisor y receptor depende principalmente del entorno en el que se encuentran los elementos de la comunicación. La mejor relación de este tipo se encuentra en el espacio libre, el resto de entornos contienen objetos que reflejan, absorben o dispersan la onda electromagnética introduciendo efectos probabilísticos, que varían con el tiempo, en la relación entre la potencia y la distancia convirtiéndose en un proceso estocástico cuyas propiedades probabilísticas o aleatorias dependen del entorno.

En espacio libre, la relación entre la potencia recibida y la distancia viene dada por la ecuación de Friis representada en (1), donde  $G_r$  y  $G_t$  representan la ganancia del receptor y transmisor respectivamente,  $d$  es la distancia entre ambos y  $\lambda$  es la longitud de onda a la que se trabaja ( $\lambda = c/f$ ), siendo  $P_t$  la potencia del transmisor y  $P_r$  la potencia en el receptor.

$$P_r = \frac{P_t \cdot G_t \cdot G_r \cdot \lambda^2}{(4\pi)^2 \cdot d^2} \quad (1)$$

Se define la ganancia de canal (PG, *Path Gain*) como la relación entre la potencia recibida y la potencia transmitida, a partir de la cual se puede observar directamente el efecto de la potencia recibida o RSS. Con la expresión del PG, mostrada en (2), se podría calcular la distancia a la que se encuentra el receptor asumiendo que los valores de ganancia y potencia en transmisión y recepción así como la frecuencia de funcionamiento son conocidos con exactitud.

$$PG = \frac{P_r}{P_t \cdot G_t \cdot G_r} = \left( \frac{\lambda}{4\pi \cdot d} \right)^2$$
$$PG_{dB} = 20 \cdot \log_{10} \left( \frac{\lambda}{4\pi \cdot d} \right) \quad (2)$$



**Figura 12.** Efecto del PL, *shadowing* y *multipath* [26].

En la práctica, la relación entre la señal recibida y la distancia no vendrá dada por la expresión (1) ni el PG será el representado en (2) ya que la potencia recibida dependerá de la suma de una serie de señales que llegan por distintas direcciones debido a reflexiones con objetos cercanos o que bloquean parcial o totalmente la visión directa. De esta manera, la potencia recibida resultante puede ser mayor o menor que la potencia en espacio libre descrita por (1). Además, si los objetos reflectantes se mueven a medida que pasa el tiempo respecto los receptores, la potencia recibida variará con el tiempo lo que implica que para una misma distancia la potencia recibida variará a lo largo del tiempo. Otro factor a tener en cuenta son los grandes obstáculos como edificios, suelo o paredes en línea de visión directa con el receptor, ya que atenúan la señal pudiendo reducir significativamente el nivel de potencia recibida. Todos estos factores que hacen variar la potencia recibida o equivalentemente las pérdidas del canal PL (*Path Loss*), se reúnen en dos efectos físicos denominados *shadowing* y *multipath* (**Figura 12**) [26', 27]:

- **Shadowing:** Causado por obstáculos situados entre el transmisor y el receptor produciendo atenuación en la señal a causa de la absorción, reflexión y difracción. Existen casos extremos en los que la atenuación es lo suficientemente grande como para eliminar totalmente el nivel de señal.
- **Multipath:** Causado por los objetos que reflejan la señal transmitida haciendo que en el receptor se obtenga la suma de señales que llegan por distintos caminos y por lo tanto con distintas fases. Estas diferencias de fases pueden provocar en la potencia recibida resultante tanto interferencias destructivas como constructivas a lo largo del tiempo. El resultado de estas interferencias hace que la señal recibida y por lo tanto su potencia varíe muy rápido.

La **Figura 12** deja presente que entornos de comunicaciones complejos, como es el caso de entornos inalámbricos, no pueden ser modelados de forma precisa con el modelo de espacio libre. Por ello, a lo largo de los años, se han desarrollado un gran número de modelos que intentan predecir el PL de entornos concretos. Estos modelos se basan en medidas empíricas realizadas a distintas distancias para un rango de frecuencia dado y en un área geográfica o edificio particular.

Entre estos modelos se encuentran:

- **Okumura** [28]: Utilizado en grandes ciudades y es aplicable para distancias en el rango [1 – 100 Km] y el rango de frecuencias [150 – 1500 MHz]
- **Hata** [29]: Formulación empírica del PL proporcionado por Okumura, siendo válido para el mismo rango de distancias y frecuencias.
- **COST 231** [30]: Extensión del modelo Hata hasta los 2 GHz.

- ***Parametrización del escenario.***

Se ha comprobado que la relación entre la potencia recibida y la distancia puede llegar a ser muy compleja con lo que para obtener estimaciones precisas de la distancia se crean modelos que permiten obtener buenas predicciones de las pérdidas de propagación, sin embargo estos modelos creados modelan escenarios muy concretos que pueden ser útiles para la predicción de pérdidas de propagación en el diseño de radio enlaces de larga distancia pero no son tan útiles para la estimación de distancias utilizadas para la localización en entornos complejos. Aun así se utiliza la idea de estos modelos para la definición de unos parámetros que permitan modelar los escenarios de localización y obtener así de forma precisa las pérdidas de propagación en ese entorno y que permitan la obtención de distancias entre motas.

Para definir estos parámetros con los que se obtendrá la relación entre la potencia recibida y la distancia, se define  $\gamma$  como el factor de decaimiento de la RSS en función de la distancia, es decir,  $\gamma$  es el exponente que acompaña a la distancia en la fórmula de la potencia recibida, en el caso de la fórmula de Friis (1) este exponente es 2 correspondiéndose al factor de decaimiento de la RSS en función de la distancia. De esta manera la expresión de Friis queda:

$$P_r = P_t \cdot G_t \cdot G_r \cdot \left(\frac{\lambda}{4\pi}\right)^2 \cdot \frac{1}{d^\gamma}$$

Definiendo  $k$  como el conjunto de constantes en la expresión anterior y tomando la potencia recibida en decibelios, teniendo en cuenta que  $n = 10 \cdot \log_{10}(k)$  se obtiene la expresión (3) que describe la relación entre la RSS y la distancia (en dB) en función de los parámetros  $n$  y  $\gamma$ :

- **$n$** : Representa los parámetros que se mantienen constantes durante la transmisión, cuyos valores se obtendrían de las especificaciones del fabricante. No obstante, las especificaciones pueden ser más o menos exactas además de variar según las circunstancias, por este motivo se define el parámetro  $n$  que incluye los valores reales de las ganancias, potencia transmitida y frecuencia de trabajo. Por otro lado, observando (3),  $n$  es la potencia recibida a 1 m de distancia ( $d_{dB} = 0$ ), que dependerá principalmente de todos estos parámetros constantes.

- $\gamma$ : Representa la tasa de disminución de la potencia en función de la distancia que dependerá básicamente de la complejidad del entorno (*shadowing y multipath*).

$$P_{r\_dB} = RSS = n - \gamma \cdot d_{dB} \quad (3)$$

Para obtener el valor de los parámetros en los diferentes escenarios de localización se realizan medidas empíricas de potencia en varias distancias, definiendo el vector  $\mathbf{z}$  como el conjunto de potencias recibidas medidas. Teniendo en cuenta que la expresión en (3) representa un único valor de potencia, se puede obtener el sistema de ecuaciones representado en forma matricial en (4), siendo  $\alpha_i = 10 \cdot \log_{10}(d_i) = d_{i\_dB}$ .

$$\mathbf{rss} = \begin{bmatrix} RSS_1 \\ RSS_2 \\ \vdots \\ RSS_N \end{bmatrix} = \begin{bmatrix} n - \gamma \cdot \alpha_1 \\ n - \gamma \cdot \alpha_2 \\ \vdots \\ n - \gamma \cdot \alpha_N \end{bmatrix} = \begin{bmatrix} 1 & \alpha_1 \\ 1 & \alpha_2 \\ \vdots & \vdots \\ 1 & \alpha_N \end{bmatrix} \cdot \begin{bmatrix} n \\ -\gamma \end{bmatrix} = \mathbf{A} \cdot \mathbf{b} \quad (4)$$

De esta manera se puede definir el vector error  $\mathbf{e}$  como  $\mathbf{e} = \mathbf{z} - \mathbf{rss} = \mathbf{z} - \mathbf{A} \cdot \mathbf{b}$  y aplicar el método *Least Squares* (LS) [31] para encontrar el vector  $\mathbf{b} = \begin{bmatrix} n \\ -\gamma \end{bmatrix}$  que minimiza el módulo al cuadrado del vector  $\mathbf{e}$ , que representa el error entre la potencia medida y la potencia que predeciría el modelo generado con los parámetros  $n$  y  $\gamma$  para cada una de las distancias a las que se han tomado las medidas de potencia. Aplicando LS al vector  $\mathbf{e}$  se obtiene el resultado de la expresión (5).

$$\mathbf{b} = \min_{\mathbf{b}} \|\mathbf{e}\|^2 = \min_{\mathbf{b}} \|\mathbf{z} - \mathbf{A} \cdot \mathbf{b}\|^2 = \begin{bmatrix} n \\ -\gamma \end{bmatrix} \quad (5)$$

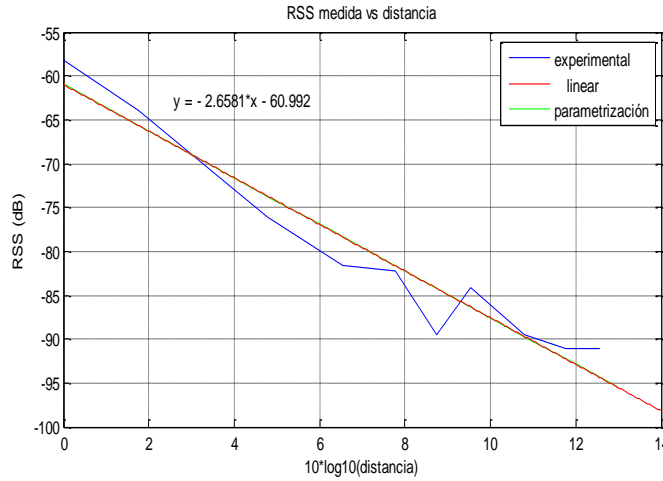
$$\mathbf{b} = (\mathbf{A}^H \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^H \cdot \mathbf{z}$$

Así, realizando medidas empíricas de potencia recibida para diferentes distancias y utilizando la expresión en (5) se consiguen los parámetros que proporcionan la mejor aproximación de la RSS en función de la distancia, en el sentido que minimiza el error entre las potencias reales medidas y las potencias estimadas por el modelo. Cabe destacar que el error mínimo se consigue en el escenario en cuestión que se han tomado las medidas, si el escenario cambia se debe volver a tomar medidas y obtener unos parámetros nuevos que modelen el nuevo escenario de la mejor forma posible.

Un ejemplo de medidas experimentales y su parametrización correspondiente se muestra en la **Figura 13**. Con los resultados experimentales obtenidos en la **Tabla 1** y utilizando la expresión (5) se obtiene  $n = -60.992$  y  $\gamma = 2.658$ , que tal y como se ve en la **Figura 13** describen la misma recta (parámetros  $n$  y  $\gamma$ ) que la obtenida tomando la regresión lineal de los datos experimentales.

DISTANCIA (m)	1	1.5	3	4.5	6	7.5	9	12	15	18
RSS (dBm)	-58	-64	-76	-82	-82	-89	-84	-90	-91	-91

**Tabla 1.** Resultados experimentales de ejemplo.



**Figura 13.** Resultados experimentales, parametrización y regresión lineal.

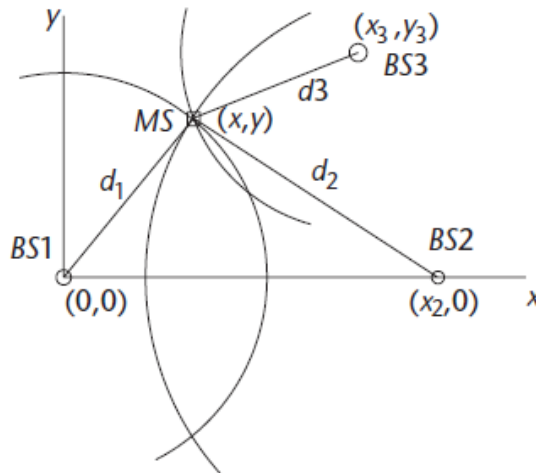
Una vez obtenidos los parámetros que modelan el escenario de localización, de la expresión (3) se puede obtener la estimación de distancia a partir de la medida de la RSS resultando en la siguiente expresión:

$$d_{dB} = 10 \cdot \log_{10} d = \frac{n - RSS}{\gamma}$$

$$\hat{d} = 10 \cdot \log_{10} d = 10^{\frac{n - RSS}{10 \cdot \gamma}} \quad (6)$$

### 2.2.3 Cálculo de la posición o localización.

Ya se ha comentado que para la obtención de la posición de un terminal sin ningún tipo de ambigüedad se necesita como mínimo tres medidas de distancia para generar tres círculos que se crucen en un mismo punto cuyas coordenadas sean las del terminal a localizar. El escenario a analizar es el mostrado en la **Figura 14**, donde se observa un terminal o estación móvil (*Mobile Station*, MS) cuyas coordenadas se desean obtener a partir de tres estaciones base (*Base Station*, BS) que se encuentran a distancias  $d_1$ ,  $d_2$  y  $d_3$  de la MS y cuyas posiciones son conocidas.



**Figura 14.** Escenario de localización [32].

Si las distancias reales  $d_1$ ,  $d_2$  y  $d_3$  pudieran ser medidas de forma exacta, las coordenadas de MS serían el punto situado en la intersección de los círculos centrados en las BS y radio igual a las distancias medidas. Teniendo en cuenta que las distancias medidas pueden estar sujetas a imprecisiones, el posicionamiento puede ser mejorado incorporando un número mayor de BS al mínimo requerido. De esta manera, suponiendo que se dispone de N estaciones base, las ecuaciones que describen los círculos definidos por la posiciones de las motas anclas (BS1, BS2,..., BSN) y las distancias medidas  $d_1$ ,  $d_2$ ,...,  $d_N$  se representan en (7).

$$\begin{aligned}(x_1 - x)^2 + (y_1 - y)^2 &= d_1^2 \\(x_2 - x)^2 + (y_2 - y)^2 &= d_2^2 \\&\vdots \\(x_N - x)^2 + (y_N - y)^2 &= d_N^2\end{aligned}\tag{7}$$

Desarrollando los cuadrados y restando la primera ecuación al resto, con un poco de cálculo se obtiene el siguiente sistema de ecuaciones formado por N-1 ecuaciones:

$$\begin{aligned}(x_1 - x_2) \cdot x + (y_1 - y_2) \cdot y &= \frac{1}{2} \cdot (x_1^2 - x_2^2 + y_1^2 - y_2^2 + d_2^2 - d_1^2) \\(x_1 - x_3) \cdot x + (y_1 - y_3) \cdot y &= \frac{1}{2} \cdot (x_1^2 - x_3^2 + y_1^2 - y_3^2 + d_3^2 - d_1^2) \\&\vdots \\(x_1 - x_N) \cdot x + (y_1 - y_N) \cdot y &= \frac{1}{2} \cdot (x_1^2 - x_N^2 + y_1^2 - y_N^2 + d_N^2 - d_1^2)\end{aligned}$$

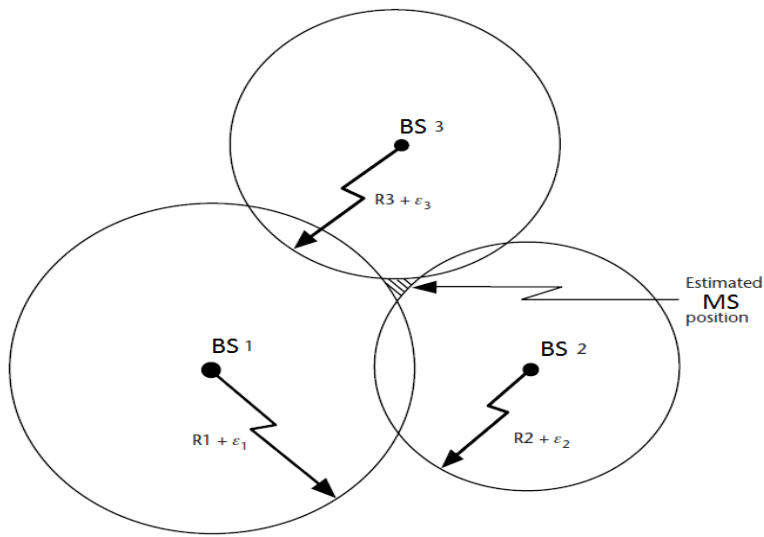
Que siendo:

$$\begin{aligned}\mathbf{B} &= \begin{bmatrix} x_1 - x_2 & y_1 - y_2 \\ x_1 - x_3 & y_1 - y_3 \\ \vdots & \vdots \\ x_1 - x_N & y_1 - y_N \end{bmatrix} \\ \mathbf{c} &= \frac{1}{2} \cdot \begin{bmatrix} x_1^2 - x_2^2 + y_1^2 - y_2^2 + d_2^2 - d_1^2 \\ x_1^2 - x_3^2 + y_1^2 - y_3^2 + d_3^2 - d_1^2 \\ \vdots \\ x_1^2 - x_N^2 + y_1^2 - y_N^2 + d_N^2 - d_1^2 \end{bmatrix} \\ \mathbf{p} &= \begin{bmatrix} x \\ y \end{bmatrix}\end{aligned}$$

Puede representarse en forma matricial como:

$$\mathbf{B} \cdot \mathbf{p} = \mathbf{c}\tag{8}$$





**Figura 15.** Efecto en localización de errores en la medida de distancia [32’].

Donde las únicas variables desconocidas son las coordenadas del terminal a localizar, MS, descritas por el vector  $\mathbf{p}$ . La matriz  $\mathbf{B}$  se forma con las coordenadas de las estaciones base, BS, con las que se realiza el proceso de localización, y el vector  $\mathbf{c}$  se forma a partir de estas mismas coordenadas y las distancias medidas entre la MS y las BS participantes en la localización.

De esta manera, formando la matriz  $\mathbf{B}$  y el vector  $\mathbf{c}$  se podría resolver el sistema de ecuaciones resultante, sin embargo las distancias medidas se obtienen con alguna de las técnicas de estimación de distancias comentadas con lo que estas distancias no serán exactas y por lo tanto es posible que los círculos no se crucen en un mismo punto o incluso que no se crucen en ninguno (**Figura 15**). Teniendo en cuenta estas posibilidades es necesario establecer un criterio para decidir las coordenadas del terminal a localizar en estos casos, ya que el sistema de ecuaciones descrito en (8) proporcionará infinitas soluciones. Este criterio se basa en el criterio del error LS, es decir, se utiliza el método LS para estimar la posición del MS de tal manera que las coordenadas estimadas  $(x_e, y_e)$  minimicen la siguiente función  $F$ , siendo  $x_i$  e  $y_i$  las coordenadas de las BS y  $\hat{d}_i$  las distancias entre MS y BS estimadas:

$$F = \sum_{i=1}^N \left( \sqrt{(x_i - x_e)^2 + (y_i - y_e)^2} - \hat{d}_i \right)^2 \quad (9)$$

La función  $F$  de (9) representa el error presente entre las distancias estimadas (entre MS y BS) y las distancias resultantes entre la posición estimada y las motas anclas. A partir del sistema representado en (8) se obtiene la siguiente solución cerrada a la minimización de la función  $F$  [33]:

$$\mathbf{p} = \begin{bmatrix} x_e \\ y_e \end{bmatrix} = (\mathbf{B}^T \cdot \mathbf{B})^{-1} \cdot \mathbf{B}^T \cdot \mathbf{c} \quad (10)$$

Esta solución permite obtener las coordenadas de un terminal a partir de la medida de distancias con más de tres estaciones base, sin embargo la solución minimiza la función  $F$  que se basa en las estimaciones de distancia entre terminal y estaciones base, con lo que la inclusión de más estaciones bases que las necesarias para la localización puede ser perjudicial para la estimación de la posición si las distancias estimadas no son precisas. Esto hace que para algunos casos, donde los métodos de estimación de distancia no son lo suficientemente precisos, la solución propuesta no proporcione buenos resultados y sea mejor utilizar únicamente las tres medidas mínimas necesarias para el cálculo de la posición del terminal a localizar.

Por otro lado, la solución proporcionada implica el cálculo de la inversa de matrices, algo que puede llegar a ser complicado de programar en ciertos entornos como el caso de las redes de sensores. Ya se ha hablado de la limitada capacidad de cálculo de las motas que forman las WSN, con lo que es habitual utilizar directamente las ecuaciones cerradas proporcionadas por la expresión (10) utilizando sólo tres medidas de distancia.



### 3 Técnicas de localización híbridas.

Ya se ha comentado en algún momento que GPS ha sido uno de los desarrollos más importantes en las tecnologías de localización gracias a que permite localizar cualquier punto del planeta con una precisión o error menor de 10 metros, dando lugar a innumerables aplicaciones relacionadas con la localización en el planeta a nivel global. Todo y ser el sistema de localización más empleado en la actualidad, GPS tiene ciertas limitaciones o inconvenientes que hacen que su uso sea inadecuado en ciertos entornos de localización:

- El uso de GPS no tiene mucho sentido en áreas pequeñas como puede ser el interior de un edificio o lugares muy concretos donde un error de 10 metros puede ser demasiado grande.
- El receptor GPS necesita visión directa con los satélites, sin embargo hay situaciones en las que no se encuentran los satélites suficientes en visión directa para permitir la localización, entre estas situaciones se encuentran:
  - Ciudades con rascacielos que bloqueen la visión directa.
  - Calles demasiado estrechas cuyos edificios pueden obstruir la visión directa de los satélites disponibles en ese momento.
  - Túneles en los que no se tiene ningún tipo de visibilidad y por lo tanto se pierde la conexión al sistema.
  - Interiores de edificios.
  - Situaciones climatológicas adversas como grandes lluvias o tormentas.
  - Bosques y zonas poco descubiertas donde la atenuación de la señal GPS se incrementa.
- Una característica básica de GPS es que proporciona coordenadas con referencia absoluta con lo que normalmente se requiere de un mapa para que el usuario pueda ubicarse, ya que la información de longitud, latitud y altitud no suelen ser de interés para el usuario.

Estos inconvenientes hacen que GPS no pueda utilizarse para la localización en interiores además de haber situaciones en exteriores donde tampoco funciona muy bien, por este motivo se muestra necesaria alguna técnica de posicionamiento que permita la localización en interiores. Una posible técnica sería la utilización de medidas de distancias a partir de la RSS, cuyas características principales y ventajas e inconvenientes respecto GPS o las técnicas TOA en general son las siguientes:

- Las técnicas RSS se basan en la medida de la potencia recibida, medida que cualquier sistema de comunicaciones inalámbrico puede llevar a cabo sin la necesidad de incluir ningún tipo de *hardware*.
- La técnica RSS no depende del tipo de modulación, la tasa de datos ni el tiempo del sistema, con lo que es una técnica ideal para dispositivos con capacidad de cálculo y *hardware* limitado como es el caso de las WSN. En las técnicas TOA estos factores se deben tener en cuenta además de necesitar sistemas de coordinación y sincronización entre transmisor y receptor.
- Las variaciones de la RSS debido a interferencias y al efecto *multipath* presente en entornos inalámbricos hace que la precisión de estas técnicas sea menor que la precisión obtenida con las técnicas TOA, ya que las estimaciones de distancias son mejores y por lo tanto es posible incluir más medidas de las necesarias y mejorar de esta manera el posicionamiento, mientras que las distancias estimadas con RSS al ser más erróneas no permiten una mejora en el posicionamiento al incluir más medidas de distancia que las necesarias.
- La propagación de las ondas electromagnéticas en medios inalámbricos es específica en cada escenario o lugar de propagación, causa por la cual las técnicas RSS no proporcionan los mismos resultados en todos los escenarios en el sentido que la precisión variará en función de la coincidencia entre el modelo de propagación utilizado y el entorno real de transmisión.

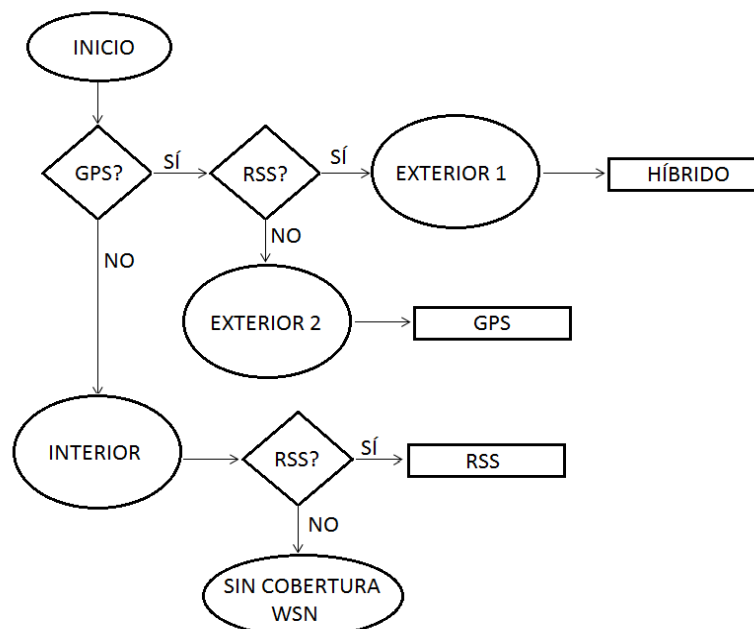
De esta manera, todo y sus limitaciones, con las técnicas RSS es posible crear un sistema de posicionamiento para interiores a partir del cual, con la combinación de un sistema de localización exterior como GPS, se puede obtener un sistema de posicionamiento híbrido que permita la localización tanto en entornos exteriores, utilizando la localización proporcionada por el sistema GPS, como en entornos interiores, utilizando la posición estimada a partir de las distancias medidas con RSS. Con las redes de sensores inalámbricos se dispone de la capacidad de obtener medidas RSS para conseguir el posicionamiento en interiores y con la ayuda de alguna *sensor-board* que permita la localización GPS se consigue el posicionamiento en exteriores. De esta forma se da solución en el presente PFC al problema de la localización híbrida.

Además, las WSN se pueden desplegar en entornos exteriores para utilizar las medidas RSS para un posicionamiento que sirva de apoyo a GPS para las situaciones comentadas que disminuye la precisión. En este PFC el sistema híbrido no sólo selecciona una técnica de localización según el tipo de escenario, sino que si se dispone de ambas técnicas en un mismo momento permite combinarlas de tal forma que el resultado sea una estimación más próxima a la realidad que las estimaciones aisladas de cada una de las técnicas de localización.

En el diagrama de flujo de la **Figura 16** se observa el funcionamiento del sistema a diseñar, donde se distinguen tres casos básicos:

- **Exterior 1:** Será el caso de disponer medidas RSS y posicionamiento GPS y por lo tanto el caso en que se utiliza la técnica de posicionamiento híbrido para obtener una mejor localización. Estos casos sólo ocurrirán en entornos exteriores donde se obtenga localización mediante GPS y se esté dentro del rango de alguna WSN que permita la medida de RSS.
- **Exterior 2:** Caso de disponer señal GPS pero no estar dentro del rango de cobertura de alguna WSN. En este caso el sistema de localización híbrido proporciona la posición obtenida de GPS.
- **Interior:** Caso de no disponer señal GPS. En esta situación el sistema híbrido supone que el usuario está en un entorno interior y por lo tanto espera estar al alcance de alguna WSN para localizarse mediante RSS.

En el momento que se pierde alguna señal (GPS, WSN) o se obtiene alguna nueva el sistema vuelve al estado de inicio para decidir el caso en el que se encuentra. Entre los casos anteriores no se distingue la posibilidad de estar en un entorno exterior y no recibir señal GPS ya que el caso de no recepción GPS es el caso de entornos interiores. Ya se ha comentado que la WSN además de brindar posicionamiento en interiores puede servir de apoyo a la localización GPS en exteriores donde tiene problemas, por ello estos casos se pueden englobar en la parte del diagrama de flujo relacionada con el caso **INTERIOR**.



**Figura 16.** Diagrama de flujo del sistema de posicionamiento híbrido.

Con el diagrama de flujo queda presente la capacidad del sistema híbrido de proporcionar posicionamiento tanto en entornos interiores o entornos exteriores, ya sean complejos o no, a partir de la identificación de señales GPS y WSN. Sin embargo, ya se ha comentado en varios momentos que la capacidad principal del sistema híbrido es la de combinar las señales, en caso de estar disponibles, para obtener una localización más precisa. Para el caso del PFC y como se puede observar en el diagrama de flujo, la única combinación que se hace es con señales GPS y WSN y sólo en entornos exteriores ya que se proponía dar solución de posicionamiento en los lugares donde GPS no funcionaba. No obstante, esta idea de combinación de señales se podría extender en entornos interiores con el objetivo de mejorar el posicionamiento en interiores proporcionado por las WSN, por ejemplo, con la combinación de señales Wi-Fi que permitan otro tipo de posicionamiento mediante medidas RSS.

La primera idea que se puede ocurrir a la hora de combinar el posicionamiento RSS con el de GPS sería la de tomar la media de esas dos medidas. Básicamente esto es lo que haría cualquier estimador si ambas medidas provinieran de la misma fuente con el objetivo de reducir su varianza y no introducir bias. Sin embargo, en el caso que nos ocupa las medidas de posición proporcionadas por GPS y las obtenidas con RSS no provienen de la misma medida y por lo tanto cada una tiene sus propias fuentes de errores con lo que cada una tendrá su propia precisión que dependerá de sus características aleatorias. De esta manera, parece más inteligente realizar algún tipo de combinación lineal de tal manera que se tenga más en cuenta las medidas menos erróneas y se menosprecie las medidas menos precisas.

A continuación se proponen técnicas para obtener combinaciones lineales entre los resultados GPS y RSS que proporcionen mejores resultados, concretamente se procede a la obtención de los dos parámetros que forman la siguiente combinación lineal, siendo  $\hat{P}_{RSS}$  y  $\hat{P}_{GPS}$  las estimaciones de posición con RSS y GPS respectivamente y  $\hat{P}_H$  la estimación del estimador híbrido:

$$\hat{P}_H = \alpha \cdot \hat{P}_{RSS} + \beta \cdot \hat{P}_{GPS} \quad (11)$$

### 3.1 Ponderación.

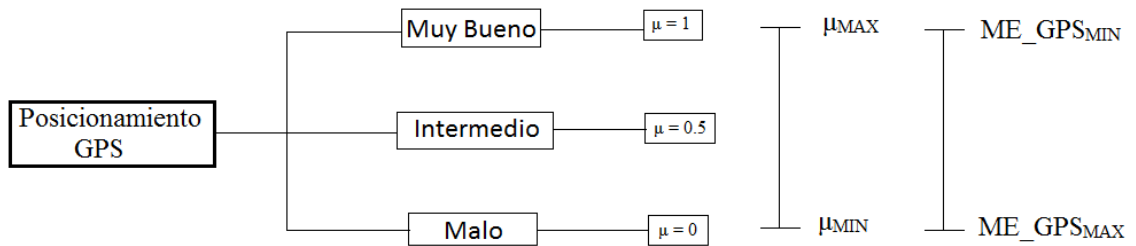
Se ha visto que para el caso del sistema a implementar en el PFC llevado a cabo el único caso de combinación de posiciones ocurre en exteriores con el objetivo de mejorar el funcionamiento de GPS en aquellas situaciones en las que no funciona adecuadamente. Por esta razón es lógico definir un parámetro  $\mu$  que represente la calidad de las estimaciones GPS y representar la expresión (11) como sigue:

$$\hat{P}_H = (1 - \mu) \cdot \hat{P}_{RSS} + \mu \cdot \hat{P}_{GPS} \quad (12)$$

Así, la estimación híbrida tendrá más en cuenta la estimación GPS o la estimación RSS según la calidad del posicionamiento GPS.

El parámetro  $\mu$  debe ser un número comprendido entre 0 y 1 de tal manera que cuando las estimaciones GPS sean correctas la estimación híbrida solo tomará en cuenta su posición ( $\mu = 1$ ) y cuando las estimaciones GPS sean muy erróneas la estimación híbrida sólo tendrá en cuenta las estimaciones RSS y anulará las de GPS ( $\mu = 0$ ), en el caso intermedio se tendría  $\mu = 0.5$  que sería el caso que ambas estimaciones (RSS y GPS) son de la misma calidad y por lo tanto la estimación híbrida toma la media de ambas estimaciones.

Una vez definido el parámetro a utilizar para la combinación de las estimaciones GPS y RSS y la expresión del estimador asociado (12) falta por comentar el método para obtener el parámetro  $\mu$  que representa la calidad de las estimaciones GPS. Una manera de representar la calidad de un sistema de posicionamiento o localización es el error obtenido en las estimaciones de las distintas posiciones estimadas, sin embargo actualmente no existe ningún método para obtener los errores en cada una de las posiciones instantáneas. En el caso de GPS si que hay receptores que proporcionan el GDOP que puede dar una idea de lo precisa que es la estimación de la posición que está mostrando, pero el error en GPS no sólo depende del GDOP y por lo tanto puede haber casos en los que el valor de GDOP es bueno mientras que en las estimaciones afectan otros efectos no relacionados con el GDOP. Lo que se suele proporcionar, normalmente como especificación del sistema, es un valor medio del error del sistema de localización, que ha sido medido a partir de pruebas experimentales del sistema conociendo con anterioridad las posiciones que posteriormente se estiman para poder calcular el error cometido en las diferentes posiciones.



**Figura 17.** Relaciones entre ME\_GPS y el parámetro  $\mu$ .



Teniendo esto en cuenta, la técnica denominada **ponderación** propuesta en este PFC se basa en el valor del error medio obtenido con posicionamiento GPS (ME\_GPS, *Mean Error GPS*) en los distintos escenarios en los que se aplica el sistema híbrido. Así, para que se cumplan las consideraciones del parámetro  $\mu$  se debe cumplir que cuando el ME\_GPS aumente  $\mu$  disminuya y viceversa, es decir, el parámetro  $\mu$  debe ser inversamente proporcional al error medio del posicionamiento GPS, estas relaciones se detallan en la **Figura 17**. Para conseguir la relación entre ME\_GPS y  $\mu$  mostrada basta con calcular el parámetro  $\mu$  a partir del inverso del error medio GPS, es decir:

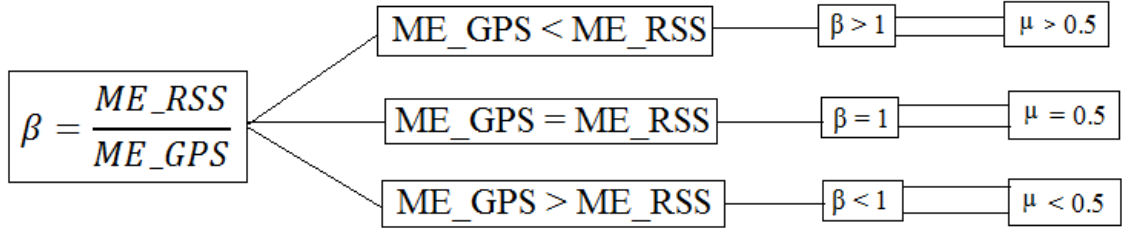
$$\mu = \frac{1}{ME\_GPS} \quad (13)$$

La expresión (13) permite obtener un parámetro  $\mu$  que es inversamente proporcional a ME\_GPS pero habrá que establecer un criterio para decidir si el posicionamiento GPS es malo y por lo tanto asignar un valor de  $\mu = 0$  o por el contrario el posicionamiento es muy bueno y por lo tanto asignar un valor de  $\mu = 1$ , ya que utilizando directamente la expresión (13) nunca se obtendrán los valores de  $\mu$  mencionados cuando el posicionamiento GPS sea bueno o malo. Hay que tener en cuenta que posicionamientos GPS malos pueden dar errores del orden de 20-30 metros y posicionamientos buenos del orden de 3-5 metros.

Para decidir si el posicionamiento GPS es bueno o malo en un determinado escenario se podría definir un ME\_GPS mínimo y otro máximo que marquen el nivel a partir del cual la localización GPS es buena o mala respectivamente. De esta manera, si el ME\_GPS obtenido en un escenario es menor que el ME\_GPS mínimo definido, en ese escenario se utilizará una  $\mu = 1$ . En el otro extremo, si el ME\_GPS obtenido en un escenario es mayor que el ME\_GPS máximo definido, en ese escenario se utilizará una  $\mu = 0$ . Con este criterio que decidirá los casos en los que sólo hay que tomar en cuenta el posicionamiento GPS o no hay que tomarlo en cuenta, y utilizando (13) para la obtención de  $\mu$  y (12) para la obtención de la estimación híbrida en el resto de los casos, en los que el ME\_GPS estará entre ME\_GPS<sub>MIN</sub> y ME\_GPS<sub>MAX</sub>, queda definida la técnica propuesta denominada **ponderación**.

### 3.1.1 *Ponderación modificada.*

La técnica anterior se basa únicamente en el error medio obtenido en el posicionamiento GPS en un escenario concreto y no tiene en cuenta el error producido por la localización obtenida con las medidas RSS. Para poder tener en cuenta la precisión del posicionamiento obtenido con RSS se podría incluir alguna medida de calidad en la obtención del parámetro de ponderación. De esta manera se consigue que el parámetro tenga en cuenta la relación de calidad entre el posicionamiento GPS y RSS, obteniendo así una estimación híbrida que se base más en una estimación u otra según lo buena que es una respecto la otra, que en un principio era el objetivo de la combinación híbrida.



**Figura 18.** Relaciones entre  $\beta$  y  $\mu$ .

Igual que para el caso anterior, al no disponer de los errores cometidos en cada una de las posiciones instantáneas, la medida de calidad del posicionamiento RSS se basará en su error medio (ME\_RSS) cometido en los distintos escenarios de localización que se utilice. Una vez se dispone de las medidas de calidad tanto de GPS como RSS hay que encontrar una medida que relacione ambas calidades a partir de la cual si ambas técnicas son igual de precisas se obtenga una  $\mu = 0.5$  (se toma la media), si GPS es más preciso que RSS  $\mu > 0.5$  y si RSS lo es más  $\mu < 0.5$ . Esta medida de relación entre las calidades de ambas técnicas puede ser la siguiente:

$$\beta = \frac{ME\_RSS}{ME\_GPS} \quad (14)$$

De tal manera que se obtienen las relaciones anteriormente comentadas entre  $\mu$  y  $\beta$  mostradas en la **Figura 18** a partir de las cuales se obtiene la siguiente expresión entre  $\mu$  y  $\beta$ :

$$\mu = \frac{\beta}{2} = \frac{ME\_RSS}{2 \cdot ME\_GPS} \quad (15)$$

Igual que en la técnica anterior el parámetro  $\mu$  debe ser menor que 1, en este caso según la expresión (15) el máximo del parámetro ( $\mu = 1$ ) se obtiene cuando el error RSS es el doble que el error GPS, de tal manera que fija la máxima diferencia entre calidades en el doble, teniendo que acotar el valor de  $\beta$  en 2 de tal manera que para valores mayores conduzcan al valor de  $\mu$  máximo.

Con la expresión (12), (14) y (15) queda definida la técnica denominada **ponderación modificada** propuesta en el presente PFC para la obtención de estimaciones híbridas. Cabe destacar que todo y modificar la obtención del parámetro de ponderación utilizado en la técnica anterior con el objetivo de tomar en cuenta también la precisión del posicionamiento RSS, aún quedan ciertas consideraciones como la definición de  $\beta$  y su valor máximo que hacen que esta técnica se base más en la precisión GPS que la RSS, por eso a continuación se propone una técnica basada en un modelo matemático a partir del cual se obtienen unos parámetros que proporcionan el estimador óptimo para ese modelo matemático.

## 3.2 BLUHE, *Best Linear Unbiased Hybrid Estimator*.

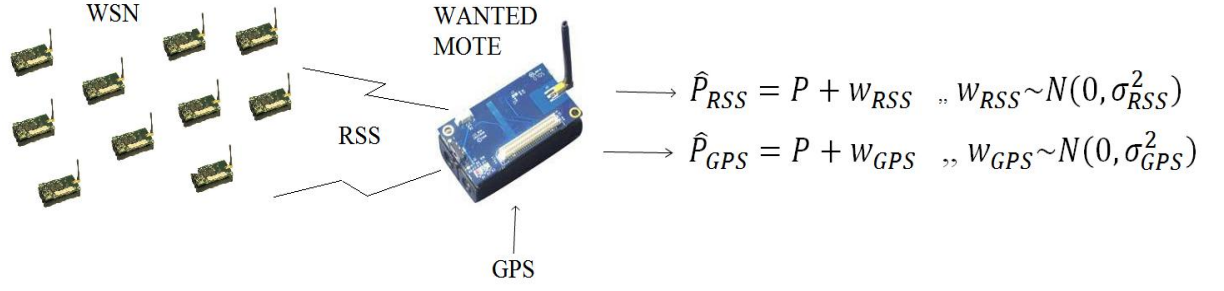
Las técnicas de ponderación anteriormente propuestas surgen de la idea de tomar la media de una variable aleatoria con el objetivo de obtener un estimador insesgado y de menor varianza que la propia variable aleatoria. A partir de esta idea se propone un método que en vez de tomar la media de las estimaciones RSS y GPS toma una ponderación que dependerá de lo preciso que sea GPS, en el caso de la ponderación simple esta dependencia se basa en el error medio GPS y en el caso de la ponderación modificada esta dependencia se basa en la precisión relativa de GPS respecto RSS. En cierto modo estas técnicas están condicionadas por el error GPS, en la ponderación simple al depender básicamente del error GPS queda claro, sin embargo en la ponderación modificada no resulta tan obvia esta dependencia pero queda presente en el momento que se debe fijar la  $\mu$  máxima cuando el error RSS es el doble que el de GPS, mientras que en el caso contrario (cuando el error GPS es el doble del error RSS) no se fija la  $\mu$  mínima, que sería el parámetro con el que sólo se tendría en cuenta el posicionamiento RSS.

Dada la dependencia de las técnicas de ponderación con el error GPS se plantea una técnica que permita la combinación de las posiciones obtenidas con GPS y RSS para obtener una estimación de la posición más precisa que las anteriores, pero valorando de la misma forma la relación entre la precisión GPS y RSS, es decir, sin estar condicionada por ninguna de las dos precisiones. Para ello, en primer lugar se definirá el modelo de señal o planteamiento matemático del problema híbrido, a continuación se buscará el estimador insesgado que realiza la mejor combinación lineal en el sentido de minimizar el error, y por último se comentan las prestaciones del estimador según la correspondencia entre el modelo de señal presentado y el sistema de posicionamiento híbrido a implementar.

### 3.2.1 *Modelo de señal utilizado.*

Para definir el modelo de señal que permita obtener el estimador comentado se parte de dos estimaciones erróneas modeladas como el valor real más un ruido AWGN (*Additive White Gaussian Noise*), es decir, de media nula y varianza  $\sigma_{RSS}^2$  y  $\sigma_{GPS}^2$  respectivamente, en el caso del presente PFC estas estimaciones se obtienen a partir de las medidas RSS y GPS extraídas de la red de sensores inalámbricos, tal y como se muestra en la **Figura 19**, donde  $P$  es la posición real de la mota desconocida y  $w_{RSS}$  y  $w_{GPS}$  son el ruido AWGN de cada una de las estimaciones, que hace que éstas no proporcionen el valor correcto de la posición. En este escenario se puede definir un vector de datos que contenga ambas estimaciones como sigue:

$$\mathbf{x} = \begin{bmatrix} \hat{P}_{RSS} \\ \hat{P}_{GPS} \end{bmatrix} = P \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} w_{RSS} \\ w_{GPS} \end{bmatrix} = P \cdot \mathbf{s} + \mathbf{w} \quad (16)$$



**Figura 19.** Escenario híbrido.

De esta manera, la combinación lineal que proporciona la estimación híbrida en (11) queda como sigue, siendo  $\mathbf{a} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$  el vector que contiene los factores de la combinación lineal:

$$\hat{P}_H = \alpha \cdot \hat{P}_{RSS} + \beta \cdot \hat{P}_{GPS} = [\alpha \quad \beta] \cdot \begin{bmatrix} \hat{P}_{RSS} \\ \hat{P}_{GPS} \end{bmatrix} = \mathbf{a}^T \cdot \mathbf{x} \quad (17)$$

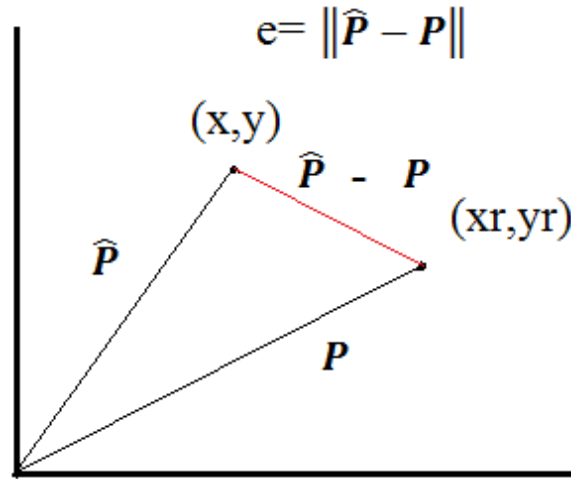
Cabe destacar que se representan las posiciones, tanto las estimadas como la real, como un escalar pero las posiciones son un conjunto de coordenadas con lo que sería más conveniente representarlas como un vector, no obstante para facilitar el desarrollo matemático es más útil representar el conjunto de coordenadas que representan las posiciones como un escalar y pensar que cualquier factor afecta por igual a ambas coordenadas de la posición.

Una vez representado el modelo utilizado para la estimación híbrida se procede a obtener los parámetros, representados por  $\mathbf{a}$ , que permiten obtener el mejor estimador lineal e insesgado en el sentido que se obtiene un estimador de varianza mínima (BLUE, *Best Linear Unbiased Estimator*), respecto el resto de estimadores lineales, de aquí el nombre BLUHE (estimador BLUE híbrido). En el caso del sistema híbrido del PFC se pretende incrementar la precisión o equivalentemente reducir el error producido en el posicionamiento, que de la **Figura 20** se obtiene la siguiente expresión:

$$e = \sqrt{(x - x_r)^2 + (y - y_r)^2} = \|\hat{\mathbf{P}} - \mathbf{P}\| \quad (18)$$

Comparando esta expresión con la expresión de la varianza de las estimaciones, se obtiene la relación (19) entre el error de posicionamiento y su varianza, donde se observa que la varianza es proporcional al error con lo que minimizar la varianza del estimador híbrido es equivalente a minimizar el error en el posicionamiento:

$$\begin{aligned} var(\hat{\mathbf{P}}) &= E[(\hat{\mathbf{P}} - E[\hat{\mathbf{P}}])^2] = E[(\hat{\mathbf{P}} - \mathbf{P})^2] = E[(\hat{\mathbf{P}} - \mathbf{P}) \cdot (\hat{\mathbf{P}} - \mathbf{P})^T] = \\ &= E[\|(\hat{\mathbf{P}} - \mathbf{P})\|^2] \\ var(\hat{\mathbf{P}}) &= E[e^2] \end{aligned} \quad (19)$$



**Figura 20.** Significado geométrico del error cometido en el posicionamiento.

### 3.2.2 Determinación del BLUE híbrido (BLUHE).

Para determinar el BLUHE se restringe el estimador híbrido a un estimador lineal e insesgado y se buscará el vector de parámetros  $\mathbf{a}$  que minimiza la varianza del estimador lineal e insesgado asumido. La condición de estimador lineal ya se ha impuesto desde el inicio al definir el estimador híbrido a partir de la expresión (11) restringiéndolo al estimador lineal de la expresión (17), a partir de la cual se obtiene la siguiente restricción de estimador insesgado, teniendo en cuenta que la esperanza de  $\mathbf{a}$  sigue siendo  $\mathbf{a}$  ya que no es aleatorio:

$$E(\hat{P}_H) = \mathbf{a}^T \cdot E(\mathbf{x}) = P \quad (20)$$

Siendo su varianza:

$$var(\hat{P}_H) = E \left[ (\mathbf{a}^T \cdot \mathbf{x} - E(\mathbf{a}^T \cdot \mathbf{x}))^2 \right]$$

Que teniendo en cuenta que  $E(\mathbf{a}^T \cdot \mathbf{x}) = \mathbf{a}^T \cdot E(\mathbf{x})$ , tras sacar factor común de  $\mathbf{a}^T$  y desarrollando el cuadrado como  $r^2 = r \cdot r^T$  se obtiene [34]:

$$var(\hat{P}_H) = E \left[ (\mathbf{a}^T \cdot \mathbf{x} - \mathbf{a}^T \cdot E(\mathbf{x}))^2 \right] = \mathbf{a}^T \cdot \mathbf{C} \cdot \mathbf{a} \quad (21)$$

Siendo  $\mathbf{C}$  la matriz de covarianza de  $\mathbf{x}$ :

$$\mathbf{C} = E[(\mathbf{x} - E[\mathbf{x}]) \cdot (\mathbf{x} - E[\mathbf{x}])^T] \quad (22)$$

Ahora para encontrar el BLUE se debe minimizar la expresión en (21) sujeta a la restricción de insesgado, que de (20) y la definición de  $\mathbf{x}$  en (16) resulta en:

$$\mathbf{a}^T \cdot E(\mathbf{x}) = P$$

$$\mathbf{a}^T \cdot \mathbf{s} \cdot P = P$$

$$\mathbf{a}^T \cdot \mathbf{s} = 1$$

Donde, en el caso que nos ocupa,  $\mathbf{s} = [1 \ 1]^T$ . La solución de la minimización de la varianza en (21) sujeta a la restricción de estimador insesgado anterior se obtiene a partir de la minimización de la función de coste obtenida al utilizar multiplicadores de Lagrange:

$$J(\mathbf{a}) = \mathbf{a}^T \cdot \mathbf{C} \cdot \mathbf{a} + \lambda \cdot (\mathbf{a}^T \cdot \mathbf{s} - 1)$$

Obteniéndose la siguiente solución, que proporciona los parámetros  $\mathbf{a}$  que hacen que el estimador sea insesgado y tenga la mínima varianza posible [35]:

$$\mathbf{a}_{BLUHE} = \frac{\mathbf{C}^{-1} \cdot \mathbf{s}}{\mathbf{s}^T \cdot \mathbf{C}^{-1} \cdot \mathbf{s}} \quad (23)$$

Con esta expresión sólo falta por obtener la matriz de covarianza  $\mathbf{C}$  a partir de (22):

$$\begin{aligned} \mathbf{C} &= E[(\mathbf{x} - E[\mathbf{x}]) \cdot (\mathbf{x} - E[\mathbf{x}])^T] = \\ &= E[(P \cdot \mathbf{s} + \mathbf{w} - P \cdot \mathbf{s}) \cdot (P \cdot \mathbf{s} + \mathbf{w} - P \cdot \mathbf{s})^T] = \\ &= E[\mathbf{w} \cdot \mathbf{w}^T] \end{aligned}$$

Que teniendo en cuenta la definición de  $\mathbf{w}$  en (16) y la independencia entre el ruido GPS y RSS se obtiene la siguiente matriz de covarianza:

$$\mathbf{C} = \begin{bmatrix} \sigma_{RSS}^2 & 0 \\ 0 & \sigma_{GPS}^2 \end{bmatrix} \text{ y } \mathbf{C}^{-1} = \begin{bmatrix} \frac{1}{\sigma_{RSS}^2} & 0 \\ 0 & \frac{1}{\sigma_{GPS}^2} \end{bmatrix}$$

Con esta matriz  $\mathbf{C}$  y el valor de  $\mathbf{s}$  se obtienen los siguientes resultados:

$$\begin{aligned} \mathbf{C}^{-1} \cdot \mathbf{s} &= \begin{bmatrix} \frac{1}{\sigma_{RSS}^2} \\ \frac{1}{\sigma_{GPS}^2} \end{bmatrix} \\ \mathbf{s}^T \cdot \mathbf{C}^{-1} \cdot \mathbf{s} &= \left( \frac{1}{\sigma_{RSS}^2} + \frac{1}{\sigma_{GPS}^2} \right) = \frac{\sigma_{RSS}^2 + \sigma_{GPS}^2}{\sigma_{RSS}^2 \cdot \sigma_{GPS}^2} \end{aligned}$$

Que permiten obtener los parámetros,  $\mathbf{a}$ , óptimos:

$$\mathbf{a}_{BLUHE} = \frac{\mathbf{C}^{-1} \cdot \mathbf{s}}{\mathbf{s}^T \cdot \mathbf{C}^{-1} \cdot \mathbf{s}} = \frac{\sigma_{RSS}^2 \cdot \sigma_{GPS}^2}{\sigma_{RSS}^2 + \sigma_{GPS}^2} \cdot \begin{bmatrix} \frac{1}{\sigma_{RSS}^2} \\ \frac{1}{\sigma_{GPS}^2} \end{bmatrix} \quad (24)$$

Produciendo el siguiente estimador híbrido:

$$\hat{P}_H = \mathbf{a}_{BLUHE}^T \cdot \mathbf{x} = \frac{\sigma_{RSS}^2 \cdot \sigma_{GPS}^2}{\sigma_{RSS}^2 + \sigma_{GPS}^2} \cdot \left( \frac{\hat{P}_{RSS}}{\sigma_{RSS}^2} + \frac{\hat{P}_{GPS}}{\sigma_{GPS}^2} \right) \quad (25)$$

De (25) se observa que para el caso de obtener medidas GPS y RSS con la misma precisión el estimador toma la media, mientras que si una de las dos medidas es mucho más errónea que la otra el estimador híbrido toma la medida más precisa, en los casos intermedios el estimador tomará una combinación lineal de las estimaciones que dependerá de la diferencia entre precisiones, tal y como se había propuesto al principio, sin ningún tipo de condicionamiento por alguna de las estimaciones. En principio estos parámetros deberían calcularse para cada una de las posiciones a estimar, según la precisión de cada una de las técnicas (RSS y GPS) en ese mismo instante, pero igual que para el caso de las técnicas de ponderación, no se dispone de estas precisiones instantáneas con lo que se utilizan unos parámetros fijos calculados a partir de los errores medios de ambas estimaciones en el escenario de localización en concreto. Realmente los parámetros quedan definidos en (25) por la varianza del ruido en las estimaciones RSS y GPS, sin embargo en el PFC no se dispone de estas varianzas con lo que se utilizan los errores medios obtenidos en los distintos escenarios, que ya se ha visto en (19) que está relacionado con la varianza de la estimación o equivalentemente la varianza del ruido que afecta a la estimación y no permite obtener el valor real. Además, el ruido que influye a las estimaciones de posición RSS y GPS no tiene por qué ser AWGN, con lo que la correspondencia entre el modelo de señal propuesto y el escenario de localización real no será del todo correcta y por lo tanto los parámetros obtenidos no se corresponderán con los óptimos presentados en (24) y la estimación híbrida no será la estimación lineal insesgada de menor varianza.







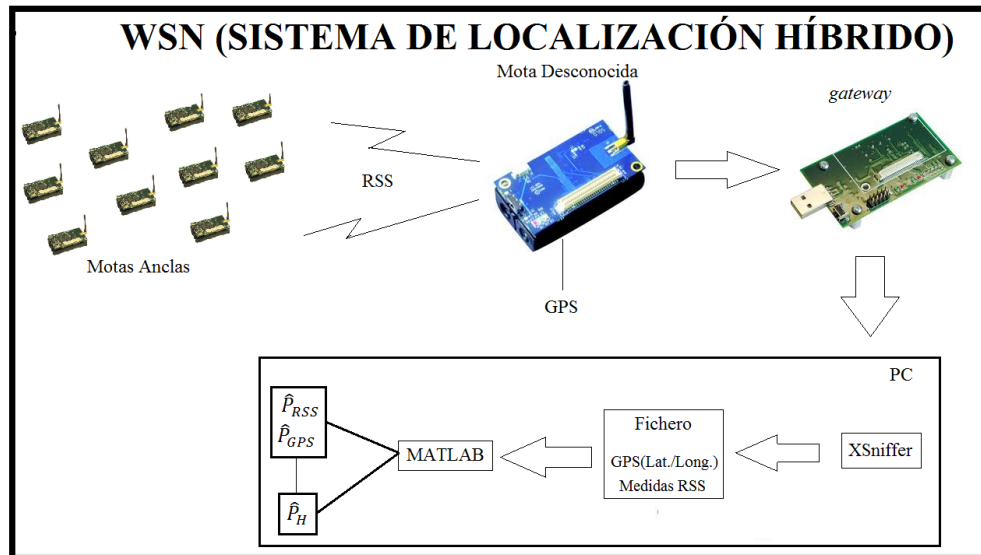
## 4 Aspectos de implementación del sistema.

Una vez mostradas las ideas y conceptos para llevar a cabo el sistema de posicionamiento híbrido tratado hay ciertos aspectos a tener en cuenta, relacionados con estas ideas y conceptos, para la implementación del sistema de posicionamiento híbrido. Básicamente estos aspectos están relacionados con la programación o *software* necesario para dar el funcionamiento deseado a la red de sensores inalámbricos utilizada para desarrollar el sistema, entre estos aspectos se encuentran:

- **Programación de las motas de la WSN** para poder tomar medidas RSS y los paquetes GPS que permitan obtener las posiciones necesarias para realizar la estimación híbrida.
- **Programación del *gateway*** que permita la adquisición de los datos que provienen de la WSN para pasarlos a la estación base.
- **Programación de la estación base** que permita procesar los datos que envíe el *gateway* y obtener las funcionalidades del sistema deseadas.

La programación de las motas como del *gateway* se realiza mediante el lenguaje de programación NesC gobernado por el sistema operativo TinyOS, mientras que en la estación base, en el caso del PFC un PC, se programan las aplicaciones en MATLAB. De esta manera, dentro de la WSN se tendrán motas estáticas cuya posición será conocida (**motas anclas**) y transmitirán mensajes sin ningún tipo de información para que el resto de motas (**motas desconocidas**), que serán motas dinámicas cuya posición se quiere estimar, puedan obtener sus RSS y con la ayuda de una *sensor-board* sea capaz de extraer su localización mediante GPS. Las motas desconocidas transmitirán la información de las RSS medidas y del posicionamiento GPS obtenido por la WSN de tal manera que el *gateway* recolectará los datos obtenidos y los pasará al ordenador donde con la ayuda de un programa denominado **XSniffer** se crearán unos archivos con la información transmitida. Estos archivos se podrán procesar con MATLAB para obtener el posicionamiento a partir de las RSS medidas y obtener la translación necesaria a las coordenadas GPS para poder combinarlas con las posiciones RSS y obtener las estimaciones híbridas estudiadas. Todo este proceso que sigue el sistema de posicionamiento híbrido para obtener la localización de las motas desconocidas se recoge en la **Figura 21**.

Para el desarrollo del software necesario para formar la WSN así como los programas de apoyo necesarios para procesar la información con un ordenador se utiliza la plataforma de desarrollo *Mote Works* proporcionada por Crossbow. Esta plataforma dispone del sistema operativo TinyOS así como aplicaciones que permiten el funcionamiento de las diferentes *sensor-boards* del fabricante. Además, esta plataforma proporciona el *software* XSniffer utilizado para la visualización y generación de ficheros con los datos recogidos por el *gateway*.



**Figura 21.** Diagrama del sistema híbrido implementado.

En el presente capítulo se detallan los aspectos de las distintas fases de programación que permiten la creación del sistema de posicionamiento híbrido a partir de los conceptos explicados en los capítulos anteriores. En primer lugar se trata la programación de las motas que componen la WSN, distinguiendo la funcionalidad de las motas anclas y las motas desconocidas. En segundo lugar se trata la programación del *gateway* y el *software* XSniffer. A continuación, se comentan los códigos de MATLAB necesarios para obtener el posicionamiento RSS, la translación GPS para la posible combinación y los parámetros utilizados para la obtención de estos resultados y del posicionamiento híbrido. Por último, tras repasar el *software* necesario para la implementación del sistema se verá necesaria una fase de entrenamiento para obtener ciertos parámetros necesarios para la puesta en marcha del sistema y que permitan el correcto funcionamiento del sistema en el escenario de localización en el que se utiliza.

Todo el código necesario tanto para la programación de las motas de la WSN como para la programación de MATLAB en el PC se incluye en el **ANEXO 1** y **ANEXO 2** donde se adjuntan respectivamente los programas en NesC para programar las motas y los programas en MATLAB para generar el sistema de localización híbrido.

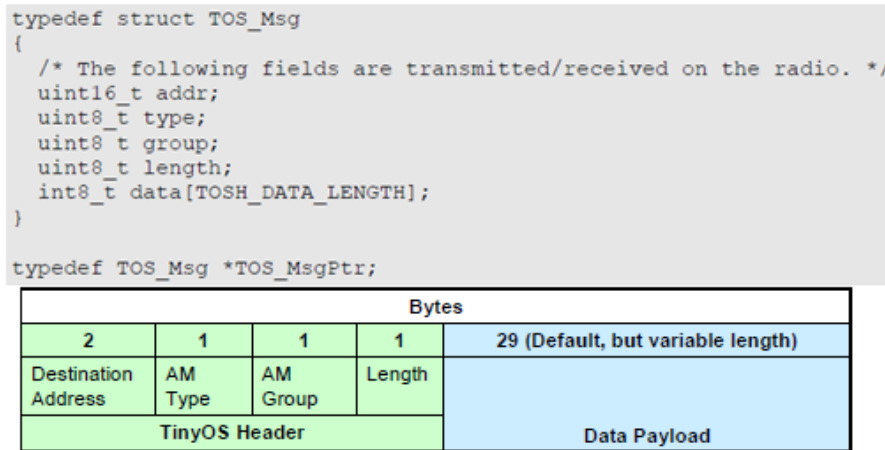
## 4.1 Programación de las motas de la WSN.

Se ha comentado que la red de sensores utilizada en el PFC está compuesta por dos tipos de motas:

- **Motas anclas:** Son motas cuya posición es conocida y únicamente transmiten mensajes en blanco para que las motas a localizar puedan medir sus potencias recibidas. Para poder identificar las distintas potencias recibidas las motas anclas transmitirán su identificador (*node id*).
- **Motas desconocidas:** Son motas cuya posición es desconocida y se pretende localizar a partir de la combinación del posicionamiento con las medidas RSS y la localización GPS proporcionada por alguna *sensor-board* que lo permita. De esta manera, estas motas deben tener la capacidad de:
  - Recibir los mensajes enviados por las motas anclas para medir las RSS.
  - Controlar la *sensor-board* para recibir datos del GPS y obtener la localización.
  - Enviar al *gateway* mensajes con la información de las RSS medidas junto los identificadores de los que se ha medido y la información obtenida del GPS.

Para la transmisión y recepción de mensajes, TinyOS define una estructura denominada *TOS\_Msg* que permite la creación de mensajes a partir de la cabecera típica de los mensajes TinyOS y los campos de datos introducidos por el usuario (**Figura 22**). Los campos del mensaje se definen igual que las variables en NesC, se indica el tipo de variable (int: entero, uint: entero sin signo) y a continuación la cantidad de bits reservados en memoria para la variable. Todos los mensajes de TinyOS contienen la misma cabecera, representada en la **Figura 22**, cuyos campos son:

- **addr:** Dirección de destino, para el caso del PFC tanto las motas anclas como las desconocidas tienen como dirección de destino la dirección de *broadcast*.
- **type:** Tipo del mensaje activo, en el caso que nos ocupa serán mensajes normales de información (*type* = 0), pero pueden haber mensaje del tipo *Ack*, sincronismo,...
- **group:** identificador de grupo que permite clasificar motas con los mismos *node id* en distintos grupos con el objetivo de ampliar la capacidad de identificar distintos nodos de la red.
- **length:** longitud de la información útil (*Data Payload*) del mensaje.
- **data:** Campo dedicado a la información útil del mensaje.



**Figura 22.** Estructura de empaquetamiento de mensaje TinyOS.

A continuación se explican los códigos utilizados para proporcionar la funcionalidad necesaria a las motas que constituyen la WSN del sistema híbrido de localización.

#### 4.1.1 Motas anclas.

El objetivo de las motas anclas es el de transmitir mensajes en el rango de cobertura de la WSN para que las motas desconocidas puedan medir la potencia recibida. Con estas potencias se podrá estimar la distancia a cada uno de los nodos con los que se tiene cobertura, a partir del modelo de propagación definido por los parámetros estimados experimentalmente (parametrización del escenario), y con las posiciones conocidas de las motas anclas se podrán crear los círculos que permitan el posicionamiento a partir de las medidas RSS. Por lo tanto, la aplicación NesC para las motas anclas simplemente debe incluir la configuración necesaria para la transmisión de mensajes en *broadcast* (hacia todos) vía radio. Además su módulo debe incluir los eventos y tareas necesarias para el funcionamiento de la mota y la radio así como rellenar la estructura del mensaje transmitido, que todo y no incluir información relacionada con los sensores se incluye un campo que indica el identificador del nodo transmisor.

En el **Capítulo 2** se ha comentado la programación en NesC y se ha explicado que las aplicaciones en este lenguaje de programación están formadas por dos archivos denominados configuración (unión de los componentes utilizados) y módulo (implementación de eventos y tareas utilizadas). Para las aplicaciones que necesiten la transmisión de mensajes es necesario otro archivo que incluya la estructura de empaquetamiento de mensaje de TinyOS (TOS\_Msg). Para ver esta estructura así como la configuración y módulo de la aplicación de las motas anclas se puede consultar el **ANEXO 1**, donde se adjuntan todos los códigos NesC utilizados.

### 4.1.2 *Motas desconocidas.*

Las motas desconocidas serán las que lleven incorporadas la *sensor-board* MTS420/400CC que permite obtener la localización GPS. Además del código necesario para hacer funcionar el GPS se necesita el código para recibir los mensajes que envían las motas anclas y medir la potencia recibida para posteriormente enviar por radio tanto los datos GPS como la potencia medida para que llegue a la mota base situada en el *gateway*.

Para la medida de la potencia recibida basta con incluir los componentes necesarios para la recepción de los mensajes enviados por las motas anclas y que permitan identificar la recepción de un mensaje. Además la radio disponible en las motas Iris utilizadas disponen de un registro con una medida relacionada con la potencia recibida, concretamente se dispone de la energía detectada. Así utilizando la función que permite leer del registro en el evento que identifique la recepción del mensaje se puede obtener la potencia recibida del nodo que ha enviado el mensaje.

Para el control del GPS se parte de un código proporcionado por Crossbow que hace funcionar la *sensor-board* utilizada. La aplicación proporcionada es la denominada *TestMTS420cc* que define los *drivers* necesarios para utilizar los distintos sensores y el módulo GPS de la placa. Para el caso del sistema a implementar las motas no necesitan la capacidad de leer los sensores disponibles en la placa, únicamente se necesita leer los datos que proporcione el módulo GPS, así que se modifica la aplicación *TestMTS420cc* para que únicamente utilice el módulo GPS de la *sensor-board*. Esta modificación lo único que hace es comentar la parte del código que activa o inicializa los componentes necesarios para el control de los sensores disponibles de tal manera que la mota únicamente activa y utiliza el módulo GPS. El resto del código se mantiene para ahorrar tiempo en el análisis de las relaciones entre el código específico de GPS y el resto.

La principal funcionalidad del código es que permite la obtención, entre otros valores, de la localización del GPS en coordenadas esféricas (Latitud, longitud y altitud) a partir de los mensajes transmitidos por el GPS que siguen el formato de los mensajes GGA (*Global Positioning System Fix Data*) (**Figura 23**) del protocolo NMEA [36]. En la **Figura 23** se muestra un ejemplo de mensaje GGA transmitido por el GPS hacia la mota, donde en la aplicación comentada se guardan los valores más importantes en variables para posteriormente transmitirlos en un mensaje. Los principales campos de estos mensajes son los siguientes:

- Tiempo en el que se toma la medida.
- Latitud/Longitud en grados y minutos, altitud en metros.
- Dirección de Latitud y Longitud, es decir, si apuntan hacia el Norte/Sud o al Este/Oeste respectivamente.
- Número de satélites a los que se está conectado en el momento de la medida.

```

/*****
* Packet received from GPS - ASCII msg
* 1st byte in pkt is number of ascii bytes
* async used only for testing
GGA - Global Positioning System Fix Data
GGA,123519,4807.038,N,01131.324,E,1,08,0.9,545.4,M,46.9,M, , *42
123519      Fix taken at 12:35:19 UTC
4807.038,N  Latitude 48 deg 07.038' N
01131.324,E Longitude 11 deg 31.324' E
1           Fix quality: 0 = invalid
                        1 = GPS fix
                        2 = DGPS fix
08          Number of satellites being tracked
0.9         Horizontal dilution of position
545.4,M     Altitude, Metres, above mean sea level
46.9,M     Height of geoid (mean sea level) above WGS84
            ellipsoid
(empty field) time in seconds since last DGPS update
(empty field) DGPS station ID number
*****/

```

**Figura 23.** Formato de los mensajes GGA.

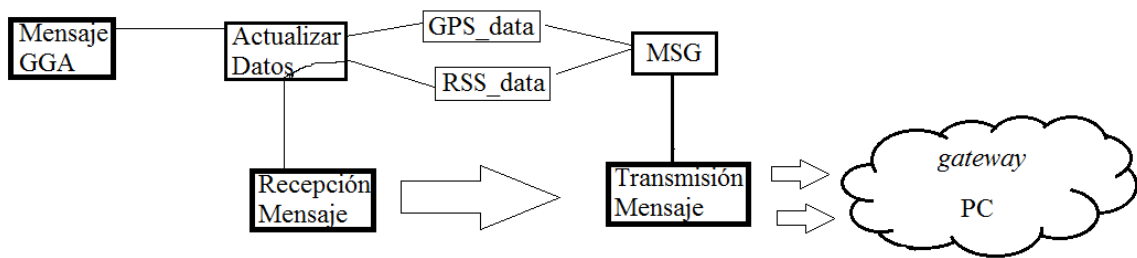
La aplicación modificada del *TestMTS420cc* lee los mensajes GGA del GPS y guarda sus valores en la estructura del mensaje definido, que una vez completada se transmitirá para que la mota base conectada al *gateway* pueda pasar los datos al ordenador utilizado como estación base. Incluyendo a esta modificación la capacidad de recibir los mensajes de las motas anclas y leer sus potencias, anteriormente comentada, se puede incluir dos campos más, que representen la RSS medida y el identificador del nodo que se ha medido la potencia, a la estructura definida por la modificación de *TestMTS420cc*. De esta manera la estructura del mensaje transmitido por las motas desconocidas es la mostrada en la **Figura 24**, que incluye los campos principales del mensaje GGA, pero hay algún campo que merece algo más de explicación como:

- **Lat/Long\_dec\_min:** Latitud/Longitud en minutos. Este valor está multiplicado por  $10^5$  para evitar decimales en el valor transmitido.
- **nsewid:** Identificador de la dirección de la latitud y longitud (N/S o E/W).

El código utilizado para las motas desconocidas del PFC va actualizando los campos del mensaje GGA a medida que van llegando datos del GPS, y cada vez que se recibe un mensaje de una mota ancla se mide la RSS correspondiente a esa mota, se actualizan los campos correspondientes en el mensaje (rss y Txnodeid) y se envía el mensaje completo para que la mota base pase la información al ordenador para que procese los datos con el objetivo de extraer el posicionamiento RSS y un posicionamiento GPS local que permita la estimación de la localización del sistema híbrido. (**Figura 25**)

BYTES										
1	1	1	1	4	4	4	1	1	1	1
Hora	Min.	Lat. (°)	Long. (°)	Seg_dec	Lat_dec (min.)	Long_dec (min.)	Nsewid	Fix	Tx_nodeid	rss
GGAMsg										

**Figura 24.** Información útil (mensaje GGA) de las motas desconocidas.



**Figura 25.** Esquema de la aplicación programada en las motas desconocidas.

## 4.2 Programación del *gateway*.

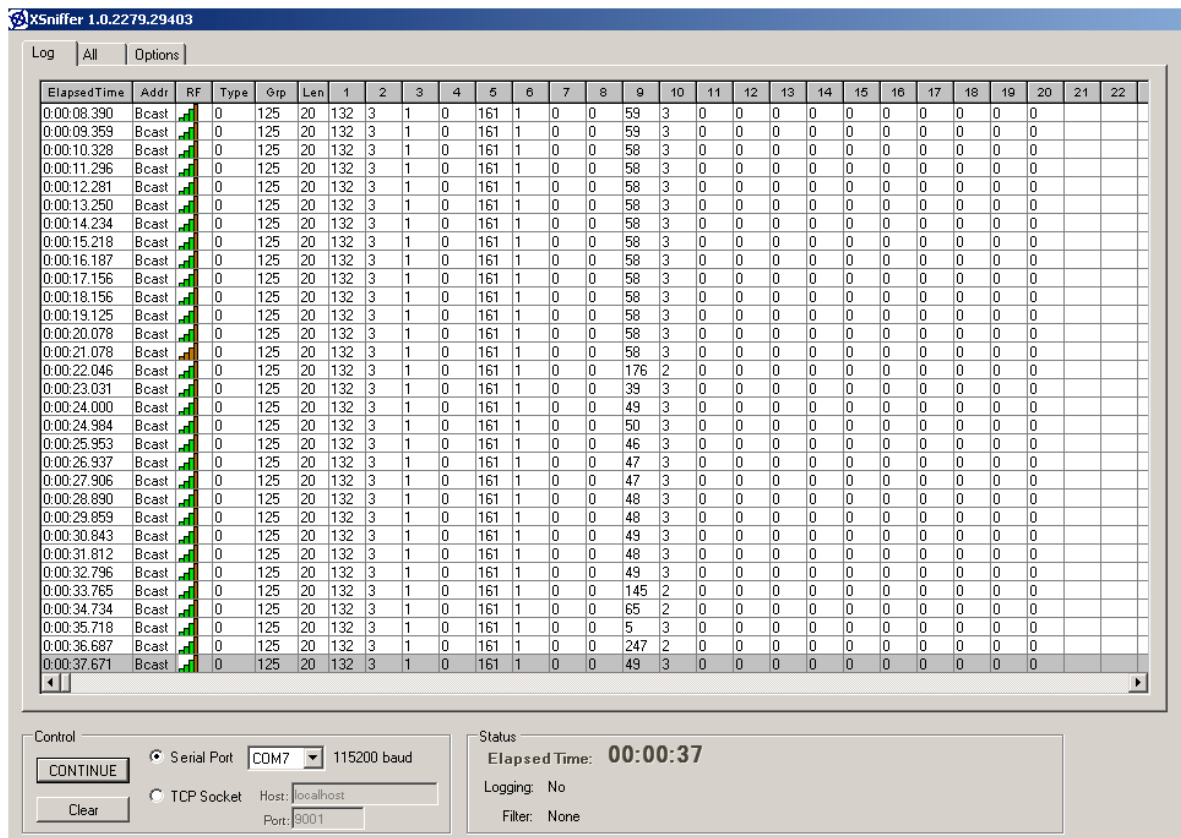
El *gateway* es el encargado de pasar la información recogida de la WSN a la estación base. En el caso del presente proyecto ya se ha comentado que para ello se utiliza una mota, correctamente programada, conectada a la placa de interfaz de mota o placa de programación MIB520. De esta manera, la mota base recibe los datos transmitidos por las motas anclas y las motas desconocidas y los pasa a la estación base a través de la conexión USB de la MIB. Dentro de la plataforma Mote Works se encuentra una aplicación que permite proporcionar esta funcionalidad a la mota base que funcionará como *gateway*, esta aplicación es la denominada *TOSBase.nc* y permite utilizar una mota conectada a la placa de programación como puente entre la WSN y un ordenador utilizado como estación base.

Los datos que provienen de la WSN son recibidos por la mota base y pasados por el puerto serie del ordenador en formato binario y con el orden dado por la estructura del mensaje. Este formato de los datos no es muy útil para trabajar con la información que transporta y mucho menos para ser visualizados por un humano que quiera monitorizar la información. Para ello se utiliza XSniffer, que es un *software* que permite:

- La lectura de los datos provenientes del puerto serie al que está conectado el *gateway*.
- Visualización de la información formateada, según los campos de la estructura del mensaje, en *little-endian*, que además de monitorizar la información permite debugar las aplicaciones instaladas en las motas.
- Almacenamiento de la información en archivos de valores separados por coma (.csv).

Al ejecutar el programa XSniffer en el PC se abre la ventana que permite seleccionar el puerto serie al que está conectado el *gateway* y empezar a presentar la información. La información se representa en filas cuyas columnas son los bytes del mensaje presentados en decimal (0-255) y con los campos formateados en *little-endian*, es decir, los bytes de cada campo se representan del menos significativo al más significativo. Por ejemplo, el valor hexadecimal 0x4A3B2C1D se representaría como {1D, 2C, 3B, 4A}.



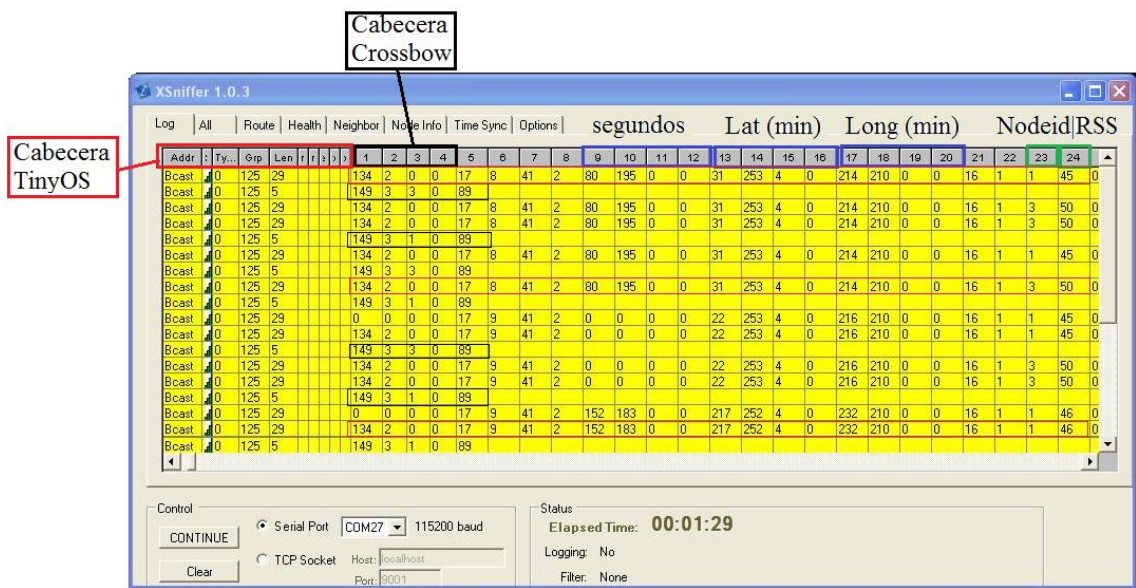


**Figura 26.** Captura de la ventana de XSniffer.

En la **Figura 26** se muestra una captura de XSniffer donde el puerto serie a leer es el COM7 y se muestran medidas de temperatura y luz en las columnas 5-6 y 9-10 respectivamente. En las medidas de luz se puede apreciar que al tapar el sensor de luz de la placa de sensores el valor mostrado disminuye pasando de un valor 58-3 a un valor 176-2, con lo que queda presente el formateo de los campos en *little-endian* con el hecho de que al disminuir el valor de la medida se reduce el byte situado más a la derecha (más significativo) y aumenta el más situado a la izquierda (menos significativo). Por último, en la pestaña “options” hay una opción que al activarla almacena la información recogida en un archivo que podrá ser consultado mediante Excel o un analizador de texto, además MATLAB tiene la posibilidad de trabajar con estos archivos.

Para el caso del presente proyecto se trabaja en el puerto serie COM27 y se muestran los datos de la estructura del mensaje de las motas anclas y desconocidas. Entre las opciones disponibles de XSniffer hay una que permite filtrar los paquetes según la mota de la que proviene, con lo que se podrían filtrar todos los paquetes provenientes de las motas anclas y únicamente representar los paquetes de las motas desconocidas, sin embargo esta selección de paquete se realizará posteriormente en MATLAB, con lo que tanto en la ventana de XSniffer como en los archivos generados se encontrarán los mensajes de las motas anclas junto con los mensajes de las motas desconocidas.

Además, no se ha comentado pero dentro de la información útil del mensaje de las motas desconocidas hay una parte que se corresponde con la cabecera introducida por Crossbow que ocupa los 4 primeros bytes de la información útil, esto hay que tenerlo en cuenta, igual que la propia cabecera de los mensajes TinyOS, para identificar correctamente los distintos campos del mensaje. De esta manera empezando por el cuarto byte de la información útil (cuarta columna de XSniffer) hay que considerar el tamaño de cada uno de los campos definido en la estructura del mensaje (**Figura 24**), es decir, todos los campos, excepto los segundos y Latitud/Longitud (min) que ocupan 4 columnas cada uno (32 bits = 4 Bytes), ocupan una columna (1 Byte) tal y como se muestra en la **Figura 27**.



**Figura 27.** Captura de mensajes de la WSN utilizada.

### 4.3 Programación en MATLAB.

Hasta ahora se ha explicado el *software* utilizado para dar capacidad a la WSN de tomar medidas RSS y del GPS para enviarlas a la estación base (PC) donde mediante XSniffer se crean unos archivos con la información recogida por la WSN. Llegados a este punto hay que tratar los archivos obtenidos para obtener las funcionalidades principales del sistema, que son básicamente el posicionamiento mediante las medidas RSS y el posicionamiento GPS referenciado al posicionamiento RSS para permitir el posicionamiento híbrido comentado en el capítulo anterior.

Para el tratamiento de los archivos generados se utilizará MATLAB, básicamente se debe abrir el fichero desde MATLAB y se genera una variable con los datos disponibles en el archivo. Una vez disponibles los datos a procesar se necesitan otras variables previamente definidas que representen el escenario, como:

- Las posiciones locales de las motas anclas.
- Parámetros de propagación a utilizar en la estimación de distancias a partir de las medidas RSS.
- Posiciones GPS de las motas anclas que permitan establecer una relación entre el sistema de coordenadas GPS y el del sistema de posicionamiento RSS.

Una vez definidas todas estas variables que representan los datos obtenidos por la WSN y las características del escenario de localización, se crean las funciones en MATLAB necesarias para simular y validar el sistema de localización híbrido del presente PFC. Todos los archivos creados se adjuntan en el ANEXO 2, pero las funciones principales son:

- **GPS**: Permite obtener la localización GPS (Lat/Long) a partir de los datos de la mota desconocida.
- **triangulación**: Realiza la localización a partir de las estimaciones de distancias obtenidas con las RSS medidas por la mota a posicionar.
- **filtroMsg**: Selecciona los datos del archivo de XSniffer que provienen de la mota a posicionar para pasarlos a las funciones anteriores.
- **Main**: Permite la comprobación del sistema a partir de archivos recogidos en distintas posiciones.

A continuación se explicará con detalle las funciones anteriormente comentadas, que son las funciones principales que permiten llevar a cabo el sistema de localización híbrido a partir de los archivos creados por XSniffer con los datos recogidos de la WSN.

### 4.3.1 Función *GPS.m*

La función principal de *GPS.m* es la extracción del posicionamiento GPS obtenido en la WSN y representado en los datos del archivo generado por MATLAB. Para ello *GPS.m* trabaja con unos datos que no contiene la cabecera de XSniffer (primera fila) y sólo dispone de los mensajes que provienen de la mota a localizar, con estos datos y basándose en la estructura de mensaje de la **Figura 24** la función selecciona los campos relacionados con el posicionamiento y los procesa para obtener los valores en formato decimal correspondientes. Se ha comentado que los datos de XSniffer se representan en formato *little-endian* con lo que los campos de más de 1 Byte hay que ordenarlos debidamente, para ello primero se debe convertir el valor decimal de cada uno de los Bytes a su valor binario y posteriormente poner los Bytes en orden inverso, tomando el valor decimal de este resultado se obtiene la representación decimal del campo en cuestión. Una vez obtenidos los valores decimales de los campos de más de 1 Byte, se obtiene el valor de la latitud y longitud proporcionado y se escoge su signo a partir del valor del campo *nsewid*, teniendo en cuenta que la latitud es negativa cuando apunta hacia el Sud y la longitud cuando apunta hacia el Oeste.

De esta manera se obtiene el posicionamiento GPS proporcionado por la placa de sensores situada en la mota a localizar. Sin embargo, estas coordenadas pueden ser muy útiles para representar la posición en un mapa del globo Terrestre, pero no es muy adecuado para utilizarse en un posicionamiento local como es el caso del sistema tratado. Por lo tanto, la función *GPS.m*, realiza una transformación de coordenadas esféricas (*Lat/Long*) a coordenadas UTM [37], que sigue siendo un sistema de coordenadas global pero cartesiano con lo que será más fácil relacionarlo con el sistema de coordenadas local del sistema de localización del PFC. Para el cambio de coordenadas se utiliza una función encontrada por Internet [38] que realiza el cambio de coordenadas esféricas a coordenadas cartesianas UTM.

### 4.3.2 Función *triangulacion.m*

La función *triangulacion.m* permite obtener la localización de la mota desconocida a partir de las RSS medidas en la WSN. El procedimiento seguido es el explicado en el **Capítulo 2**, se estiman las distancias a partir de las RSS medidas mediante los parámetros que modelan el escenario de localización y se obtiene la posición mediante la expresión (10) pero utilizando únicamente tres medidas de distancia. Ya se ha comentado que hay casos en los que las estimaciones de distancias no son muy precisas y la inclusión de más medidas que las necesarias en la expresión (10) afecta al posicionamiento, éste es el caso del posicionamiento mediante estimación de distancia con medidas RSS ya que a distancias largas las estimaciones son erróneas, por ello sólo se utilizan las tres RSS mayores (suponiendo que serán de las motas más cercanas) para la estimación de distancias y posterior cálculo de la posición.

Teniendo en cuenta lo anterior, la función espera a disponer 3 RSS de nodos distintos para empezar a calcular posiciones, a partir de este momento si se obtiene una RSS de otro nodo mayor que alguna de las anteriores se cambia por la menor de las tres, y si se reciben nuevas RSS de los nodos con los que se está localizando se actualiza el valor de la potencia correspondiente. En ambos casos se sigue calculando las posiciones con las tres medidas de RSS disponibles.

De este modo, la función *triangulacion.m* un vez dispone de tres medidas RSS proporciona una posición cada vez que se recibe una nueva RSS, sea de un nuevo nodo o no. Además esta función proporciona los valores de distancia (entre las motas anclas y la mota desconocida) estimados a partir de las RSS medidas por la mota a localizar y los parámetros del modelo de propagación disponibles.

### **4.3.3 Archivo principal (Main).**

El *Main* es el archivo que simula y proporciona los resultados que permiten validar el sistema de localización híbrido implementado en el PFC. Para ello, lo primero que hace es definir el escenario de localización a validar mediante las variables necesarias comentadas en la introducción de la programación en MATLAB, concretamente el procedimiento para la definición del escenario ha sido crear una carpeta para guardar todas las variables relacionadas con el escenario de tal manera que al arrancar el archivo *Main* se preguntará por el directorio del escenario y se abrirán directamente las variables almacenadas, con lo que se podrá simular y validar cualquier escenario siguiendo este procedimiento y utilizando el mismo nombre de variables en todos los escenarios. En cuanto a los parámetros de propagación se dispondrá de una tabla de parámetros disponibles obtenidos en distintos escenarios que se irá ampliando a medida que se procesen escenarios nuevos. Para validar el sistema será necesario comprobar el error entre las posiciones estimadas y las posiciones reales, por lo tanto entre estas variables que definen el escenario de localización se encuentran las posiciones reales en las que se han tomado las medidas. Concretamente la manera de validar el sistema ha sido creando archivos de XSniffer en cada una de las posiciones en las que se toman las medidas de tal manera que para cada una de las posiciones se tendrá un archivo XSniffer para procesar con las funciones anteriores.

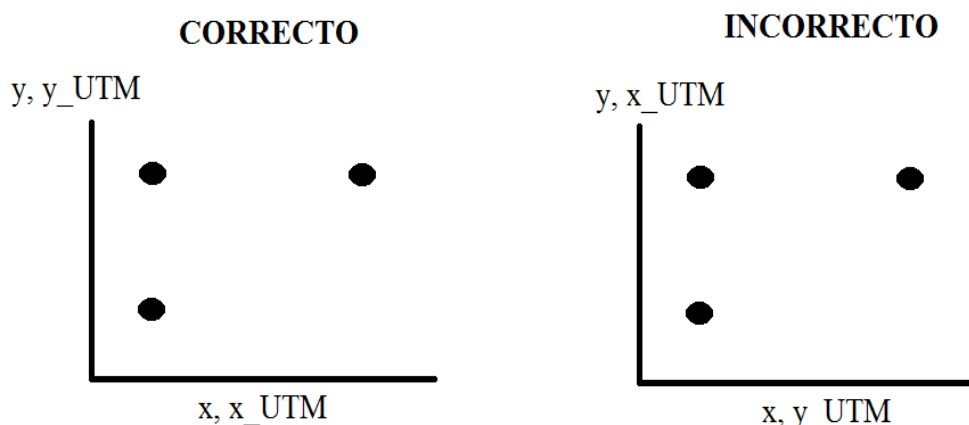
Una vez definidas las variables que representan al escenario y las variables que representan los archivos XSniffer obtenidos en cada una de las posiciones es posible obtener las posiciones estimadas con RSS y las coordenadas UTM calculadas a partir de la estimación GPS. Estas posiciones se obtienen a partir de las funciones *triangulacion.m* y *GPS.m* respectivamente, pero estas funciones trabajan con los mensajes de la mota desconocida con lo que a la hora de llamarlas hay que pasarle únicamente estos mensajes. Para seleccionar los mensajes de la mota desconocida en los archivos XSniffer, que contiene además los mensajes de las motas anclas, se define otra función denominada *filtroMsg.m* que realiza esta selección y llama a las dos funciones anteriores para obtener las posiciones RSS y UTM estimadas, además permite realizar estimaciones RSS con distintos parámetros de propagación.

Así, para obtener las posiciones estimadas el *Main* utiliza la función *filtroMsg.m* para cada una de las posiciones medidas. Al disponer de un archivo XSniffer en cada una de las posiciones, las funciones *triangulacion.m* y *GPS.m* proporcionan tantas posiciones como mensajes recibidos de la mota desconocida, para el caso de la primera las tres primeras posiciones proporcionadas serán 0 representando la falta de las tres medidas mínimas necesarias para calcular la posición. Normalmente la variación de las distintas posiciones proporcionadas no es muy grande, debida básicamente al ruido (error) de cada una de las estimaciones, con lo que en cada una de las posiciones el archivo *Main* toma la media del conjunto de posiciones obtenidas.

Para el caso de las posiciones estimadas con RSS, se ha comentado que la función *filtroMsg.m* permite obtener el posicionamiento RSS para distintos parámetros de propagación, en el caso del *Main* creado para el PFC se utilizan todos los parámetros disponibles en la tabla de parámetros con el objetivo de obtener los parámetros que mejores resultados proporcionen en el escenario de localización concreto, de esta manera para cada una de las posiciones se obtendrán tantas estimaciones RSS como parámetros de propagación disponibles y posteriormente se escogen los parámetros que proporcionen el menor error medio (media de los errores obtenidos en cada una de las posiciones estimadas).

Por otro lado, las posiciones GPS obtenidas siguen siendo coordenadas globales con lo que se trasladan al sistema de coordenadas local referenciándolas a las coordenadas UTM de la mota ancla de referencia, que en el caso que nos ocupa es la mota con coordenadas (0,0) en el sistema de coordenadas local. Para hacer correctamente esta translación y que coincidan las coordenadas del sistema local con las coordenadas referenciadas es necesario que los sistemas de coordenadas local y UTM se correspondan, es decir, que la variación en las coordenadas x, y del sistema local se corresponda con la variación de las coordenadas UTM (**Figura 28**).

En este punto, con las mejores estimaciones de posición RSS (según los parámetros disponibles) y las coordenadas UTM del GPS referenciadas al sistema de coordenadas local, es posible obtener las estimaciones híbridas estudiadas en el capítulo anterior y evaluar las prestaciones comparando las distintas estimaciones con las posiciones reales a las que se han tomado las medidas. Para ello se crea una función denominada *hibrido.m* que calcula las estimaciones híbridas estudiadas a partir de las posiciones estimadas por RSS y GPS, además proporciona los resultados tanto utilizando errores medios como los errores instantáneos en cada una de las posiciones para el cálculo de los parámetros de combinación híbrida. En este caso sí que se tienen los errores instantáneos ya que el *Main* está destinado para simular y validar el sistema, con lo que es necesario conocer las posiciones reales que se pretenden estimar. La función *hibrido.m* también proporciona los errores cometidos por las estimaciones híbridas, tanto error medio como error en cada una de las posiciones medidas. Para poder utilizar la función *hibrido.m* se define una variable (GPS\_ON) que indica las posiciones en las que se obtiene posicionamiento GPS y por lo tanto permite diferenciar los escenarios interiores, donde sólo se utiliza RSS, con el resto de escenarios.



**Figura 28.** Correspondencia sistema coordenadas local y UTM.

Por último para mostrar los distintos resultados que permitan evaluar las prestaciones del sistema y estudiar posibles problemas o mejoras se crea un script denominado *resultado* que muestra por pantalla y representa en figuras la información de mayor interés para la valoración del sistema de localización híbrido.

#### **4.3.4 Fase de entrenamiento.**

Una vez tratadas las distintas fases de programación, para conseguir el funcionamiento correcto de la WSN y la posibilidad de simular y validar el sistema de localización híbrido estudiado en la presente memoria, se tratan los aspectos relacionados con la implementación del sistema en una aplicación concreta. Se ha visto que para el funcionamiento de las aplicaciones que forman el sistema se necesitan ciertos parámetros o criterios que deben definirse u obtenerse con anterioridad a la puesta en marcha del sistema. Entre estos parámetros se encuentran:

- Parámetros de propagación a utilizar para el posicionamiento RSS. Se ha comentado que se dispone de una tabla que contiene varios parámetros de propagación obtenidos en distintos escenarios, sin embargo ya se habló en el **Capítulo 2** que estos parámetros se deberían obtener a partir de medidas tomadas en el escenario que se utilizará el sistema de localización. De esta manera, lo inteligente sería tomar estas medidas antes de poner en marcha el sistema.
- En el capítulo anterior se proponen varias técnicas de estimación híbrida, sin embargo el sistema de localización sólo utilizará una de estas técnicas con lo que antes de iniciar el sistema hay que decidir la técnica a utilizar y para ello sería útil probar las distintas técnicas en el escenario del sistema de localización.
- La correspondencia entre el sistema local y el UTM puede no ser la correcta, siendo necesaria una translación en las coordenadas referenciadas obtenidas para que coincidan con las coordenadas del sistema local. Esta translación se debe encontrar con anterioridad a la implementación final del sistema para informar al código principal (*Main*).

De este modo, dentro del diseño del sistema se introduce una etapa previa a la puesta en marcha que se denominará **fase de entrenamiento**. En esta fase de entrenamiento se obtienen estos parámetros o criterios necesarios para la implementación final del sistema a partir de la simulación del sistema mediante el código *Main*, que permitirá escoger:

- Los parámetros que proporcionan un mejor posicionamiento en el escenario analizado.
- La mejor técnica híbrida para el escenario de localización en cuestión.
- La translación necesaria para hacer coincidir el sistema de coordenadas local con el sistema de coordenadas GPS.

En el archivo *Main* se incluye una función que permite calcular los parámetros de propagación en el escenario dado (*parametros.m*), es decir, a partir de las posiciones reales y las posiciones de las motas anclas que definen el escenario de localización se pueden obtener las distancias entre las posiciones reales y las motas anclas. Con estas distancias y las RSS medidas para cada una de las motas anclas en las distintas posiciones reales se puede aplicar el método de parametrización del escenario mostrado en el apartado 2.2.2. En el capítulo de resultados se comprobará que obteniendo los parámetros de propagación con este procedimiento proporciona mejores resultados de localización que obteniendo los parámetros de propagación a partir de medidas en línea recta. Ya se ha estudiado que los entornos inalámbricos pueden variar en el tiempo, por eso puede ser útil repetir esta fase de entrenamiento para ir actualizando los parámetros de propagación utilizados en el sistema de localización híbrido. Es posible que los parámetros de propagación obtenidos tras repetir la fase de entrenamiento varíen respecto los de la última fase de entrenamiento, sin embargo esto no quiere decir que los resultados de la localización sean mejores ya que cabe la posibilidad que las medidas en la última fase de entrenamiento no tuvieran la calidad suficiente como para modelar correctamente el escenario, por ello el código *Main* permite comparar los resultados obtenidos con los distintos parámetros y escoger aquellos que proporcionan un mejor posicionamiento.

Normalmente, todo y que el entrono inalámbrico sea variable y pueda cambiar a lo largo del tiempo, la precisión relativa entre el posicionamiento GPS y RSS suele ser la misma, es decir, en los escenarios en los que GPS va bien o mal seguirá siendo así todo y que el entrono varíe, puede mejorar o empeorar un poco pero a grandes rasgos seguirá igual. Con esto se quiere decir que para la selección de la mejor técnica híbrida en un escenario en particular bastaría con realizar una sola vez la fase de entrenamiento ya que al mantenerse la precisión relativa entre estimaciones la mejor técnica no cambiará de la BLUHE a alguna de ponderación o viceversa. Si embargo si que es posible que cambie de la ponderación simple a la ponderación modificada, con lo que no está de más aprovechar las fases de entrenamiento para comprobar si la mejor técnica de estimación sigue siendo la ponderación simple o modificada.



Por último, en cuanto a la translación de las coordenadas UTM referenciadas hay dos formas de obtenerla a partir de la fase de entrenamiento:

1. Hacer coincidir el sistema de coordenadas local con el sistema de coordenadas UTM directamente, es decir, colocar las motas anclas de tal forma que el eje x es positivo cuando se desplaza hacia el Norte y el eje y cuando se desplaza hacia el Este.
2. A partir de las coordenadas UTM de las motas anclas, referenciadas a la mota ancla de referencia. De esta manera se verá si coinciden los ejes y si sus sentidos son los correctos.

En el *Main* se utiliza la segunda alternativa comparando los signos de las coordenadas UTM referenciadas con los signos de las coordenadas locales de las motas anclas. Aun así tras haber realizado medidas en varios escenarios es posible determinar la dirección del Norte y el Este fácilmente y utilizar la primera alternativa.

Un aspecto importante de la fase de entrenamiento es que además de ser la fase en la que se obtienen todos los criterios comentados debe ser la fase en la que se decidan los aspectos relacionados con los aspectos de programación que dependan de las distintas aplicaciones. El archivo principal (*Main*) utilizado en el PFC permite validar y simular el sistema de localización híbrido, que es uno de los objetivos principales del proyecto, además de ser el archivo utilizado para realizar la fase de entrenamiento. Para obtener las distintas capacidades necesarias para las aplicaciones utilizadas se debería tener en cuenta básicamente dos aspectos:

1. Los archivos de XSniffer generados para utilizar el *Main* se toman en posiciones estáticas. En la mayoría de aplicaciones los datos irán llegando y no necesariamente la mota desconocida debe estar en una posición estática. Una posible modificación sería generar estos archivos cada cierto instante de tiempo, de tal manera que el archivo generado contenga medidas obtenidas en distintas posiciones que al hacer la media (en el *Main*) de las posiciones estimadas se debería obtener la posición media del conjunto de posiciones en las que se han tomado las medidas que contiene el archivo de XSniffer.
2. Se ha mencionado que para la función *hibrido.m* se necesita una variable (GPS\_ON) que indique las posiciones en las que no se recibe GPS. En el caso del PFC se utiliza una variable previamente definida que indica las posiciones en las que se han tomado medidas y no se obtenía localización GPS. En cualquier tipo de aplicación, no relacionada con la simulación de sistemas, esta variable se debería generar instantáneamente cada vez que se recibe un mensaje de la mota desconocida, así que habría que modificar el código para que generase esta variable a partir de los datos GPS obtenidos en la WSN, de tal manera que se identifique la no recepción GPS con la recepción de los campos GPS a 0.

Estos aspectos y modificaciones y otros relacionados con las mismas aplicaciones en las que se utilizará el sistema se podrán establecer a partir de la fase de entrenamiento, que sea cual sea la aplicación a desarrollar siempre se podrá utilizar la presentada en el presente proyecto realizada con el archivo *Main*. Estas modificaciones y las posibles ampliaciones generarán un archivo que deberá ejecutar el dispositivo que deseemos que proporcione el posicionamiento, en el caso del presente proyecto se ha utilizado un PC como el dispositivo que genera la localización, sin embargo este dispositivo podría ser la misma mota, algún dispositivo de presentación de datos (paneles informativos), un teléfono móvil,... Es importante destacar que según sea este dispositivo el lenguaje de programación será distinto, en nuestro caso se utiliza MATLAB pero en la mayoría de dispositivos no se puede permitir el lujo de utilizar este tipo de *software* con lo que se deberían generar las funciones creadas en este PFC con MATLAB con el lenguaje de programación adecuado para programar el dispositivo en cuestión.

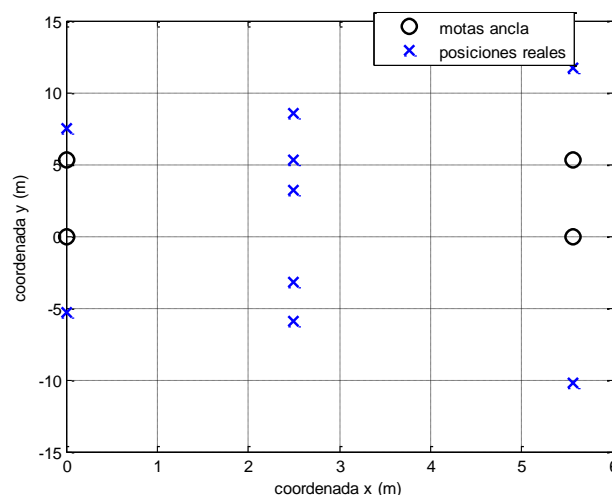


## 5 Resultados Híbridos.

Los capítulos anteriores se han dedicado a proporcionar las bases del sistema de localización híbrido y su construcción a partir de la formación de una WSN correctamente configurada. En este capítulo se muestran los resultados del posicionamiento del sistema obtenidos a partir de medidas experimentales, que permitirán validar o no de forma general el sistema diseñado e implementado en la presente memoria. Además, estos resultados mostrarán las limitaciones o problemas que pueda tener el sistema en ciertos casos particulares. Por lo tanto el objetivo principal del capítulo es proporcionar los resultados necesarios para validar el sistema de posicionamiento híbrido, para ello se necesita:

- Medidas en distintos escenarios que permitan obtener unos parámetros de propagación a utilizar en el cálculo de distancias con RSS para obtener la localización. Con los parámetros obtenidos se crea una tabla de parámetros para posteriormente poder elegir los parámetros que mejores resultados proporcione.
- Resultados de posicionamiento RSS en distintos escenarios interiores y exteriores y con distintos parámetros de propagación.
- Resultados del posicionamiento GPS extraído de la *sensor-board*.
- Resultados del posicionamiento híbrido a partir de los resultados anteriores.

Para obtener todas estas medidas y resultados de posicionamiento se utiliza el archivo *Main* con datos obtenidos en distintos entornos y con un despliegue de motas distinto. Para la realización de las medidas se dispone de un total de 6 motas Iris de las cuales 4 se utilizan como motas anclas, 1 como mota desconocida o a localizar y la restante se utiliza como *gateway*, cada una programada con la aplicación que le corresponda (**Capítulo 4**). De esta manera, en cada uno de los escenarios en los que se tomen las medidas se repartirán las 4 motas anclas tomando una como referencia y anotando las posiciones de las otras tres relativas a la de referencia que es la posición (0,0). Una vez distribuidas las cuatro motas anclas en el escenario de medida y fijado el sistema de coordenadas local, se sitúa la mota desconocida en varias posiciones dentro del rango de cobertura de la WSN (**Figura 29**) y se van recibiendo sus datos en la mota funcionando como *gateway*, que pasará los datos a XSniffer y se creará un archivo en cada una de las posiciones. No se ha comentado, pero el *Main* trabaja únicamente con una variable que contiene todos los datos de XSniffer, es decir, esta variable debe contener todos los archivos tomados en cada una de las posiciones. Para ello, una vez creado los archivos con XSniffer se abren y se guarda la variable creada en MATLAB en el directorio del escenario correspondiente, una vez se tienen todas las variables en MATLAB se unen todas en una única matriz y se crea otra variable que contenga el tamaño de cada una de las variables creadas en MATLAB, de tal manera que se puedan seleccionar los datos de cada una de las posiciones en todo momento.



**Figura 29.** Procedimiento para tomar las medidas.

Además de esta variable principal, ya se ha comentado, que para el funcionamiento del *Main* son necesarias otras variables como las posiciones de las motas anclas, por ello se crea un *script*, *creacion\_variables.m*, que permite generar todas las variables en MATLAB necesarias para el funcionamiento del *Main* en cada uno de los escenarios de medida. Este *script* permite la creación de las siguientes variables:

- Posiciones de las motas anclas según el sistema de coordenadas local definido en el escenario.
- Posiciones en las que se ha situado la mota desconocida para obtener las medidas que permitan obtener las distintas estimaciones.
- Número de posiciones a estimar.
- Variable con las longitudes de las variables que representan los archivos generados con XSniffer en cada una de las posiciones.
- Variable con todos los archivos generados con XSniffer.
- Variable que indica las posiciones en las que se recibe localización GPS.

A continuación se muestran las medidas y resultados necesarios para la validación del sistema. En primer lugar, se comenta las medidas tomadas para la obtención de los parámetros de propagación y se presentan los escenarios tratados en el capítulo. A continuación, se muestran, comentan y comparan los resultados del posicionamiento obtenido con RSS, GPS y las técnicas híbridas a partir de las medidas obtenidas. Comentar que en el capítulo sólo se presentan los escenarios más significativos, sin embargo en el **ANEXO 3** se adjunta el resto de medidas y resultados obtenidos en el resto de escenarios procesados a lo largo del PFC. Por último se presentan las conclusiones extraídas de los resultados obtenidos.

## 5.1 Parámetros de propagación y escenarios de medida.

Ya se ha visto que la relación entre la potencia recibida y la distancia puede llegar a ser compleja en entornos de propagación inalámbricos como es el caso de las WSN. Por ello se han definido los parámetros  $n$  y  $\gamma$  que modelan la potencia recibida o RSS en función de la distancia. Estos parámetros se obtienen a partir de medidas empíricas de potencia recibida en distintas distancias con el objetivo de obtener una aproximación de la relación entre RSS y la distancia en el entorno de propagación en el que se toman las medidas empíricas. Como el objetivo es encontrar la variación de la RSS con la distancia estas medidas se pueden tomar en línea recta y percibir así como afecta la distancia en la potencia recibida, concretamente se sitúa una mota en un lugar fijo que recibirá la potencia de otra mota que se va alejando hasta que ya no se obtiene potencia, en las distintas posiciones que se toman las medidas es posible que éstas varíen un poco con lo que en estos casos se toma la media de todos los valores recogidos y se utiliza ese valor como el valor de la potencia recibida. De esta manera se obtienen los parámetros que modelarán la potencia recibida en el entorno en el que se han tomado las medidas, sin embargo estas medidas no tomarán en cuenta efectos como la no visión directa entre motas, presencia de obstáculos que puedan absorber potencia o los efectos que pueda provocar el despliegue de las motas de la WSN ya que estas motas pueden estar enterradas en el suelo, elevadas, dentro de un vehículo,... Así que no es suficiente con obtener la variación de la potencia con la distancia debida a los efectos de propagación del entorno, si no que además hay que considerar todos los efectos relacionados con el escenario de localización en cuestión, por ello las medidas empíricas que permitan obtener los parámetros de propagación se tomarán en los escenarios de localización, donde las motas estarán situadas en el sitio que ocuparían en la WSN que forma el sistema de localización híbrido tratado.

### 5.1.1 Parámetros en línea recta.

Ya se ha comentado que los parámetros extraídos con medidas en línea recta pueden no tener en cuenta ciertos efectos importantes presentes en los escenarios de localización del sistema híbrido, sin embargo puede haber casos en los que estos parámetros proporcionen una representación del escenario más próxima a la real que los parámetros extraídos en los escenarios de localización, además interesa tener el máximo de parámetros de propagación posibles para poder describir así un gran número de entornos de propagación y tener así siempre disponible algún parámetro de propagación para los nuevos escenarios en los que se aplique el sistema. Ya se ha comentado que en la fase de entrenamiento del sistema se escogerán aquellos parámetros, de entre todos los disponibles, que proporcionen los mejores resultados en la localización, por lo tanto contra más parámetros de propagación disponibles más posibilidades de obtener un posicionamiento preciso. Por todo esto, a continuación se muestra la toma de medidas en línea recta en un escenario interior y otro exterior y los parámetros resultantes tras tomar la expresión (5) en el apartado de parametrización del escenario del **Capítulo 2**.

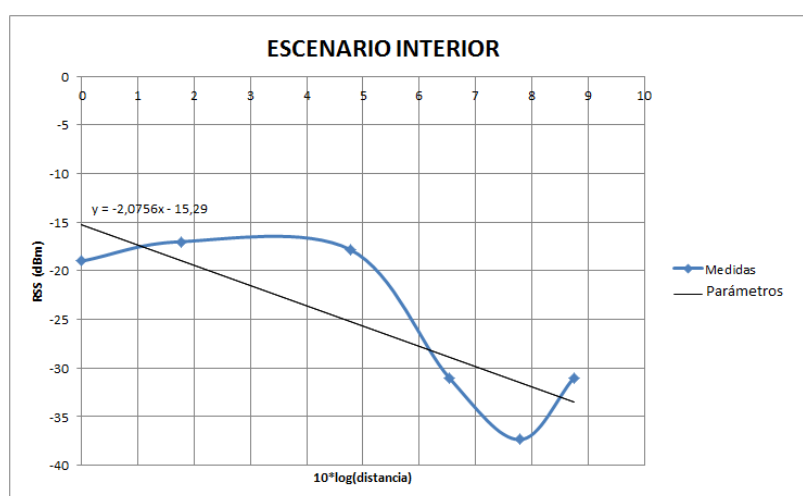
- **Escenario interior.**

En este escenario se pretende realizar medidas de potencia para la obtención de unos parámetros que modelen un entorno de propagación interior, donde se encuentran varios obstáculos como paredes, puertas, muebles,... concretamente este escenario es el pasillo de una vivienda de un piso. Habitualmente, en este tipo de escenarios, se obtiene una  $\gamma$  superior a 2, debido a la absorción de potencia por los obstáculos y las reflexiones que puedan provocar las interferencias destructivas o constructivas en la potencia recibida. Si que hay casos concretos de entornos interiores donde la  $\gamma$  puede llegar a ser menor que dos, estos casos concretos son aquellos en los que se encuentran pasillos largos y altos que producen un efecto de guía de onda en el medio de propagación. En el caso del escenario interior en el que se toman las medidas el pasillo tiene unos 7 metros de largo, obteniendo las medidas mostradas en la **Tabla 2** que representan la **Figura 30** donde se representan las potencias medidas en función de la distancia de medida.

ESCENARIO INTERIOR		
DISTANCIA (m)	10*LOG(dist.)	RSS (dBm)
1	0	-19
1.5	1.76	-17
3	4.77	-17.81
4.5	6.53	-31
6	7.78	-37.36
7.5	8.75	-31

**Tabla 2.** Medidas de potencia en escenario interior.

Con estas medidas se obtienen los siguientes parámetros ( $n = -15.29$ ,  $\gamma = 2.07$ ) que determinan la recta representada en la **Figura 30**. Esta  $\gamma$  no se corresponde con lo esperado en un escenario interior, debido básicamente al efecto que provoca que en el margen de distancias entre 1 y 3 metros la potencia aumente en vez de disminuir, esto hace que los parámetros tengan en cuenta una pendiente menor que la original presente en la última parte de las medidas (entre 4 y 7 metros). En la última medida se observa claramente una interferencia constructiva ya que la potencia aumenta considerablemente.



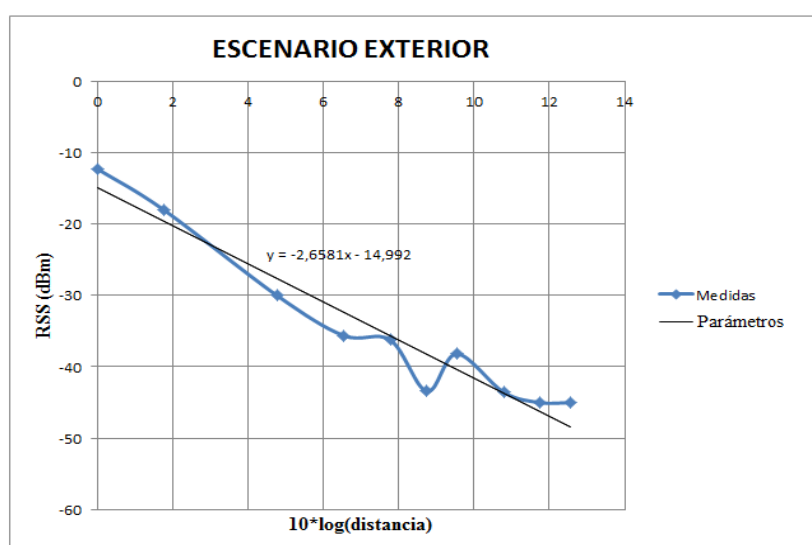
**Figura 30.** Medidas en escenario interior y parámetros resultantes.

- **Escenario exterior.**

En este caso se pretende obtener unos parámetros que modelen un entorno de propagación exterior, donde no se encuentran muchos obstáculos que obstruyan la visión directa y represente un entorno al aire libre. En este tipo de escenarios donde las medidas son limpias en el sentido de no haber ningún elemento que intervenga en la propagación de las señales se suele obtener un valor de  $\gamma$  igual 2 (fórmula de Friis). El escenario exterior en el que se han tomado las medidas es un terreno llano completamente asfaltado, en el que no circula ningún tipo de vehículo (es una carretera cortada) excepto en algún momento que pasa alguna persona en bicicleta, corriendo o paseando al perro. A diferencia del caso anterior, se toma medidas hasta la distancia en la que se recibe la potencia mínima (-45 dBm) y se toma un conjunto de medidas mayor mostrado en la **Tabla 3**, que describen la variación de la RSS en función de la distancia representada en la **Figura 31**. Los parámetros obtenidos son los siguientes: ( $n = -14.99$ ,  $\gamma = 2.65$ ) y representan la recta de la **Figura 31** que ha diferencia que en el caso anterior se aproxima considerablemente a la variación de la RSS medida.

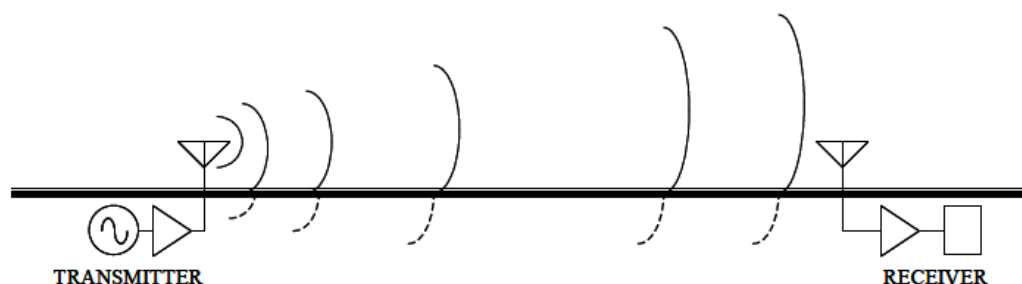
ESCENARIO EXTERIOR		
DISTANCIA (m)	10*log(dist.)	RSS (dBm)
1	0	-12.27
1.5	1.76	-18
3	4.77	-30
4.5	6.53	-35.6
6	7.78	-36.24
7.5	8.75	-43.43
9	9.54	-38.17
12	10.79	-43.56
15	11.76	-45
18	12.55	-45

**Tabla 3.** Medidas de potencia en escenario exterior.



**Figura 31.** Medidas en escenario exterior y parámetros resultantes.





**Figura 32.** Efecto suelo que produce la absorción de la señal [39].

Por otro lado, se obtiene una  $\gamma$  mayor que 2 debido a las reflexiones que pueda producir el suelo y principalmente a la absorción de la señal por el mismo, ya que las motas se encuentran situadas sobre él produciéndose el efecto suelo mostrado en la **Figura 32**. Este efecto se puede apreciar también en la **Figura 31** donde la pendiente en la primera parte de la gráfica (entre 1-4 metros) es más pronunciada que la pendiente en la última parte de la gráfica, se puede comprobar como en la primera parte la RSS disminuye en unos 20 dB mientras que en la segunda parte sólo disminuye unos 10 dB, de tal manera que el efecto suelo disminuye con la distancia. Además en las medidas tomadas en 7.5 y 9 metros parece haber interferencias destructivas y constructivas respectivamente, efecto que también contribuye al valor de los parámetros de propagación obtenido.

A diferencia del caso anterior, las medidas representan un decaimiento de la potencia lineal con la distancia, exceptuando las interferencias constructivas y destructivas identificadas, que se corresponde con entornos exteriores no complejos, es decir, sin obstáculos que obstruyan visión directa y reflejen la señal produciendo efecto *multipath*. Aun así ese decaimiento es mayor al presentado en espacio libre, todo y ser un entorno en espacio libre, mientras que en el caso anterior todo y no ser un entorno en espacio libre se aprecia un decaimiento más próximo al de espacio libre. Con esto se quiere dejar presente que no se pueden elegir unos parámetros de propagación basándose en si el escenario es interior o exterior, pero que la aproximación de los parámetros será más correcta en entornos exteriores que no en interiores, básicamente a la complejidad presente en esos escenarios. Por ello resulta más conveniente tomar estas medidas en escenarios de localización, es decir, tomar medidas tanto en entornos interiores como exteriores pero en vez de tomar las potencias en línea recta y con visión directa, tomarlas situando las motas como si estuvieran en la WSN del sistema de localización. A continuación se muestra este tipo de medidas y los parámetros obtenidos.

### 5.1.2 Parámetros en escenarios de localización.

Para que los parámetros de propagación tengan en cuenta todos los efectos que afectan a la propagación de las señales en los escenarios en los que se utilizará el sistema de localización híbrido se montarán distintos escenarios de posicionamiento para medir las potencias recibidas y extraer los parámetros. Para ello se distribuyen las motas anclas tal y como estarían en el sistema de localización y se va situando la mota desconocida en distintas posiciones. Es importante que estas posiciones sean variadas para poder incluir todo tipo de efectos (efecto suelo, visión directa, *multipath*, *shadowing*,...) y de tal manera que se mida la RSS de las distintas motas anclas en una gran cantidad de distancias, que muchas de ellas serán iguales obteniendo distintos valores de RSS, pudiendo así considerar también el efecto aleatorio de la RSS que hace que en una misma distancia se puedan obtener valores distintos de RSS.

A continuación se presentan los parámetros extraídos en los escenarios de localización cuyos resultados de posicionamiento serán tratados más adelante.

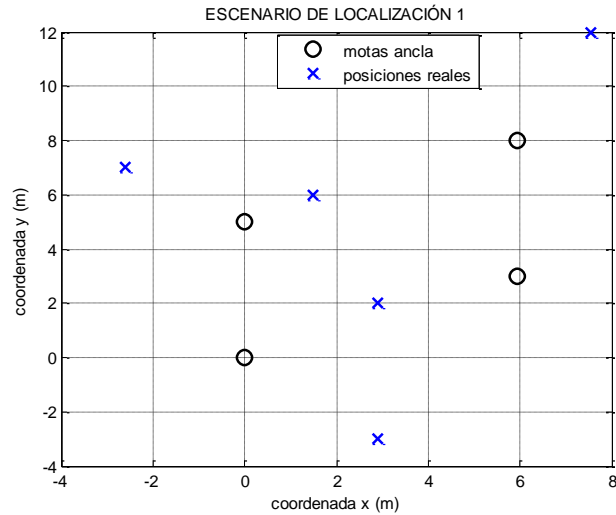
#### - Escenario de localización 1 (E1).

El escenario de localización 1 está situado en el entorno del escenario exterior en el que se han tomado las medidas en línea recta del apartado anterior. En este caso las motas anclas han sido elevadas 0.3 metros del suelo y la mota desconocida se ha situado encima de él. De esta manera se pretende simular los casos en los que las motas anclas se deben situar en un lugar elevado y la mota desconocida se desplace por el suelo como puede ser el caso de posicionar un animal terrestre, vehículo,... Las motas anclas se sitúan separadas entre sí más de 5 metros formando el sistema de coordenadas local representado en la **Figura 33**, donde además se muestran las distintas posiciones en las que se han tomado medidas de RSS. Estas posiciones se recogen en la **Tabla 4** donde P1, P2,..., P5 y M1,..., M4 se corresponden con las posiciones donde se toman las medidas y las posiciones de las motas anclas respectivamente.

Coordenadas	P1	P2	P3	P4	P5	M1	M2	M3	M4
X	2.9	2.9	1.5	-2.6	7.5	0	5.95	0	5.95
Y	-3	2	6	7	12	0	3	5	8

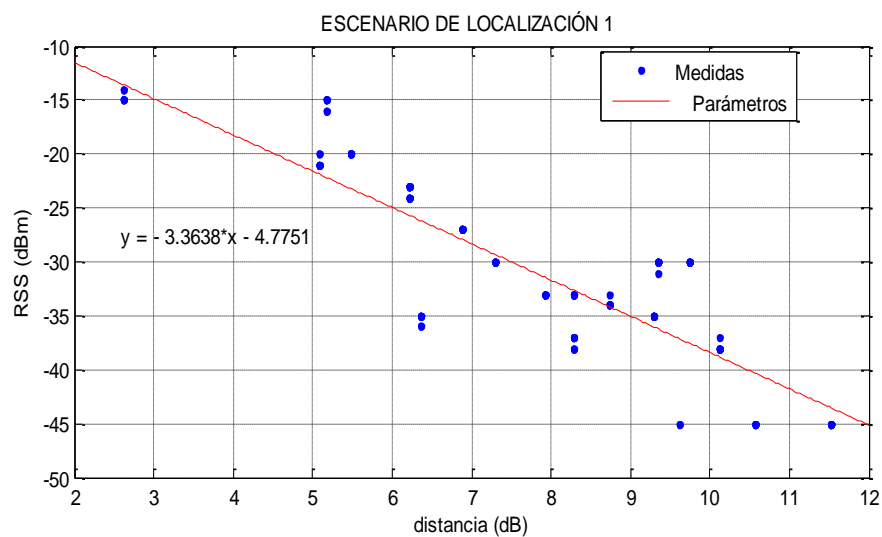
**Tabla 4.** Coordenadas del escenario de localización 1.

Con los valores de la **Tabla 4** se puede extraer las distancias entre las distintas motas anclas y la posición a la que se toman las medidas, concretamente se obtendrán 4 distancias en cada una de las posiciones de medida (una por cada mota ancla). Con este conjunto de distancias y las RSS medidas en cada una de las posiciones para cada una de las motas anclas se obtiene el conjunto de RSS en función de la distancia representado en la **Figura 34**, donde además se incluye la recta definida por los parámetros de propagación obtenidos ( $n = -4.77$  ,  $\gamma = 3.36$  ). En este caso se aprecia un claro aumento del parámetro  $n$  (potencia recibida a 1m de distancia) respecto al parámetro  $n$  obtenido anteriormente, debido a la reducción del efecto suelo causada por la elevación de las motas anclas.



**Figura 33.** Escenario de localización 1.

En cuanto al parámetro  $\gamma$  se observa un ligero aumento respecto el caso de línea recta debido al hecho de dejar la mota desconocida sobre el suelo, que no sólo produce el efecto suelo comentado sino que además modifica el diagrama de radiación de las motas. En el caso anterior al tener todas las motas sobre el suelo la modificación del diagrama de radiación era común a todas las motas y se traducían en una atenuación constante en todas las distancias, sin embargo en el caso del escenario de localización 1 sólo se sitúa sobre el suelo la mota desconocida y por lo tanto sólo su diagrama de radiación se ve afectado produciendo atenuaciones que dependerán de la posición en la que se reciben las señales. Esto se puede apreciar en la **Figura 34** donde las medidas entre 0-5.5 dB de distancia disminuyen más o menos pero sin cambios bruscos, mientras para distancias mayores las medidas varían considerablemente según la distancia siendo el promedio de estas medidas el valor que los parámetros obtenidos tratarán de seguir, por eso la pendiente y por lo tanto el parámetro  $\gamma$  obtenido.



**Figura 34.** Medidas en escenario 1 y parámetros resultantes.

- **Escenario de localización 2 (E2).**

Igual que el escenario de localización anterior, este escenario se encuentra en la zona del escenario exterior en el que se han tomado las medidas en línea recta, un lugar algo más alejado donde se encuentran algunos árboles en cada banda de la calle, aun así es un espacio en zona libre y despejada como para tomar medidas GPS. Además en este escenario se han elevado todas las motas 33cm respecto el suelo, concretamente se sitúan sobre botellas de 2 litros. El escenario montado se muestra en la **Figura 35** cuyas coordenadas de motas anclas y posiciones en las que se toman las medidas se detallan en la **Tabla 5** y **Tabla 6** respectivamente.

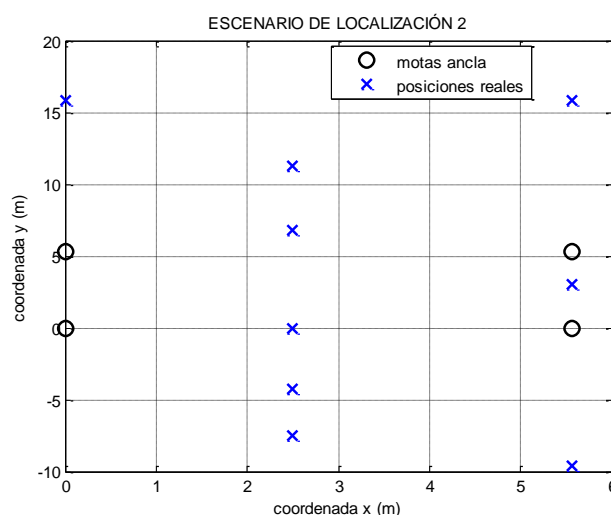
Coordenadas	M1	M2	M3	M4
X	0	0	5.57	5.57
Y	0	5.3	0	5.3

**Tabla 5.** Coordenadas de motas anclas en el escenario 2.

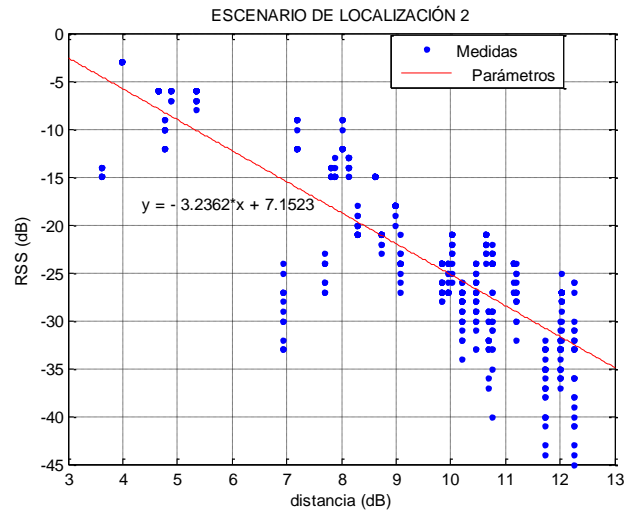
Coordenadas	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	5.57	2.5	2.5	2.5	5.57	2.5	2.5	5.57	0
Y	-9.6	-7.48	-4.24	0	3	6.8	11.3	15.8	15.8

**Tabla 6.** Posiciones de medida del escenario 2.

Igual que antes, con la **Tabla 5** y **6** se obtienen las distancias entre las posiciones de medida y las cuatro motas anclas que junto las medidas obtenidas en cada una de las posiciones se obtienen las RSS y los parámetros de propagación ( $n = 7.15$ ,  $\gamma = 3.23$ ) representados en la **Figura 36**. Igual que antes, al elevar la mota desconocida, el valor del parámetro  $n$  ha aumentado considerablemente respecto el caso anterior y el caso de las medidas en línea recta debido a la eliminación del efecto suelo y los efectos relacionados con la variación del diagrama de radiación de las motas.



**Figura 35.** Escenario de localización 2.



**Figura 36.** Medidas en escenario 2 y parámetros resultantes.

En este caso el parámetro  $\gamma$  no varía mucho con el del escenario 1, pero en este caso el aumento del parámetro no es debido a la modificación del diagrama de radiación de la mota si no a las interferencias constructivas y destructivas que se aprecian en distancias medias y largas en la **Figura 36**. Estas interferencias quedan reflejadas en el hecho que hay algunas distancias en las que se reciben distintas RSS, esta variación de la RSS con el tiempo para una misma distancia es debida al efecto *multipath* que aparece por los objetos que puedan reflejar las señales y provoquen distintos caminos por donde llegan las señales a la mota desconocida. Ya se ha dicho que en este escenario no se encuentra ningún obstáculo que impida la visión directa entre las motas anclas y la desconocida, con lo que lo único que puede causar estas interferencias es el suelo que al tener las motas elevadas cierta parte de la potencia radiada rebota en el suelo y llega a la mota desconocida por un camino que no es el directo.

- **Escenario de localización 3 (E3).**

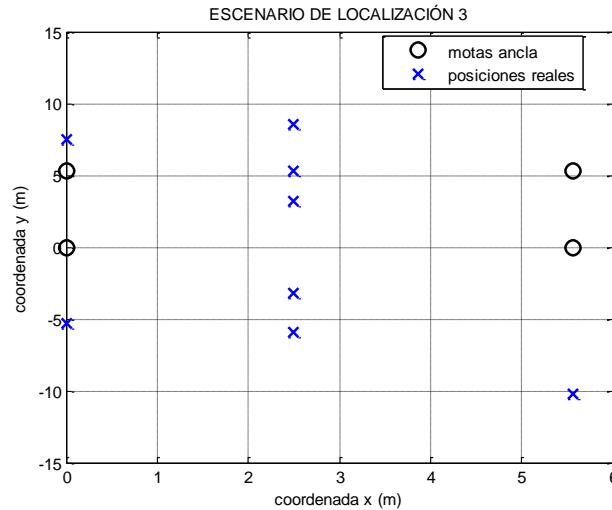
Este escenario es el mismo escenario de localización que el anterior (escenario de localización 2) pero realizando medidas en distintas posiciones, es decir, es el mismo entorno o escenario que el anterior (entorno exterior con arboles) y las motas anclas están situadas en las mismas coordenadas relativas que en el caso anterior (**Figura 37**). Con este escenario se pretende simular una nueva etapa de entrenamiento, considerando el escenario anterior como la etapa de entrenamiento principal con la que se pone en marcha el sistema. Las coordenadas de las motas anclas y las posiciones en las que se toman las medidas se muestran en la **Tabla 7** y **8** respectivamente.

Coordenadas	M1	M2	M3	M4
X	0	0	5.57	5.57
Y	0	5.3	0	5.3

**Tabla 7.** Coordenadas de motas anclas en el escenario 3.

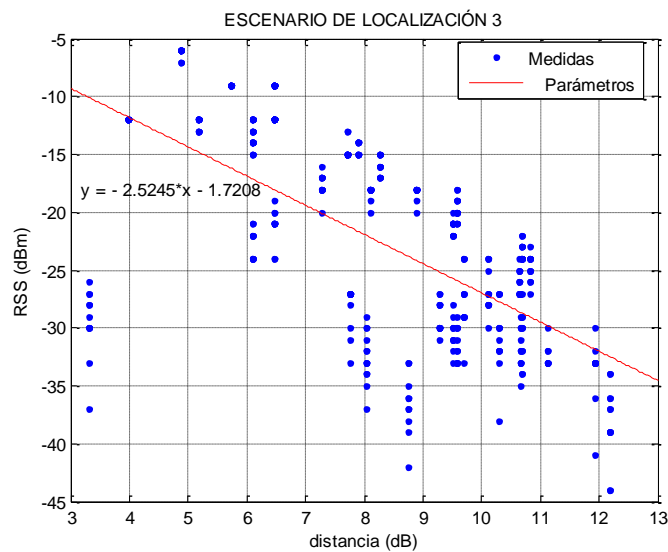
Coordenadas	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	5.57	2.5	0	2.5	2.5	2.5	0	2.5	5.57
Y	-10.23	-5.95	-5.35	-3.21	3.21	5.35	7.49	8.56	11.77

**Tabla 8.** Posiciones de medida del escenario 3.



**Figura 37.** Escenario de localización 3.

En la **Figura 38** se representa el conjunto de medidas obtenidas y los parámetros de propagación resultantes ( $n = -1.7$ ,  $\gamma = 2.52$ ). En este caso el parámetro  $n$  es menor que el obtenido con las medidas del escenario 2 debido posiblemente a la disminución de las baterías de las motas anclas. Por otro lado el parámetro  $\gamma$  es más próximo al parámetro obtenido en el escenario exterior tomando medidas en línea recta que ningún otro. En este caso se observan igualmente interferencias constructivas y destructivas, pero el efecto en la RSS hace que decaimiento promedio o equivalentemente el parámetro  $\gamma$  sea el obtenido. De la misma forma se puede pensar que de promedio en este caso hay más interferencias constructivas que destructivas respecto el caso del escenario 2 y por ello el parámetro  $\gamma$  obtenido en este caso es menor que el anterior, dejando presente que el decaimiento promedio de la potencia es menor. Igual que en el caso anterior las interferencias son producidas por el efecto *multipath* provocado por el rebote de las señales en el suelo, sin embargo al tomar medidas en distintas posiciones estas interferencias son distintas.



**Figura 38.** Medidas en escenario 3 y parámetros resultantes.

### 5.1.3 *Tabla de parámetros de propagación.*

En los apartados anteriores se han presentado algunos de los parámetros obtenidos en el PFC aprovechando también para presentar los escenarios de localización en los que se han realizado las pruebas del sistema de posicionamiento híbrido implementado. Sin embargo a lo largo de la realización del PFC se han podido tomar medidas en otros escenarios de localización que nos permiten obtener una tabla de parámetros de propagación con la que se elegirán los parámetros que proporcionen una localización más precisa en los escenarios de localización en los que se utiliza el sistema. Todos estos parámetros obtenidos a lo largo del PFC se reúnen en la **Tabla 9** y en el **ANEXO 3** se incluyen las medidas tomadas para obtener estos parámetros. Estas medidas se toman en una gran variedad de escenarios con distintas configuraciones, es decir, se toman medidas tanto en escenarios interiores como exteriores y se toman medidas con todas las motas elevadas, situadas en el suelo, con obstáculos o sin obstáculos que obstruyan la visión directa, ... De esta manera se obtiene un conjunto de parámetros que tienen en cuenta un gran tipo de efectos y entornos que permitirán tener disponible siempre un parámetro que modele un nuevo entorno de localización en el que se quiera utilizar el sistema híbrido.

TABLA DE PARÁMETROS DE PROPAGACIÓN		
Parámetros	n	$\gamma$
<b>1</b>	-4.77	3.36
<b>2</b>	7.15	3.23
<b>3</b>	-1.72	2.52
<b>4</b>	-15.29	2.07
<b>5</b>	-14.99	2.65
<b>6</b>	-15.28	2.19
<b>7</b>	-12.93	2.03
<b>8</b>	-12.62	2.63
<b>9</b>	-16.75	2.46
<b>10</b>	-18.83	3.13
<b>11</b>	-18.13	3.46
<b>12</b>	-23.94	1.81

**Tabla 9.** Tabla de parámetros de propagación utilizados en el PFC.

## 5.2 Resultados del posicionamiento.

Una vez se dispone de parámetros de propagación que modelan una gran variedad de escenarios y entornos de localización, se puede obtener el posicionamiento con RSS y combinarlo con la localización GPS para calcular las estimaciones híbridas. En este tema se tratan los resultados obtenidos del posicionamiento RSS, GPS y de las estimaciones híbridas en distintos escenarios de localización. Concretamente se mostrarán tres escenarios exteriores para comprobar el posicionamiento híbrido y un escenario interior para demostrar que es posible obtener localización en entornos interiores utilizando el posicionamiento RSS.

Para cada uno de los escenarios analizados se presentarán los siguientes datos:

- Sistema de coordenadas utilizado donde se muestran las posiciones de las motas anclas, relativas a la de referencia que será la mota ancla número 1 situada en el origen (0,0). Además se representan también las posiciones en las que se toman las medidas.
- Comentario de los parámetros que proporcionan mejores resultados en el posicionamiento.
- Se representan las estimaciones GPS y RSS y se muestran los resultados numéricos de las posiciones estimadas.
- Representación de las estimaciones híbridas y presentación de los errores cometidos en cada una de las posiciones y errores medios para cada una de las técnicas de localización.
- Las técnicas híbridas se obtienen utilizando parámetros instantáneos (un parámetro distinto para cada posición) y promedios (el mismo parámetro para todas las posiciones).

Recordar que las técnicas híbridas deberían utilizarse con los errores instantáneos en cada una de las posiciones, sin embargo como estos errores en la práctica no se tienen en el caso de este PFC se utilizan errores promedios. De aquí la diferencia entre los parámetros instantáneos y promedios utilizados para las técnicas híbridas. Por otro lado comentar que para la obtención de la localización con las medidas RSS se utiliza la expresión (10) utilizando únicamente 3 estimaciones de distancia, en [39'] se comprueba que es mejor utilizar sólo 3 medidas de distancia. Para el cálculo de los errores en cada una de las posiciones se utiliza la expresión (18) a partir de los que se obtiene el error medio de cada técnica de localización tomando la media de todos los errores instantáneos. Para el cálculo de los parámetros de combinación híbrida se utilizan las expresiones y criterios expuestos en el capítulo 3 para cada una de las técnicas propuestas. Estas expresiones se recogen en la **Figura 39**.



$$p = \begin{bmatrix} x \\ y \end{bmatrix} = (B^T \cdot B)^{-1} \cdot B^T \cdot c \quad (10)$$

$$B = \begin{bmatrix} x_1 - x_2 & y_1 - y_2 \\ x_1 - x_3 & y_1 - y_3 \end{bmatrix} \quad c = \frac{1}{2} \cdot \begin{bmatrix} x_1^2 - x_2^2 + y_1^2 - y_2^2 + d_2^2 - d_1^2 \\ x_1^2 - x_3^2 + y_1^2 - y_3^2 + d_3^2 - d_1^2 \end{bmatrix}$$

$$e = \sqrt{(x - x_r)^2 + (y - y_r)^2} \quad (18)$$

**PONDERACIÓN**

$$\hat{P}_H = (1 - \mu) \cdot \hat{P}_{RSS} + \mu \cdot \hat{P}_{GPS}$$

$$\mu = \frac{1}{ME_{GPS}} \quad \mu = \frac{\beta}{2} = \frac{ME_{RSS}}{2 \cdot ME_{GPS}}$$

**SIMPLE                      MODIFICADA**

**BLUHE**

$$\hat{P}_H = a_{BLUHE}^T \cdot x = \frac{\sigma_{RSS}^2 \cdot \sigma_{GPS}^2}{\sigma_{RSS}^2 + \sigma_{GPS}^2} \cdot \left( \frac{\hat{P}_{RSS}}{\sigma_{RSS}^2} + \frac{\hat{P}_{GPS}}{\sigma_{GPS}^2} \right)$$

$$a_{BLUHE} = \frac{\sigma_{RSS}^2 \cdot \sigma_{GPS}^2}{\sigma_{RSS}^2 + \sigma_{GPS}^2} \cdot \begin{bmatrix} \frac{1}{\sigma_{RSS}^2} \\ \frac{1}{\sigma_{GPS}^2} \end{bmatrix}$$

**Figura 39.** Expresiones utilizadas.

### 5.2.1 Escenarios exteriores.

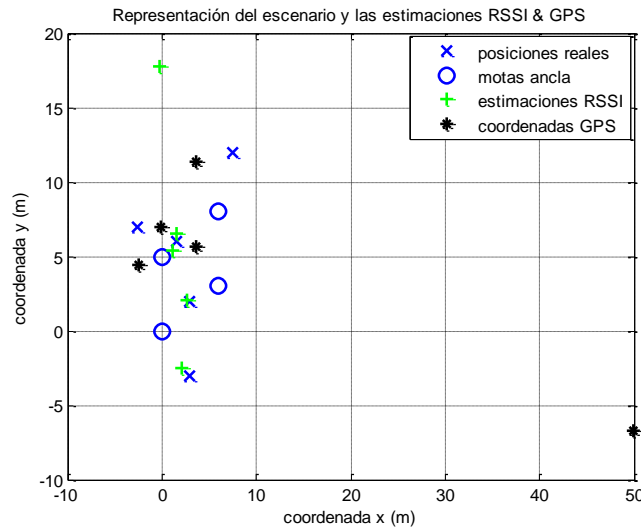
A continuación se muestran los resultados del posicionamiento en los escenarios exteriores presentados en el apartado 5.1.2. Estos escenarios no han sido los únicos en los que se han tomado medidas para comprobar el posicionamiento, no obstante son los escenarios que más información pueden aportar sobre el funcionamiento del sistema de posicionamiento híbrido. El resto de escenarios exteriores analizados se incluyen en el **ANEXO 3**.

- **Escenario de localización 1 (E1).**

Tal y como se ha dicho este escenario es el representado en la **Figura 33**, en el que se obtenían los parámetros de propagación **1** de la tabla de parámetros (**Tabla 9**). Antes de obtener el posicionamiento RSS que permita obtener las estimaciones híbridas, es necesario escoger los parámetros de propagación que se utilizarán para las estimaciones de distancia a partir de las medidas RSS. Para ello se comparan los errores medios en el posicionamiento para cada uno de los parámetros disponibles en la tabla de parámetros, presentados en la **Tabla 10** donde se observa que el menor error de posicionamiento se obtiene con los parámetros de propagación extraídos en el propio escenario de localización (Parámetros **1**:  $n = -4.77$   $\gamma = 3.36$ ).

Parámetros	1	2	3	4	5	6	7	8	9	10	11	12
Error medio (m)	<b>3.11</b>	30.96	50.74	12.38	3.52	8.56	25.19	4.4	3.84	3.75	4.21	4.93

**Tabla 10.** Errores medios de posicionamiento según los parámetros (E1).



**Figura 40.** Estimaciones RSS y GPS en escenario de localización 1.

Con los valores de la **Tabla 10** queda presente la dependencia del posicionamiento con la correspondencia entre los parámetros de propagación utilizados y el entorno en el que se encuentra el escenario de localización. Por ejemplo, los Parámetros **2** y **3** se obtuvieron elevando las motas del suelo y los Parámetros **4** en un escenario interior, mientras que **5** se obtuvo con las motas en el suelo y en un entorno exterior parecido al del escenario de localización 1 y por ello con esos parámetros se obtiene un error aceptable, aunque mayor que con los Parámetros **1**, mientras que con los otros se obtiene un error muy grande. Así, extrayendo los parámetros de propagación en una fase de entrenamiento en los distintos escenarios de localización y utilizándolos posteriormente para el funcionamiento del sistema, se puede asegurar que los resultados de la localización serán aceptables.

Una vez en escogidos los parámetros a utilizar en la estimación de distancias para el posicionamiento con RSS, se obtienen las estimaciones de posición con RSS representadas en la **Figura 40** donde además se incluyen las posiciones GPS obtenidas tras realizar las transformaciones y translaciones adecuadas, todo esto junto con el escenario de localización utilizado (Escenario de localización 1).

Las coordenadas de las posiciones obtenidas para cada una de las dos técnicas se detallan en la **Tabla 11** y **12**. Con el posicionamiento GPS se obtiene un error medio de 11.87 m, debido básicamente al error cometido en la primera posición que es de aproximadamente 50 metros mientras que en el resto de posiciones el error es siempre menor que 5 metros hasta incluso obteniendo errores de tan solo 2 m, con lo que el posicionamiento GPS puede ser aceptable en la mayoría de casos pero hay que tener cuidado con ciertas posiciones en las que el GPS no está funcionando correctamente. Al ser la primera posición en la que se toman medidas, es posible que el error tan grande en esa posición sea debido a que el receptor GPS aún no esté correctamente sincronizado con los satélites a los que esté conectado.

Coordenadas	P1_GPS	P2_GPS	P3_GPS	P4_GPS	P5_GPS
X	49.94	3.57	-0.06	-2.49	3.56
Y	-6.68	5.7	6.96	4.47	11.37
Error Inst.	47.19	3.76	1.83	2.53	4.04

**Tabla 11.** Coordenadas GPS.

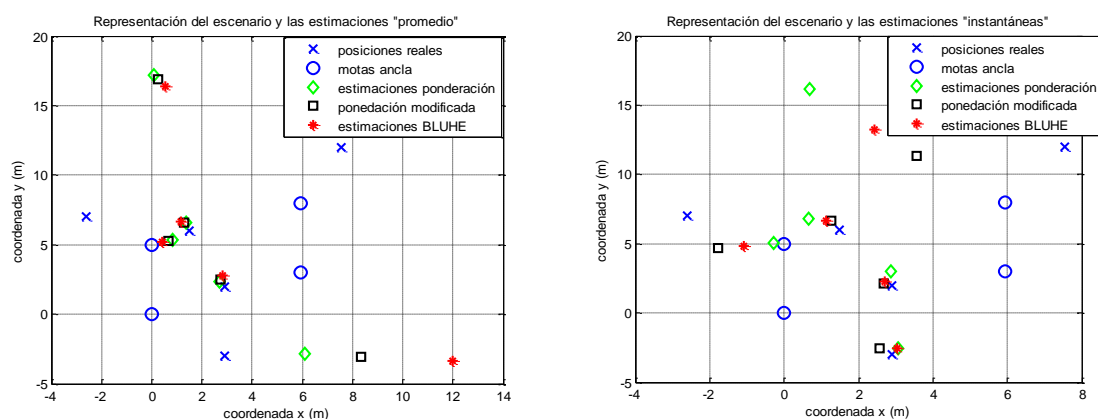
Como el error GPS medio es una medida importante para posteriormente utilizar en las estimaciones híbridas con el objetivo de obtener un error medio GPS más preciso, en el sentido que no se diferencie tanto con la mayoría de errores instantáneos, sería recomendable realizar medidas en más posiciones que las utilizadas en este escenario para poder realizar un buen promedio de tal manera que errores tan grandes como es el caso de la primera posición de este escenario de localización no afecten tanto al error medio.

Coordenadas	P1_RSS	P2_RSS	P3_RSS	P4_RSS	P5_RSS
X	2.05	2.63	1.51	1.15	-0.25
Y	-2.49	2.03	6.56	5.42	17.73
Error Inst.	0.99	0.27	0.56	4.07	9.68

**Tabla 12.** Coordenadas RSS.

En cuanto al posicionamiento con RSS el error medio utilizando los parámetros comentados es de 3.11 metros, obteniendo un error máximo de 9.68 m y llegando a obtener errores menores que 60 cm. A pesar de las estimaciones que son más o menos precisas se observa claramente como las estimaciones RSS siguen bastante bien la trayectoria de las posiciones reales, sin embargo se observa una posición en la que el error se dispara respecto el error del resto de posiciones (**Figura 40**). Esta posición es la posición que se sitúa más alejada de las motas anclas con lo que la localización se realiza a partir de la estimación de tres distancias grandes. En la obtención de los parámetros de propagación se ha observado que a largas distancias, donde la potencia recibida suele ser baja, hay más presencia de interferencias constructivas y destructivas que para distancias cortas, por este motivo las estimaciones de largas distancias con estos parámetros se ve perjudicada proporcionando normalmente unas estimaciones más erróneas de lo normal. En la **Figura 40** queda presente ya que las estimaciones tomadas con motas anclas cercanas se obtienen errores menores a 1 metro, mientras que las estimaciones tomadas con motas anclas más lejanas se obtienen errores entre 4 y 10 metros.

En posesión de ambas estimaciones básicas y los errores instantáneos y promedios de cada una se procede a la comprobación de los resultados obtenidos con las técnicas híbridas propuestas en el capítulo 3, obteniendo las posiciones estimadas para cada una de las técnicas representadas en la **Figura 41** donde la parte izquierda se corresponde con las estimaciones tomadas con parámetros obtenidos a partir de errores promedios y la parte derecha con parámetros obtenidos a partir de los errores instantáneos de las estimaciones RSS y GPS.



**Figura 41.** Estimaciones híbridas con parámetros promedios e instantáneos.

Con las estimaciones promedio se obtiene aproximadamente el mismo resultado que con el posicionamiento RSS pero perdiendo precisión en la primera posición, en cambio con las estimaciones instantáneas si que se obtienen buenos resultados en la mayoría de las posiciones excepto, como en el caso del posicionamiento RSS, aquellas que están más alejadas de las motas anclas. Para apreciar mejor las relaciones entre la precisión de las técnicas en la **Tabla 13** se muestran los errores obtenidos en cada una de las posiciones para cada una de las técnicas híbridas y sus errores promedio.

ERROR INSTANTANEO						ERROR MEDIO
TÉCNICA	P1	P2	P3	P4	P5	PROMEDIO
RSS	0.99	0.27	0.56	4.07	9.68	3.11
GPS	47.19	3.76	1.83	2.53	4.04	11.87
Ponderación	3.19	0.39	0.61	3.83	9.1	3.42
Ponderación Modificada	5.44	0.53	0.65	3.69	8.79	3.82
BLUHE	9.12	0.79	0.72	3.48	8.28	4.48
Ponderación Inst.	0.45	1	1.15	3.03	8.02	2.73
Ponderación Mod. Inst.	0.58	0.28	0.67	2.49	4.04	1.61
BLUHE Inst.	0.44	0.34	0.75	2.64	5.26	1.89

**Tabla 13.** Error de posicionamiento para cada técnica de localización (E1).

Con los resultados obtenidos se observa que para obtener mejora en el posicionamiento básico (RSS & GPS) se deben utilizar las técnicas híbridas con errores instantáneos para poder utilizar un parámetro de combinación híbrida en cada una de las posiciones, ya que en este caso obteniendo los parámetros de combinación híbrida con los errores medios de la localización básica no sólo no se consigue una mejora si no que se empeora los resultados del mejor posicionamiento básico, en este caso RSS. El problema de las estimaciones promedio, en este escenario de localización, es que el error medio de GPS difiere considerablemente del error instantáneo en todas las posiciones lo que provoca que las estimaciones híbridas (con parámetros promedio) no tengan tan en cuenta las estimaciones GPS cuando realmente están funcionando bien y lo mismo con las estimaciones RSS.

Todo y que los resultados empeoren utilizando las técnicas híbridas propuestas no significa que no funcionen bien, ya que utilizando los errores instantáneos las técnicas mejoran, si no que los parámetros utilizados para obtener las estimaciones híbridas no se obtienen correctamente o no se corresponden con los definidos en el capítulo 3. En la **Tabla 13** se puede comprobar como, para el caso de parámetros instantáneos, los mejores resultados se obtienen con la técnica híbrida de ponderación modificada seguida de la técnica híbrida BLUHE. Si se examina con más detalle los errores en cada una de las posiciones se comprueba que excepto en la primera posición el error obtenido con la técnica de ponderación modificada es menor que con la técnica BLUHE, esto es debido a que la diferencia entre errores en esas posiciones suele ser grande y esto permite a la ponderación modificada escoger la estimación más precisa y menospreciar la estimación errónea, mientras que en el BLUHE esta diferencia entre errores hace que la estimación errónea intervenga como ruido en la estimación más precisa y por lo tanto se obtiene una estimación algo más errónea que la más precisa de las estimaciones básicas.

Para terminar remarcar la dependencia de la ponderación modificada con el error GPS, ya comentada en su momento. En la posición **P4** y **P5** la mejor estimación básica es GPS y los resultados obtenidos con ponderación modificada son mejores en la posición 4 y en la posición 5 tienen la misma precisión, mientras que en las posiciones **P2** y **P3** la mejor estimación básica es RSS y los resultados con ponderación modificada obtenidos son peores en ambos casos, por lo tanto la ponderación modificada proporciona mejores resultados cuando la mejor estimación básica es la de GPS.

- **Escenario de localización 2 (E2).**

Este escenario de localización es el representado en la **Figura 35** obteniendo los parámetros de propagación **2** de la tabla de parámetros (**Tabla 9**). En este caso se puede comprobar en la **Tabla 14** como el mejor posicionamiento se obtienen con los parámetros de propagación extraídos en el escenario de localización (Parámetros **2**:  $n = 7.27$   $\gamma = 3.24$ ). Igual que en el caso anterior queda presente la dependencia del posicionamiento con los parámetros de propagación utilizados.

Parámetros	1	2	3	4	5	6	7	8	9	10	11	12
Error medio (m)	7.49	<b>5.28</b>	6.08	7.4	7.9	7.58	6.67	7.69	7.94	8.18	8.18	8.12

**Tabla 14.** Errores medios de posicionamiento según los parámetros (E2).

Los resultados de posicionamiento RSS obtenidos utilizando estos parámetros así como el posicionamiento GPS se exponen en la **Tabla 15**, en este caso no se incluye imagen de las coordenadas obtenidas porque no se aprecia la correspondencia entre las distintas posiciones reales y las posiciones estimadas. Con estos resultados se obtiene un error medio de 5.27 m para el posicionamiento RSS donde, igual que en el caso anterior, los mejores resultados se obtienen cuando se localiza con motas anclas cercanas, en este caso la mayoría de posiciones en las que se toman medidas se encuentran lejos de las motas anclas (**Figura 35**) por eso el error medio obtenido con posicionamiento RSS es mayor que en el caso anterior. Con esto queda comprobado la disminución de la precisión del posicionamiento RSS a medida que la distancia entre la mota desconocida y las motas anclas con las que se localiza aumenta.

Coordenadas	P1	P2	P3	P4	P5	P6	P7	P8	P9
<b>X_REAL</b>	5.57	2.5	2.5	2.5	5.57	2.5	2.5	5.57	0
<b>Y_REAL</b>	-9.6	-7.48	-4.24	0	3	6.8	11.3	15.8	15.8
<b>X_RSS</b>	6.88	-1.12	6.89	2.56	2.86	2.74	-0.01	-2.57	-1.49
<b>Y_RSS</b>	3.26	-2.72	-2.73	1.16	1.88	4.25	6.74	16.06	19.36
<b>X_GPS</b>	-2.61	-0.38	1.23	3.67	7.48	7.25	7.72	13.73	10.01
<b>Y_GPS</b>	-9.4	-7.65	2.33	6.42	4.23	10.49	9.96	15.62	19.59

**Tabla 15.** Coordenadas GPS y RSS en escenario de localización 2.

En cuanto al posicionamiento GPS se obtiene un error medio de 6.31 m reduciéndose considerablemente en comparación con el error medio GPS obtenido en el escenario anterior. Esta disminución es debida principalmente que en ninguna posición se obtiene un error tan grande como el error máximo obtenido en el escenario anterior, siendo en este caso el error máximo obtenido de aproximadamente 10 m obteniendo en el resto de posiciones errores entre 2 y 8 metros, tal y como se puede comprobar en la **Tabla 16**, donde además de los errores instantáneos y promedio de las estimaciones básicas se muestran los errores obtenidos con las distintas estimaciones híbridas.

ERROR INSTANTANEO										ERROR
TÉCNICA	P1	P2	P3	P4	P5	P6	P7	P8	P9	PROMEDIO
RSS	12.93	5.98	4.65	1.17	2.93	2.57	5.21	8.16	3.95	5.28
GPS	8.18	2.89	6.7	6.52	2.27	6.02	5.39	8.16	10.7	6.31
Ponderación	10.86	5.3	4.19	2.01	2.11	1.83	4.25	5.56	3.61	4.41
*PM	8.03	4.27	4.15	3.4	0.79	2.13	3.3	1.34	4.93	3.59
BLUHE	7.71	4.14	4.22	3.6	0.61	2.31	3.26	0.73	5.24	<b>3.54</b>
Pond. Inst.	11.31	4.54	4.21	1.98	0.68	1.81	4.11	6.15	3.6	4.26
**PMI	6.81	2.89	4.07	1.64	0.48	1.71	3.24	0.04	3.65	<b>2.73</b>
BLUHE Inst.	6.81	3.44	4.14	1.98	0.23	1.73	3.24	0.04	3.94	2.84

\*PM: Ponderación Modificada; \*\*PMI: PM Instantánea.

**Tabla 16.** Error de posicionamiento para cada técnica de localización (E2).

A diferencia del escenario anterior, en este caso las técnicas híbridas utilizando parámetros promedio si que consiguen reducir el error medio de las estimaciones básica. En el caso anterior se tenía una medida GPS muy errónea que a la hora de promediar hace que se obtenga un promedio del error GPS imprecisa (difiere en la mayoría de posiciones con el error instantáneo), en el escenario de localización actual al tener más posiciones con las que promediar se obtiene un error GPS medio que se parece en la mayoría de los casos, o al menos no difiere tanto, al error instantáneo en cada una de las posiciones, este es el motivo por el cual en el escenario actual las técnicas híbridas funcionan correctamente, es decir, consiguen disminuir el error de las estimaciones básicas.

Para las técnicas utilizando parámetros promediados los mejores resultados se obtienen con el BLUHE consiguiendo una reducción del menor de los errores medios básicos (RSS) del 32.83 % mientras que en el error GPS medio se obtiene una reducción del 44 %. Con estos resultados queda presente el correcto funcionamiento de las técnicas híbridas promedio cuando se obtienen correctamente los parámetros de combinación híbrida, es decir, cuando las medidas utilizadas para la obtención de los parámetros se corresponde con la definida por la técnica de localización híbrida.

En cuanto las técnicas híbridas instantáneas, igual que en el escenario anterior, los mejores resultados los proporciona la técnica de ponderación modificada obteniendo una reducción del 48% en el error medio del posicionamiento RSS, que en este caso es el mejor posicionamiento básico. Comparando los resultados de las estimaciones híbridas instantáneas se observa que los resultados de la ponderación modificada se aproximan bastante a los obtenidos con BLUHE, es decir, que siempre sale a cuenta combinar las estimaciones que no escoger únicamente la mejor de las dos, cosa que no ocurre con la ponderación simple. La diferencia y lo que hace que la ponderación modificada sea algo mejor que BLUHE es que para las posiciones en las que se tiene una gran diferencia entre el error GPS y RSS la ponderación modificada escoge la estimación menos errónea con el objetivo que la estimación imprecisa actúe como ruido en la estimación de la posición híbrida, que es el caso del BLUHE. Con esto se quiere demostrar que todo y que la ponderación modificada sea una técnica intuitiva sus resultados son iguales o hasta incluso mejores que la técnica BLUHE que ha sido obtenida con un razonamiento matemático.

- **Escenario de localización 3 (E3).**

Por último se muestran los resultados del escenario de localización representado en la **Figura 37** donde se obtienen los parámetros de propagación **3** de la tabla de parámetros. Igual que en los escenarios anteriores, en la **Tabla 17** se detallan los errores medios del posicionamiento RSS según los parámetros de propagación utilizados, obteniendo los mejores resultados con los parámetros de propagación extraídos en el escenario de localización 2 (Parámetros **2**:  $n = 7.27$   $\gamma = 3.24$ ). A diferencia de los casos anteriores, en este caso no se obtiene el mejor posicionamiento con los parámetros extraídos en el propio escenario de localización, sin embargo los parámetros óptimos han sido extraídos en el escenario de localización 2, que es el mismo escenario pero tomando medidas en distintas posiciones.

Parámetros	1	2	3	4	5	6	7	8	9	10	11	12
Error medio (m)	5.86	<b>5.17</b>	5.93	5.77	6.3	5.95	5.35	6.07	6.33	6.58	6.58	6.53

**Tabla 17.** Errores medios de posicionamiento según los parámetros (E3).

Si se revisa la **Tabla 9** se nota una diferencia considerable entre los parámetros obtenidos en este escenario de localización y los obtenidos en el escenario 2, con lo que se puede concluir que las posiciones en las que se han tomado las medidas en el presente escenario no proporcionan la suficiente información como para modelar correctamente el entorno y los efectos de propagación del escenario. De esta manera se ve la importancia de la repetición de la fase de entrenamiento previa a la puesta en marcha del sistema, ya que cabe la posibilidad que en la primera fase de entrenamiento no se obtuvieran los parámetros que modelan correctamente el escenario de localización.

Si se diera el caso que en la primera fase de entrenamiento no se obtienen los parámetros correctos y se dispone de otros parámetros que proporcionen mejores resultados, aunque no se hayan obtenido en el mismo entorno de propagación, se utilizarían éstos hasta que en alguna fase de entrenamiento posterior se obtuvieran unos parámetros mejores. Por ejemplo, si no se hubieran realizado las medidas que proporcionan los Parámetros **2** el mejor posicionamiento se obtendría con los Parámetros **7** (**Tabla 17**) y la diferencia entre el error de posicionamiento obtenido por cada uno es del 3.36% con lo que no habría una gran diferencia en la precisión media del posicionamiento obtenido con cada uno de los parámetros de propagación, todo y que estos parámetros (Parámetros **7**) no se hayan extraído del propio escenario de localización.

En la **Tabla 18** se representan las coordenadas obtenidas con el posicionamiento RSS utilizando los Parámetros **2** junto con las coordenadas GPS extraídas en el escenario de localización 3. En este caso tanto los resultados RSS como GPS son muy parecidos a los del escenario 2, obteniendo un error medio RSS de 5.17 m y un error medio GPS de 6.38 m.



Coordenadas	P1	P2	P3	P4	P5	P6	P7	P8	P9
<b>X_REAL</b>	5.57	2.5	0	2.5	2.5	2.5	0	2.5	5.57
<b>Y_REAL</b>	-10.23	-5.95	-5.35	-3.21	3.21	5.35	7.49	8.56	11.77
<b>X_RSS</b>	12.21	3.69	2.57	3.36	3.33	3.55	6.43	3.91	1.37
<b>Y_RSS</b>	-15.21	-1.9	-2.79	-0.76	3.17	3.71	1.99	18.07	6.24
<b>X_GPS</b>	-1.09	-2.07	-4.15	-1.01	3.09	4.37	6.48	8.78	12.96
<b>Y_GPS</b>	-2.02	-4.83	-1.76	0.32	2.92	-0.11	15.48	12.71	12.11

**Tabla 18.** Coordenadas GPS y RSS en escenario de localización 3.

Igual que en los dos casos anteriores los peores resultados del posicionamiento RSS se obtiene cuando se calcula la posición con motas anclas alejadas, en cuanto GPS se obtiene una medida de error medio que se aproxima en la mayoría de los casos a los errores cometidos en cada una de las posiciones, sólo difiere en las posiciones en las que se obtiene el error máximo (10 m) o mínimo (65 cm) obtenido. En la **Tabla 19** se puede comprobar estos errores, además de los resultados obtenidos con las técnicas de localización híbridas.

ERROR INSTANTANEO										ERROR
TÉCNICA	P1	P2	P3	P4	P5	P6	P7	P8	P9	PROMEDIO
RSS	8.3	4.21	3.63	2.59	0.84	1.95	8.46	9.61	6.9	5.17
GPS	10.57	4.71	5.48	4.98	0.65	5.77	10.29	7.53	7.4	6.38
Ponderación	5.41	3.6	3.12	2.62	0.8	2.53	7.28	8.94	5.14	4.38
*PM	1.31	3.08	2.98	3.03	0.75	3.47	6.45	8.08	3.14	3.59
BLUHE	1.15	3.07	3.05	3.13	0.74	3.64	6.48	7.97	3.01	<b>3.58</b>
Pond. Inst.	6.55	3.42	3.06	2.66	0.62	2.59	7.68	9.03	5.37	4.56
**PMI	1.43	3.07	2.92	2.74	0.71	2.58	6.45	7.58	2.99	<b>3.39</b>
BLUHE Inst.	1.14	3.07	2.97	2.89	0.72	2.89	6.48	7.71	2.99	3.43

\*PM: Ponderación Modificada; \*\*PMI: PM Instantánea.

**Tabla 19.** Error de posicionamiento para cada técnica de localización (E3).

En este caso las técnicas híbridas con parámetros de combinación promediados también brindan un correcto funcionamiento, debido como en el caso anterior a los errores medios obtenidos, tanto de GPS como de RSS, que son bastante próximos a los errores instantáneos cometidos en cada una de las posiciones por las localizaciones básicas. Si se analiza un poco más los resultados ponderados se puede observar que el buen funcionamiento de las técnicas híbridas promediadas no depende realmente de lo próximo que esté el error medio con el error cometido en cada una de las posiciones, si no que depende más bien en que la relación entre los errores medios se mantenga en los errores instantáneos. Esto se comprueba viendo por ejemplo que en P8 la relación entre errores instantáneos del posicionamiento básico es de 1.2 (RSS/GPS) mientras que la relación entre errores medios es de 0.8, obteniendo un error en las estimaciones híbridas mayor que el menor de los errores de las estimaciones básicas, que en el caso de la P8 es el posicionamiento GPS. En cambio en la posición P1 la primera relación es de 0.78, muy próxima a la relación entre errores medios, y el error obtenido por las estimaciones híbridas si que consigue disminuir el menor error del posicionamiento básico, en esta posición RSS.

En la **Tabla 19** se puede ver que los mejores resultados obtenidos por las técnicas híbridas promediadas se consiguen con las estimaciones BLUHE, llegando a una reducción del 30% del error RSS medio, que es el mínimo de las estimaciones básicas, mientras que para las técnicas híbridas instantáneas los mejores resultados se obtienen con la ponderación modificada consiguiendo una reducción del 34%. De esta manera, se observa que todo y obtener los mejores resultados con las mismas técnicas híbridas que en el escenario anterior, la diferencia en la reducción del error medio para las técnicas promediadas e instantáneas no es tan grande como anteriormente.

### **5.2.2 Escenarios interiores.**

En cuanto al posicionamiento en interiores, se encuentra una gran cantidad de trabajos y artículos relacionados con el posicionamiento en interiores utilizando medidas RSS en una red de sensores inalámbricos. En estos trabajos y estudios se comprueba la calidad del posicionamiento brindado por las WSN obteniendo unos resultados de posicionamiento local hasta incluso mejor que los proporcionados por los sistemas de posicionamiento global en exteriores como GPS. Por ello en este PFC no se ha prestado mucha atención al posicionamiento en interiores, ya que está demostrado que con WSN se puede obtener un posicionamiento más que aceptable. Aun así en el **ANEXO 3** se incluyen algunas medidas en un escenario interior para tener una idea de la precisión de este posicionamiento. Para el caso de las pruebas realizadas en este PFC se obtiene un error medio en el posicionamiento RSS en interiores del orden de 3 metros, obteniéndose los peores resultados cuando se obtiene la localización con motas alejadas o con las que no se tiene visión directa. En los casos interiores la probabilidad de encontrarse con motas sin visión directa es mayor que en el caso de los escenarios exteriores, por lo tanto en los casos interiores es importante el despliegue de las motas anclas de tal manera que en todo momento se pueda obtener la localización a partir de motas anclas relativamente cercanas o con la mayor visión directa posible.

Ejemplos de trabajos relacionados con el posicionamiento WSN en escenarios interiores se pueden encontrar en las siguientes referencias: [40,41]

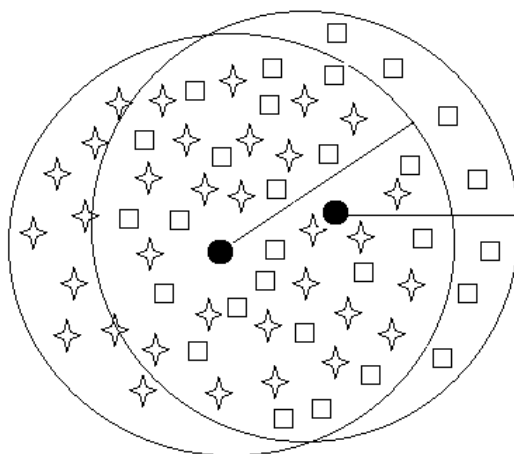
### 5.3 Conclusiones de los resultados híbridos.

En este capítulo se han tratado los resultados obtenidos del posicionamiento proporcionado por el sistema híbrido presentado a lo largo de la memoria. Estos resultados se basan mayormente en el posicionamiento obtenido en varios escenarios exteriores con el objetivo de poder obtener localización mediante GPS y la localización mediante RSS proporcionado por una WSN. Además, una pequeña parte de estos resultados se dedica a demostrar la capacidad de las WSN y las medidas RSS para brindar un posicionamiento en espacios interiores, donde la localización mediante GPS es imposible de adquirir.

Tras el análisis de todos estos resultados es posible extraer una serie de conclusiones que permitan definir las características que presenta el sistema de localización híbrido tratado a lo largo de la memoria. Entre estas conclusiones o características se encuentran las siguientes:

- Se ha podido comprobar la dependencia del posicionamiento basado en medidas de RSS con los parámetros de propagación utilizados para la estimación de distancias. Normalmente los parámetros de propagación extraídos en un escenario en particular son los que proporcionan los mejores resultados en el posicionamiento en ese escenario de localización, sin embargo se ha comprobado que hay casos en los que las medidas realizadas para extraer estos parámetros de propagación no son lo suficiente adecuadas para obtener unos parámetros que modelen correctamente el entorno de propagación. Por este motivo se define una **tabla de parámetros de propagación** que incluya varios parámetros de propagación obtenidos en una gran variedad de escenarios, con el objetivo de probar cada uno de estos parámetros para escoger aquellos que proporcionen el mejor posicionamiento en el escenario de localización en el que se utilizará el sistema de posicionamiento híbrido.
- Se ha comprobado tanto las prestaciones del posicionamiento obtenido a partir de medidas RSS como las prestaciones de la localización GPS brindada por la *sensor-board* utilizada. En todos los escenarios analizados se ha obtenido un error GPS medio mayor que el error medio obtenido con el posicionamiento RSS, con lo que queda presente la gran capacidad de posicionamiento que tienen las WSN en entornos locales (limitados en un espacio relativamente reducido) respecto la capacidad de GPS, que es más bien un sistema pensado para localización global (en todo el espacio terrestre). Ambos errores medios podrían reducirse tras la implementación final del sistema:

- La precisión del posicionamiento RSS disminuye a medida que la distancia con las motas anclas con las que se obtiene la localización aumenta. En el caso del PFC únicamente se disponía de 4 motas anclas, con lo que sólo en un pequeño rango de distancias se podía conseguir que al menos 3 motas anclas estuvieran a menos de 5 metros, por ello la mayoría de posiciones en las que se tomaban las medidas las motas anclas estaban suficientemente alejadas como la precisión del posicionamiento se viera afectada. Sin embargo, en la implementación final se podría desplegar el número de motas anclas necesarias para conseguir que en todo el rango del escenario de localización se obtuviera al menos 3 motas anclas en una distancia lo suficientemente corta como para no afectar el posicionamiento RSS.
- Un problema a la hora de la obtención del posicionamiento GPS es que dado su error medio las posiciones estimadas estarán dentro del círculo cuyo centro es la posición real y radio el error GPS medio. De esta manera, surge un problema cuando se pretende estimar posiciones consecutivas dentro del rango del error medio GPS ya que se darán el caso que la relación real entre las dos posiciones no se corresponda con la relación presente en las dos estimaciones, es decir, puede ser que realmente las posiciones se desplace hacia un sentido mientras que las estimaciones hacen que el desplazamiento sea en sentido contrario o en una dirección totalmente distinta (**Figura 42**). De esta manera, la precisión del posicionamiento GPS podría ser mejorada realizando el promedio de aquellas posiciones que se obtengan a partir de una misma posición y en el caso del desplazamiento del usuario realizar el promedio durante un cierto tiempo adecuado para que las posiciones reales se hayan separado lo suficiente como para estar fuera del rango del error GPS medio.
- Por el mismo motivo, sería conveniente utilizar una referencia lejana al escenario de localización con el objetivo de poder referenciar todas las coordenadas UTM a partir de unas coordenadas lo suficientemente lejos del sistema de localización para no tener problemas relacionados con el rango del error GPS medio.



**Figura 42.** Posicionamiento dentro del rango del error GPS.

- Por último se ha podido comprobar la capacidad de posicionamiento de las técnicas de localización híbridas propuestas en el **Capítulo 3**. Básicamente estas técnicas deberían permitir obtener un posicionamiento más preciso en media que el posicionamiento obtenido por cualquiera de las técnicas de localización básicas (RSS y GPS). Las técnicas originalmente propuestas utilizan parámetros de combinación híbrida promedios, es decir, se calcula un único parámetro de combinación a partir de los errores medios del posicionamiento RSS y GPS, sin embargo para comprobar la esencia de estas técnicas se utilizan unos parámetros de combinación híbrida instantáneos, obtenidos a partir de los errores cometidos en cada una de las posiciones a estimar por las técnicas básicas.
  - Con los parámetros promedios se ha comprobado que para obtener un correcto funcionamiento de las técnicas híbridas es necesario obtener unas buenas medidas del error medio cometido por las técnicas básicas. Se entiende por buena medida aquella que no difiere considerablemente con los errores reales cometidos en cada una de las posiciones. Obteniendo esta buena medida las técnicas híbridas funcionan correctamente siendo la técnica que mejores resultados proporciona la técnica **BLUHE**, obteniendo un error medio de 3.5 metros consiguiendo reducir el menor de los errores básicos hasta un 33%.
  - Con los parámetros instantáneos queda clara la capacidad de las técnicas híbridas en proporcionar unos errores instantáneos menores que el de las técnicas básicas, siendo en este caso la técnica que mejores resultados proporciona la **ponderación modificada**, obteniendo en este caso un error medio de todos los escenarios de 2.5 metros consiguiendo reducir por lo tanto el menor de los errores básicos hasta un 48%.

Con estos resultados la mayoría de los objetivos a conseguir con el presente PFC quedan concluidos ya que se ha obtenido tanto el posicionamiento RSS como el posicionamiento GPS a partir de una WSN para la posterior obtención del posicionamiento híbrido definido por el sistema de localización híbrido presentado en la memoria. Si las medidas necesarias para la definición de los parámetros híbridos se toman correctamente las estimaciones híbridas serán aceptables y válidas para el correcto funcionamiento del sistema, sin embargo si estas medidas no son las correctas el posicionamiento híbrido no sería aceptable por ello a continuación se pretende estudiar los errores en las distancias estimadas y su propagación en las posiciones obtenidas por las técnicas básicas, con el objetivo de obtener algún método de estimación de errores instantáneos.



## 6 Estimación del error instantáneo.

En el capítulo anterior se han mostrado los resultados del sistema de localización híbrido tratado en la presente memoria. Con los resultados obtenidos se comprueba que para obtener el funcionamiento deseado de las estimaciones híbridas en cualquier tipo de escenario es necesario la utilización de los errores instantáneos cometidos en cada una de las posiciones, para poder así obtener unos parámetros de combinación híbrida diferentes en cada una de las posiciones a localizar que permitan la selección de la mejor técnica básica en cada momento, no sólo eso si no que se ha visto que al combinar las estimaciones básicas se consigue una disminución del menor de los errores básicos. En su momento ya se comentó que actualmente no existe ningún método que permita la obtención de estos errores instantáneos en cada una de las posiciones que se estima la localización. Para el caso de GPS si que se ha podido estudiar a fondo su error en el posicionamiento permitiendo establecer una relación entre el error cometido y ciertos parámetros que puedan intervenir en el cálculo de la estimación de la posición, aun así los receptores GPS no permiten obtener esta relación y por lo tanto obtener el error cometido en una cierta posición. Para el caso del posicionamiento RSS la cosa está peor ya que por ahora no se ha realizado un estudio tan profundo como en GPS y por lo tanto no se puede extraer ningún tipo de relación entre la precisión del posicionamiento RSS y factores que afecten al cálculo de la posición, lo máximo que se puede obtener viendo los resultados obtenidos del posicionamiento RSS es que la precisión de su localización disminuye a medida que la distancia con las motas anclas aumenta.

Dada la necesidad del conocimiento de los errores instantáneos de las técnicas básicas para el correcto funcionamiento de las técnicas híbridas, en el presente capítulo se pretende:

- Exponer los factores que afectan al error del posicionamiento obtenido por GPS de tal manera que se pueda realizar una analogía en el error del posicionamiento en RSS.
- Estudiar el error cometido en la estimación de las distancias a partir de las medidas RSS y los parámetros de propagación utilizados, con el objetivo de obtener un resultado teórico que explique la disminución de la precisión del posicionamiento RSS a medida que la distancia aumenta.
- Estudiar el cálculo del posicionamiento con el objetivo de obtener una relación entre el error de posicionamiento, las distancias estimadas y las coordenadas obtenidas.

El estudio del capítulo se basa principalmente en el posicionamiento RSS de tal manera que a partir de las medidas disponibles en el capítulo anterior se puedan comprobar los métodos de estimación de errores propuestos a continuación. Los conceptos relacionados con GPS están más que comprobados y aceptados generalmente.



## 6.1 Error del posicionamiento GPS.

La precisión con la que un usuario de GPS puede determinar su posición depende de una complicada interacción de varios factores. En general, la precisión del posicionamiento GPS depende de la calidad de las medidas de distancia y fase de la portadora además de la información del mensaje de navegación transmitido por los satélites. La calidad de las medidas de distancia, denominadas *pseudorange*, se representa con el denominado UERE (*User-Equivalent Range Error*) que para un satélite dado se define como la suma (estadística) de las contribuciones de cada una de las fuentes de error asociadas con el satélite. Normalmente, las fuentes de error se consideran independientes con lo que el UERE resultante de un satélite se aproxima como una variable aleatoria Gaussiana de media nula y cuya varianza es la suma de las varianzas de cada una de las fuentes de error. Habitualmente, el UERE se supone independiente e idénticamente distribuido de satélite a satélite.

La precisión de la posición determinada por GPS no únicamente depende de la calidad del *pseudorange*, sino que depende del producto de un factor de geometría y un factor de error de *pseudorange* o medidas de distancia. Bajo ciertas condiciones, el factor de error relacionado con las medidas de distancia se obtiene con el UERE de los satélites, mientras que el factor geométrico expresa el efecto en el error de la geometría relativa satélite/usuario generalmente conocido como GDOP (*Geometric Dilution Of Precision*). Con estas definiciones se puede expresar el error del posicionamiento GPS ( $e_{GPS}$ ) con la siguiente fórmula:

$$e_{GPS} = GDOP \cdot UERE \quad (26)$$

Entre las fuentes de error GPS que influyen en el UERE se encuentran:

- Error del reloj del satélite.
- Error de efemérides (estimación de la posición de los satélites).
- Efectos relacionados con la teoría de la relatividad de Einstein.
- Efectos atmosféricos.
- Efectos de *shadowing* y *multipath*.

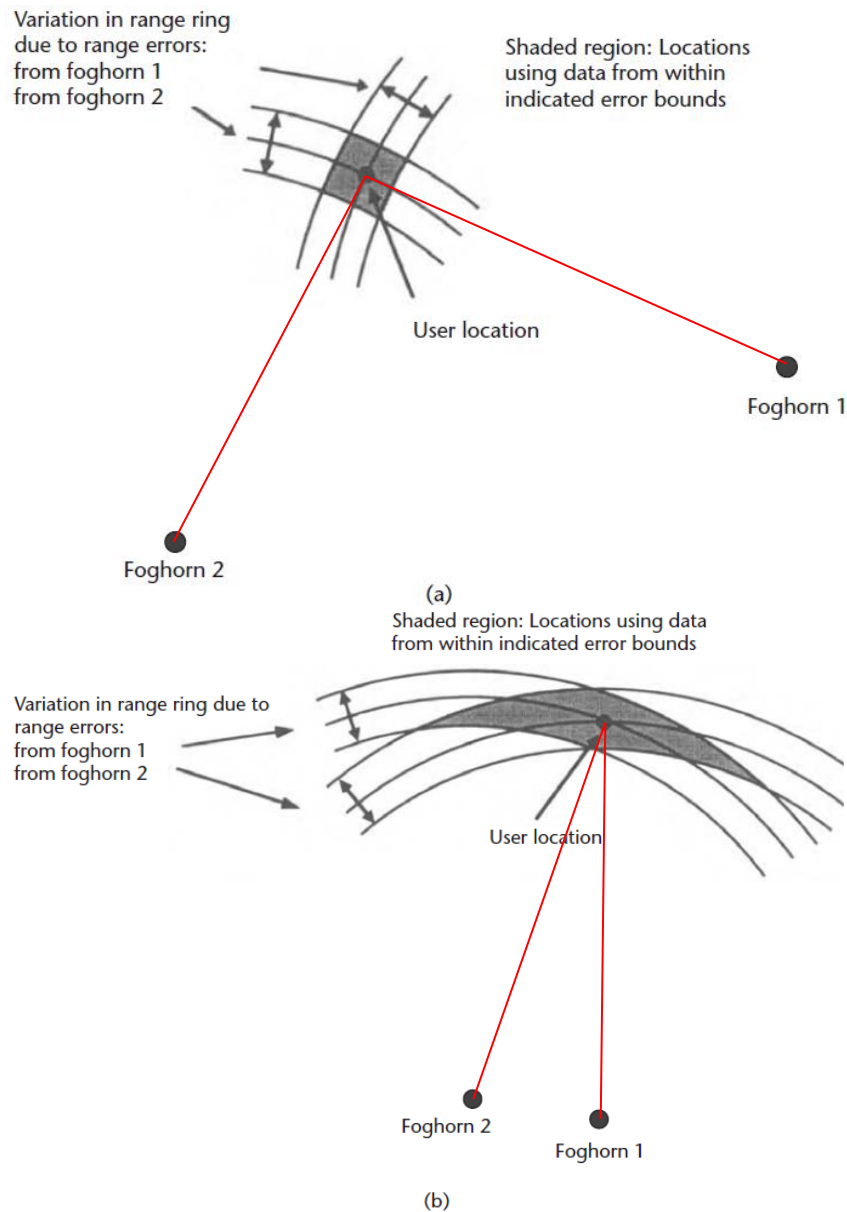
Como se supone que las fuentes de error son variables aleatorias independientes, el error total en la medida de distancia en GPS (UERE) se obtiene a partir de la combinación de los errores introducidos por las fuentes mencionadas, de tal manera que al ser fuentes independientes la varianza del UERE será igual a la suma de varianzas de los errores involucrados en la medida de distancia. En la **Figura 43** se muestra un ejemplo de UERE, donde  $1\sigma$  Error (m) representa al error de las diferentes fuentes de error y es equivalente a la raíz de la varianza de cada una de las fuentes de error.

<i>Segment Source</i>	<i>Error Source</i>	<i>1<math>\sigma</math> Error (m)</i>
<b>Space/control</b>	Broadcast clock	1.1
	Broadcast ephemeris	0.8
<b>User</b>	Residual ionospheric delay	0.1
	Residual tropospheric delay	0.2
	Receiver noise and resolution	0.1
	Multipath	0.2
<b>System UERE</b>	<b>Total (RSS)</b>	<b>1.4</b>

\*RSS: *Root-Squared-Sum* =  $\sqrt{x_1^2 + x_2^2 + \dots + x_N^2}$

**Figura 43.** Ejemplo de UERE extraído de [42].

Para introducir el concepto de dilución de precisión (DOP, *Dilution Of Precision*) se considera un escenario de localización con dos estaciones bases de referencia (*Foghorn* 1 y 2) y un usuario que realiza medidas TOA para determinar la distancia que le separa de ambas referencias y poder describir las dos circunferencias que se cruzarán en su posición. Con la presencia de errores de medida, las circunferencias utilizadas pueden ser erróneas y resultar en una posición estimada errónea. El concepto de DOP es la idea que el error de posicionamiento debido a los errores de medida depende de la geometría relativa entre usuario y estaciones bases de referencia. Gráficamente esta idea se ilustra en la **Figura 44** donde se representan dos geometrías para el escenario de localización considerado. En la **Figura 44 (a)** las referencias están situadas aproximadamente en ángulos rectos respecto la localización del usuario, mientras que en la **Figura 44 (b)** el ángulo entre las referencias visto desde el usuario es mucho menor. En ambos casos se incluyen porciones de las circunferencias sin error, que se cortan en la posición del usuario, y los segmentos de circunferencia que representan la variación de las circunferencias reales (sin error) debido a los errores de medidas. El rango de error introducido es el mismo para ambas figuras, siendo la región sombreada el conjunto de posiciones que se pueden obtener utilizando las medidas de distancia con el margen de error indicado. Se observa que la precisión de la localización calculada en ambos casos es muy diferente de tal manera que, con la misma variación del error en las medidas, la geometría en la **Figura 44 (b)** proporciona un error considerablemente mayor en la localización del usuario que la geometría de la **Figura 44 (a)**, siendo evidente comparando las regiones sombreadas de la figura. De la geometría en la **Figura 44 (b)** se diría que tiene una mayor dilución de precisión que la geometría en la **Figura 44 (a)**, de tal manera que a mayor DOP se obtiene una menor precisión o equivalentemente una geometría que proporciona unos errores mayores.



**Figura 44.** GDOP: (a) geometría con bajo DOP y (b) con alto DOP [43].

Para una explicación detallada de cada una de las fuentes de error que afectan a la medida de distancias así como del efecto que produce en la estimación de la posición GPS y todo lo relacionado con el GDOP y su cálculo consultar [44]. Principalmente, lo que se quiere dejar claro con esta explicación es que el error GPS, tal y como se expresa en la ecuación (26), depende de dos factores básicos:

- **Geometría usuario/referencias:** que influye en el impacto de los errores de distancia en los errores de la posición calculada.
- **Entorno GPS:** que influye directamente en los errores de la medida de distancia que se traducen en un error mayormente constante y común en cada una de las posiciones calculadas (dentro de un mismo escenario de localización).

## 6.2 Error del posicionamiento RSS.

En el apartado anterior, se ha comprobado que el error en el posicionamiento GPS depende principalmente de dos factores, uno de ellos indica el efecto que tiene el error de las medidas de distancias en el error del posicionamiento debido a la geometría entre los satélites y el usuario (GDOP). El otro factor del que depende el error GPS representa los efectos del entorno o del sistema GPS que hacen que las medidas de distancia sean erróneas (UERE). A continuación, se pretende extraer una dependencia análoga entre el error en el posicionamiento RSS y dos factores que dependan de los mismos efectos que en el caso del error GPS.

Experimentalmente se ha comprobado que el posicionamiento obtenido con RSS empeora a medida que la distancia entre el terminal a localizar y las estaciones base con las que se posiciona aumenta. La causa de la pérdida de precisión en el posicionamiento a distancias grandes es debido a que las estimaciones de distancia obtenidas con las medidas RSS pierden precisión a medida que la distancia a estimar aumenta. Por ello, en primer lugar, se estudia la dependencia con la distancia de las estimaciones obtenidas a partir de las medidas RSS.

En el **Capítulo 2** se definió el modelo teórico que describe la potencia recibida en función de la distancia a partir de los parámetros de propagación que describen el entorno ( $n, \gamma$ ). Este modelo teórico puede representarse a partir de la siguiente ecuación:

$$P_r(dB) = n - 10 \cdot \gamma \cdot \log_{10} d - v \quad (27)$$

Donde  $d$  es la distancia a la que se recibe la potencia  $P_r$  y  $v$  representa el ruido en la medida de potencia y provoca las variaciones de la potencia recibida en una misma distancia. Este ruido es de media nula y varianza  $\sigma_v^2$ , es decir,  $v \sim N(0, \sigma_v^2)$  (dB). A partir de las potencias recibidas se obtienen las estimaciones de distancia con la siguiente expresión:

$$\hat{d} = 10^{\frac{n - P_r(dB)}{10 \cdot \gamma}} \quad (28)$$

Suponiendo que los parámetros de propagación  $n$  y  $\gamma$  son los correctos, tomando el logaritmo de (28) y sustituyendo la expresión (27), se obtiene que:

$$\log_{10} \hat{d} = \frac{n - P_r}{10 \cdot \gamma} = \log_{10} d + \frac{v}{10 \cdot \gamma} = \log_{10} d + e \quad (29)$$

Donde  $e = \frac{v}{10 \cdot \gamma}$  es una variable aleatoria relacionada con el error cometido en la distancia, concretamente representa el error cometido en el logaritmo de la distancia ( $\log_{10} d$ ) y sigue una distribución normal tal que  $e \sim N\left(0, \frac{\sigma_v^2}{100 \cdot \gamma^2}\right)$ . Ahora se tiene una descripción del error cometido en la estimación del logaritmo de la distancia, sin embargo interesa obtener la descripción del error cometido directamente en la distancia.

Para ello se elimina el logaritmo de la expresión (29) y teniendo en cuenta que  $e = \log_{10} e'$ , se obtiene la siguiente expresión para la distancia estimada en función de la distancia real y  $e'$ :

$$\hat{d} = 10^{(\log_{10} d + e)} = 10^{\log_{10}(d \cdot e')} = d \cdot e'$$

Que permite, tras una simple manipulación matemática, obtener la dependencia de las estimaciones de distancias basadas en RSS con la distancia real:

$$\hat{d} = d \cdot (1 - 1 + e') = d + e_{\hat{d}} \quad (30)$$

Siendo  $e_{\hat{d}}$  el error cometido en las estimaciones de distancias y que puede expresarse como sigue:

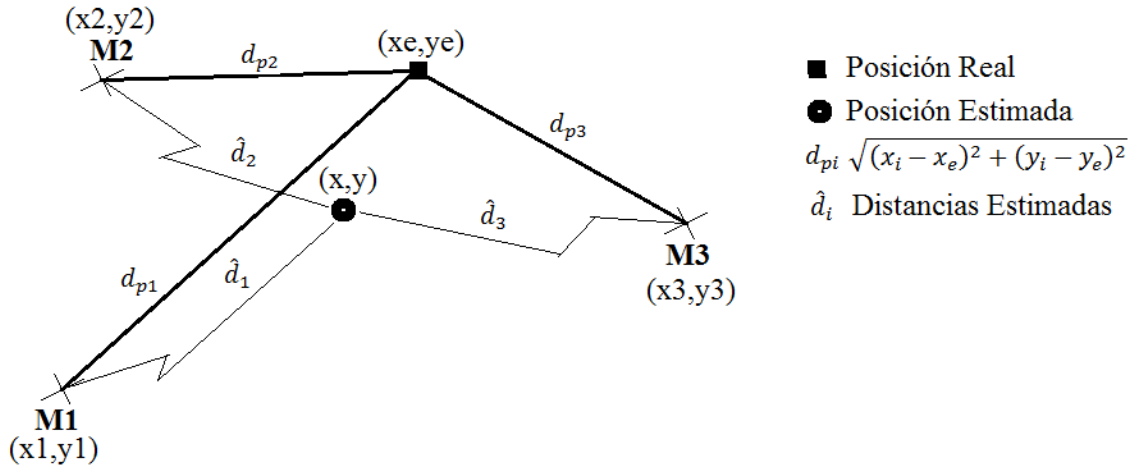
$$e_{\hat{d}} = (e' - 1) \cdot d = e'' \cdot d \quad (31)$$

Donde queda presente la dependencia del error en las estimaciones de distancias a partir de medidas RSS con la distancia real. Además, igual que en el caso del error GPS, interviene otro factor que depende básicamente del entorno ( $\sigma_v^2$  y  $\gamma$ ). Por lo tanto, equivalentemente al caso del error cometido en el posicionamiento GPS, el error cometido en las estimaciones de distancia basadas en la medida de RSS depende de dos factores, uno de ellos representa los efectos del entorno de propagación ( $e''$ ) y el otro la dependencia del error con la distancia real.

Una vez analizada la dependencia de las estimaciones de distancia con la distancia real, se procede al estudio del error en el posicionamiento con medidas RSS. En el **Capítulo 2** ya se comentó que debido a los errores cometidos en las estimaciones de distancia entre el terminal a posicionar y las estaciones base con las que se localiza, se puede dar el caso que las circunferencias descritas por las estimaciones de distancia no se crucen en un mismo punto o que no se crucen en ningún punto. Por ello, para la determinación de la posición estimada se utiliza el criterio del error LS (*Least Squares*) para estimar la posición cuyas coordenadas sean las que minimizan la siguiente expresión:

$$F = \sum_{i=1}^N \left( \sqrt{(x_i - x_e)^2 + (y_i - y_e)^2} - \hat{d}_i \right)^2 = \sum_{i=1}^3 (d_{pi} - \hat{d}_i)^2 \quad (32)$$

Donde en el caso del presente PFC N=3 siendo el número de motas anclas con las que se obtiene el posicionamiento RSS. Las distancias  $d_{pi}$ 's son las distancias entre la posición estimada y las 3 motas con las que se obtiene la localización, mientras que  $\hat{d}_i$  es la distancia estimada a partir de las medidas RSS de cada una de las motas anclas con las que se posiciona (**Figura 45**).



**Figura 45.** Significado geométrico de las distancias  $d_{pi}$ .

Es evidente que cuando las distancias se estimen correctamente la diferencia de la expresión (32) será nula y la posición estimada será la posición real, no obstante es posible que las distancias estimadas generen unas circunferencias que se cruzan en un punto cuyas coordenadas provocan también un diferencia nula, por lo tanto no es suficiente con la expresión (32) para obtener el error del posicionamiento RSS sino que se necesita un factor adicional que represente el efecto del error de las estimaciones de distancia en las posiciones estimadas. Haciendo la analogía con el error en el posicionamiento GPS y en las estimaciones de distancias con medidas RSS, se puede representar el error del posicionamiento RSS como sigue:

$$e_{RSS} = F \cdot r \quad (33)$$

Donde de  $F$  es el resultado obtenido con la expresión (32) y representa el efecto que tienen las distancias estimadas en el cálculo de la posición (el análogo al GDOP), mientras que  $r$  sería el factor que representa el efecto de obtener estimaciones de distancias erróneas debido al entrono del escenario de localización. En la expresión (31) se observa que el error en las distancias estimadas es directamente proporcional a la distancia real, con lo que se puede suponer que  $r$  es lineal con la distancia y que describe una recta tal que:

$$r = k \cdot d + r_0$$

Con esta descripción de  $r$  se podría obtener los parámetros  $k$  y  $r_0$  con el fin de calcular el valor de  $r$  para cualquier distancia a la que se tomen las medidas RSS que permitan el posterior cálculo del posicionamiento. De esta manera, obteniendo el valor de  $F$  para cada una de las posiciones estimadas y haciendo el producto con el valor de  $r$  obtenido se podría calcular el valor del error cometido en la posición estimada. Para el cálculo de  $F$  se necesitan medidas que se poseen una vez calculada la estimación de la posición, sin embargo la recta que describe  $r$  utiliza la distancia real que es una medida de la que no se dispone, con lo que en vez de utilizar la distancia real se puede utilizar la distancia estimada y obtener los parámetros que describen la recta con el mismo procedimiento que el utilizado en el **Capítulo 2** para la obtención de los parámetros de propagación. Este método, que está basado en LS, se denomina “*curve fitting*”.

$$\hat{r} = k \cdot \hat{d} + r_0 \quad (34)$$

Para obtener el valor de los parámetros que representan la recta que describe (34) en los distintos escenarios de localización se realizan medidas empíricas del error cometido en el posicionamiento RSS en varias posiciones, una vez se dispone de las medidas del error RSS se obtiene el valor de  $F$  a través de la expresión (32) para cada una de las posiciones en las que se han tomado medidas y se obtiene el cociente entre el error RSS y el valor de  $F$  obtenido en cada una de las posiciones, definiendo el vector  $\mathbf{z}$  como el conjunto de cocientes obtenidos en cada una de las posiciones, que equivalen al valor de  $r$  en cada una de las posiciones:

$$\mathbf{z} = \begin{bmatrix} e_{RSS_1}/F_1 \\ \vdots \\ e_{RSS_N}/F_N \end{bmatrix} = \begin{bmatrix} r_1 \\ \vdots \\ r_N \end{bmatrix} = \mathbf{r} \quad (35)$$

La expresión en (34) representa un único valor del factor  $r$  obtenido con la distancia estimada en cada una de las posiciones ( $\hat{r}$ ), pudiendo obtener el siguiente sistema matricial que considera más de un valor de  $\hat{r}$  obtenido:

$$\hat{\mathbf{r}} = \begin{bmatrix} \hat{r}_1 \\ \vdots \\ \hat{r}_N \end{bmatrix} = \begin{bmatrix} k \cdot \hat{d}_1 + r_0 \\ \vdots \\ k \cdot \hat{d}_N + r_0 \end{bmatrix} = \begin{bmatrix} \hat{d}_1 & 1 \\ \vdots & \vdots \\ \hat{d}_N & 1 \end{bmatrix} \cdot \begin{bmatrix} k \\ r_0 \end{bmatrix} = \mathbf{R} \cdot \mathbf{m} \quad (36)$$

Así, obteniendo el vector  $\mathbf{z}$  tal y como se define en (35) a partir de medidas experimentales y con la representación en (36) se obtiene **(2.2.2/Parametrización del escenario)**:

$$\hat{\mathbf{m}} = \begin{bmatrix} \hat{k} \\ \hat{r}_0 \end{bmatrix} = (\mathbf{R}^H \cdot \mathbf{R})^{-1} \cdot \mathbf{R}^H \cdot \mathbf{z} \quad (37)$$

De esta manera, se puede estimar el error en cada una de las posiciones obtenidas con la localización RSS a partir de la siguiente expresión:

$$\hat{e}_{RSS} = F \cdot \hat{r} = F \cdot (\hat{k} \cdot \hat{d} + \hat{r}_0) \quad (38)$$

Siendo  $F$  el factor obtenido con la expresión (32),  $\hat{k}$  y  $\hat{r}_0$  los parámetros estimados mediante (37) para la representación de la recta que describe  $\hat{r}$  y  $\hat{d}$  representa la distancia estimada que se utiliza para la obtención del posicionamiento, sin embargo para el posicionamiento se utilizan tres estimaciones de distancia, una para cada una de las motas anclas con las que se posiciona, por lo tanto a la hora de obtener tanto  $\hat{\mathbf{m}}$  con la expresión (37) como  $\hat{e}_{RSS}$  en la expresión (38) se utiliza el promedio de las tres distancias estimadas en representación de  $\hat{d}$ .

### 6.3 Resultados experimentales.

Se ha visto que existe la posibilidad de obtener medidas de la precisión o del error cometido en el posicionamiento, tanto para la localización GPS como para el posicionamiento basado en medidas RSS, a partir de dos factores relacionados con el impacto de las medidas erróneas de distancia y los efectos del entorno o del sistema que hacen que se obtengan medidas de distancia imprecisas. Para el caso de GPS, ya se ha mencionado varias ocasiones, se disponen receptores con capacidad de calcular el GDOP y proporcionarlo al usuario y poder saber así si la posición obtenida por el receptor puede ser errónea debido a la geometría entre los satélites y el usuario. Sin embargo, se ha visto en el apartado de dicado a GPS que su error no sólo depende de la geometría sino que están los factores relacionados con el ambiente o el UERE que pueden provocar una estimación de posición errónea todo y que se posea una buena geometría entre usuario y satélites (GDOP). En el caso del receptor GPS disponible para las motas de la WSN formada en el PFC permite obtener el HDOP (DOP Horizontal), pero como el resto de receptores el valor del UERE en cada una de las posiciones en las que se obtiene posicionamiento GPS no se proporciona.

Para el caso del posicionamiento RSS, al contrario que GPS, si que hay la posibilidad de obtener los dos parámetros que representan el error cometido en cada una de las posiciones.  $F$  se obtiene a partir de las distancias estimadas y las distancias entre la posición obtenida en la localización y las motas anclas con las que se ha localizado ( $d_{pi}$ ), mientras que el factor restante se calcula mediante el promedio de las distancias estimadas de las motas anclas con las que se posiciona y unos parámetros obtenidos a partir de resultados experimentales (“*curve fitting*”). Las expresiones para obtener estos factores y los parámetros necesarios se han presentado en el apartado dedicado al error con RSS, sin embargo tras realizar las pruebas con resultados experimentales se observa que las expresiones mostradas deben ser modificadas básicamente para obtener valores de error y no de varianza. En el **Capítulo 3** ya se comentó la relación entre la varianza y el error del posicionamiento, siendo la varianza el error al cuadrado o equivalentemente el error del posicionamiento es igual a la raíz de la varianza obtenida en la estimación de la posición, por ello las expresiones del apartado anterior se ven modificadas como siguen:

$$F = \frac{1}{N} \cdot \sqrt{\sum_{i=1}^N \left( \sqrt{(x_i - x_e)^2 + (y_i - y_e)^2} - \hat{d}_i \right)^2} = \frac{1}{3} \cdot \sqrt{\sum_{i=1}^3 (d_{pi} - \hat{d}_i)^2} \quad (38)$$

$$\hat{e}_{RSS} = F \cdot \sqrt{\hat{r}} = F \cdot \sqrt{(\hat{k} \cdot \hat{d} + \hat{r}_0)}$$

Para el cálculo de la expresión (37) con la que se obtienen los parámetros que representan a  $\hat{r}$  se utiliza el vector  $\mathbf{z}$  definido en (35) pero utilizando el factor  $F$  definido en (38).



### 6.3.1 Estimaciones del error instantáneo RSS.

A continuación, se analizan los datos obtenidos en los escenarios de localización tratados en el capítulo de resultados híbridos (**Capítulo 5**) con el objetivo de obtener estimaciones de los errores cometidos por el posicionamiento RSS en cada una de las posiciones en las que se han tomado medidas. En concreto se muestran las distancias estimadas en cada una de las posiciones con las motas anclas correspondientes a las distancias medidas, así como la distancia real y la distancia entre la posición obtenida y las motas anclas con las que se obtiene la posición. Con estas distancias se calcula el factor  $F$  a partir de la expresión en (38) con el que se obtienen los parámetros que describen  $\hat{r}$  con (37) para finalmente obtener la estimación del error del posicionamiento RSS en cada una de las posiciones con la expresión del error en (38).

#### - Escenario de localización 1 (E1).

Antes de poder calcular los factores necesarios para la estimación del error de posicionamiento se deben obtener las distancias estimadas entre la mota a posicionar y las tres motas ancla con las que se localiza, además de las distancias entre la posición obtenida por la localización RSS y las tres motas ancla con las que se posiciona, denominadas **distancias posición**. Estas distancias junto con las distancias reales y las motas anclas utilizadas en la localización en cada una de las posiciones se detallan en la **Tabla 20**.

DISTANCIA ESTIMADA					DISTANCIA POSICIÓN				
P1	P2	P3	P4	P5	P1	P2	P3	P4	P5
6.90	2.98	4.58	5.69	9.70	6.73	3.45	4.67	5.45	12.74
7.92	3.56	5.62	5.62	15.70	7.77	3.97	5.70	5.37	17.74
3.57	2.84	1.97	2.06	8.06	3.23	3.33	2.17	1.23	11.54
MOTAS ANCLA					DISTANCIA REAL				
P1	P2	P3	P4	P5	P1	P2	P3	P4	P5
2	2	4	4	3	6.73	3.21	4.88	8.61	10.30
3	3	2	2	1	8.51	4.17	5.37	9.44	14.18
1	1	3	3	4	4.17	3.52	1.80	3.28	4.31

**Tabla 20.** Distancias y motas anclas del posicionamiento en E1.

Con las distancias estimadas y las distancias posición es posible obtener el factor  $F$  necesario en cada una de las posiciones, resultando en los valores indicados en la **Tabla 21** entre los que se observa que el valor máximo se corresponde con la posición donde la diferencia entre las distancias estimadas y las distancias posición es mayor, mientras que el valor mínimo se obtiene en la posición donde la diferencia entre estas distancias es menor, que es justo el efecto que debe representar el factor  $F$  definido en los apartados previos.

ERROR RSS					$F$				
P1	P2	P3	P4	P5	P1	P2	P3	P4	P5
0.99	0.27	0.56	4.07	9.68	0.23	0.46	0.13	0.52	2.91

**Tabla 21.** Factor  $F$  obtenido en E1.

Por otro lado, en la **Tabla 21** se representa el error cometido con el posicionamiento RSS en cada una de las posiciones, observando la necesidad de un factor adicional que multiplicado al factor  $F$  proporcione una estimación del error RSS precisa. Este factor es necesario por el hecho que los valores de  $F$  no se corresponden con los valores del error del posicionamiento RSS y que se pueden dar casos en los que este factor sea pequeño y el error del posicionamiento en esa posición sea grande, esto se observa en P4 donde  $F$  toma un valor relativamente pequeño (0.52) y el error en esa posición es considerablemente grande (4.07). Esto deja presente la posibilidad de que las distancias estimadas describan unas circunferencias que proporcionan una posición cuyas distancias hacia las motas anclas con las que se localiza sean parecidas a las distancias estimadas, o equivalentemente  $F$  próximo a 0, pero la posición estimada es errónea debido a otros factores que  $F$  no considera. Además, se puede comprobar que este factor debe ser proporcional a la distancia, tal y como se ha definido el parámetro  $r$  anteriormente, viendo en la **Tabla 20** y **Tabla 21** que el valor multiplicativo adicional debe ser mayor a medida que la distancia real aumenta.

Definido  $r$  como en el apartado anterior, en la **Tabla 22** se muestra tanto el valor teórico necesario ( $z = r$ ) como el valor estimado a partir de los parámetros obtenidos con la expresión (6.12), donde se puede observar como en cierto modo  $z$  aumenta a medida que las distancias estimadas promedio aumenta pero que hay casos en los que para distancias promedio mayores se tiene un valor de  $z$  menor que para posiciones en las que se tiene distancias promedio menores (P1 y P4) o se obtiene un  $z$  considerablemente diferente con valores de distancia promedio parecidos (P3 y P4). El problema de esto es debido a que el valor de  $z$  mostrado en la **Tabla 22** se obtiene a partir del promedio de las distancias estimadas entre motas anclas y desconocida en cada una de las posiciones, cuyo valor puede diferir del valor promedio de las distancias reales, que es el caso de P4 donde el promedio de las distancias reales es de 7.11 mientras que el promedio de las estimaciones es de 4.46. Además, hay otra incoherencia en la posición P4 y P5 en el sentido que el factor de  $z$  es mayor en P4 cuando tanto la distancia real promedio como la estimada son mayores en P5. Todo y estos efectos adicionales, de media se observa cómo el factor  $r = z$  aumenta a medida que la distancia entre las motas anclas participantes en la localización y la mota a posicionar aumenta, la variación de esta relación es debida por algún otro efecto participante en la localización que no se ha tenido en cuenta en la definición del parámetro  $r$ .

FACTOR $r$	P1	P2	P3	P4	P5
$z$	4.23	0.58	4.22	7.79	3.32
$\sqrt{\hat{r}}$	2.0076	2.0057	2.0063	2.0066	2.0107
Dist. Est. Promedio (m)	6.13	3.13	4.06	4.46	11.15
Dist. Promedio (m)	6.47	3.63	4.02	7.11	9.60

**Tabla 22.** Factor  $r$  necesario y obtenido en las posiciones de E1.

Con los valores de  $r$  necesarios ( $z$ ) y las distancias estimadas promedio en cada una de las posiciones se obtienen unos parámetros que describen un factor  $r$  aproximadamente constante con la distancia estimada de valor próximo a  $2(\sqrt{\hat{r}})$  (**Tabla 22**). Ahora que se dispone tanto del factor  $F$  como del factor  $r$  estimado es posible obtener la estimación del error de posicionamiento RSS propuesto en el apartado anterior (6.14) para cada una de las posiciones del escenario de localización 1 (**Tabla 23**).

ERROR (m)	P1	P2	P3	P4	P5	PROMEDIO
RSS	0.99	0.27	0.56	4.07	9.68	3.11
RSS EST.	0.47	0.92	0.27	1.05	5.86	1.71
<b>DIFERENCIA</b>	<b>0.52</b>	<b>0.65</b>	<b>0.29</b>	<b>3.02</b>	<b>3.82</b>	<b>1.66</b>

**Tabla 23.** Error RSS estimado en E1.

En la **Tabla 23** se presentan los resultados de la estimación del error propuesta comparada con el error real en cada una de las posiciones en las que se toman medidas. Se observa que para los valores pequeños del error real se obtienen estimaciones considerablemente buenas, sin embargo para las dos últimas posiciones donde el error del posicionamiento es grande no se obtienen unas estimaciones lo suficientemente grandes, esto hace que el valor promedio de las estimaciones del error en cada una de las posiciones dé un valor un 45% menor que el error promedio real. Todo y esta disminución del error promedio estimado respecto el real, la diferencia media entre los valores estimados y los reales en cada una de las posiciones es de 1.66 metros que puede ser lo suficientemente pequeña como para que las estimaciones híbridas tengan un funcionamiento satisfactorio, por eso posteriormente se dedicará un apartado para comprobar los resultados de las estimaciones híbridas utilizando estas estimaciones de los errores instantáneos del posicionamiento RSS en cada una de las posiciones.

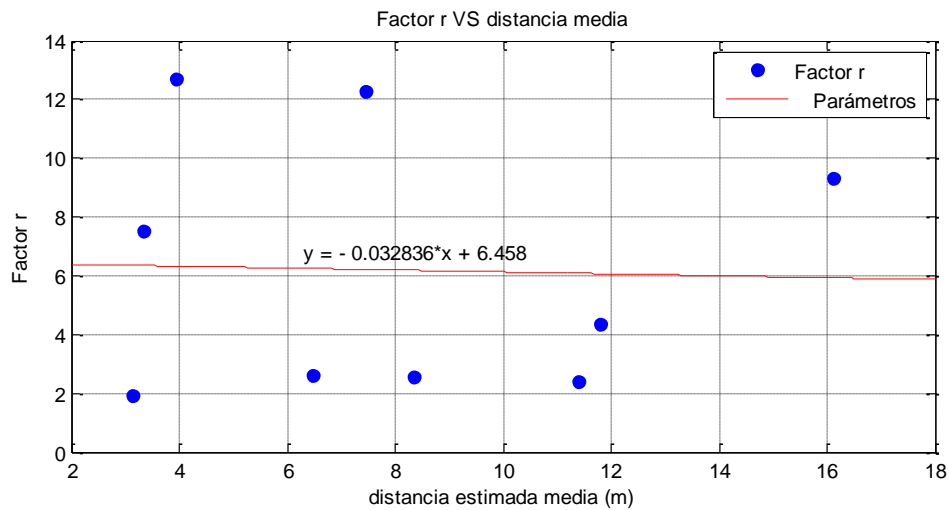
- **Escenario de localización 2 (E2).**

Igual que en el escenario anterior, antes de poder calcular ningún factor es necesario disponer de las distancias estimadas y las distancias posición (distancias entre motas anclas y posición estimada), por ello en la **Tabla 24** se muestran estas distancias junto con las reales y las motas anclas utilizadas para localizar en cada una de las posiciones. En el caso anterior no se ha comentado nada sobre la relación entre la distancia real y la estimada, pero igual que en este caso se puede observar que generalmente las distancias reales grandes se estiman con menor precisión que las distancias pequeñas. Por ejemplo, en P1 para la distancia real de 16 metros se estima una distancia de 12 metros (4 metros de diferencia) mientras que en P6 para la distancia real de 2.92 m se estima una distancia de 2.55 m (38 cm de diferencia). Este comportamiento es general, pero eso no significa que haya casos en los que para distancias mayores el error sea menor, observando la **Tabla 24** para el caso de P3 se estima una distancia de 10.4 m para una distancia real de 9.86 (18 cm) mientras que se ha visto que en P6 para una distancia real de unos 3 metros se obtiene una diferencia respecto la distancia estimada de 38 cm. Este efecto es el efecto obtenido teóricamente en el apartado anterior sobre el error en la distancia estimada con medidas RSS, es decir, el error en la distancia estimada aumenta a medida que aumenta la distancia real pero al ser el error una variable aleatoria su varianza puede provocar que este error en ciertas circunstancias disminuya respecto el error obtenido en distancias menores.

MOTAS ANCLA								
P1	P2	P3	P4	P5	P6	P7	P8	P9
2	2	4	1	2	4	4	4	2
3	1	2	3	1	1	1	1	4
1	3	3	4	3	2	2	2	1
DISTANCIA ESTIMADA (m)								
11.93	9.67	8.32	2.06	4.60	2.63	7.11	11.79	13.81
10.17	6.10	10.40	2.59	3.68	4.84	7.91	14.70	15.35
12.05	9.23	3.63	4.72	3.57	2.55	4.42	8.92	19.15
DISTANCIA POSICIÓN (m)								
7.19	8.10	8.13	2.82	4.45	3.02	5.76	13.54	14.23
3.53	2.95	10.58	3.23	3.43	5.05	6.73	16.31	15.83
7.63	7.22	3.03	5.11	3.30	2.94	1.43	11.11	19.51
DISTANCIA REAL (m)								
15.91	13.02	10.02	2.50	6.03	3.42	3.74	10.50	10.50
9.60	7.89	9.86	3.07	6.33	7.25	11.57	16.75	11.89
11.10	8.09	5.23	6.12	3	2.92	6.50	11.89	15.80

**Tabla 24.** Distancias y motas anclas del posicionamiento en E2.

Con las distancias disponibles en la **Tabla 24** se puede obtener el factor  $F$  necesario en cada una de las posiciones para poder obtener el error del posicionamiento RSS, el valor obtenido en cada una de las posiciones de medida se representa en la **Tabla 25** donde igual que en el caso anterior se observa que depende de la coincidencia entre las distancias estimadas y las distancia posición, de tal manera que si coinciden el factor es 0 y si no coinciden el factor aumentará a medida que la coincidencia sea menor.



**Figura 46.** Factor  $r$  necesario y parámetros obtenidos para calcularlo.

$F$								
P1	P2	P3	P4	P5	P6	P7	P8	P9
5.36	2.34	0.38	0.61	0.23	0.34	2.01	1.87	0.42

**Tabla 25.** Factor  $F$  obtenido en E2.

Estos valores de  $F$  permiten obtener el factor  $r$  necesario y el estimado a partir del método propuesto en cada una de las posiciones de medida. Estos valores serán los que multiplicados por los valores del factor  $F$  obtenidos proporcionen la estimación del error en el posicionamiento RSS, concretamente la expresión en (6.14) del error se realiza la multiplicación de  $F$  con la raíz del factor  $r$  que es precisamente lo que se representa en la **Tabla 26**. Los valores teóricos necesarios del factor  $r$  y los estimados se muestran en la **Tabla 26** con los que a partir de las distancias estimadas medias correspondientes se obtiene la representación en la **Figura 45**, donde se representan los valores necesarios del factor  $r$  en función de la distancia media estimada (el promedio de las 3 distancias estimadas para cada una de las motas anclas) además de la recta descrita por los parámetros que representan el factor  $r$  estimado.

FACTOR $r$	P1	P2	P3	P4	P5	P6	P7	P8	P9
$z$	2.41	2.55	12.31	1.91	12.71	7.54	2.59	4.37	9.34
$\sqrt{\hat{r}}$	2.47	2.49	2.49	2.52	2.52	2.52	2.50	2.46	2.43

**Tabla 26.** Factor  $r$  necesario y obtenido en las posiciones de E2.

Con los valores de  $F$  y la descripción del factor  $r$  con la distancia proporcionada por los parámetros  $k$  y  $r_0$  se obtienen las estimaciones del error del posicionamiento RSS mostradas en la **Tabla 27** donde además se compara con el error real obtenido en cada una de las posiciones. Se observa que los resultados son en la mayoría de casos próximos a los reales pero en los casos que los resultados difieren del real la estimación siempre es menor, motivo por el cual se obtiene, igual que en el caso anterior, un error promedio estimado menor que el promedio real, en este caso es un 29% más pequeño que es una diferencia menor que la del caso anterior. En este caso la diferencia media entre los valores estimados y los reales es de 1.69 metros, que es muy parecida a la obtenida en el escenario anterior.

ERROR RSS (m)									PROMEDIO
P1	P2	P3	P4	P5	P6	P7	P8	P9	(m)
12.93	5.98	4.65	1.17	2.93	2.57	5.21	8.16	3.95	5.28
ERROR RSS ESTIMADO (m)									
13.21	5.82	0.94	1.55	0.58	0.86	5.02	4.60	1.03	3.73
DIFERENCIA									
0.28	0.15	3.71	0.37	2.35	1.71	0.19	3.56	2.92	1.69

**Tabla 27.** Error RSS estimado en E2.

### 6.3.2 Resultados híbridos con estimaciones de error.

A continuación, se aplican las técnicas híbridas estudiadas en el PFC utilizando las estimaciones de error obtenidas en el apartado anterior como el error instantáneo cometido en cada una de las posiciones, de esta manera se puede comprobar el efecto que tiene la diferencia entre las estimaciones anteriores y el error real en la precisión de las estimaciones de posición híbridas tratadas. En el **Capítulo 5** se presentaron los resultados que proporcionarían las técnicas híbridas en el caso de disponer de los errores instantáneos reales en cada una de las posiciones, sin embargo estos resultados sirven para comprobar el correcto funcionamiento de las técnicas en el caso que se le proporcionan unos parámetros o medidas de precisión correctas y no sirven para validar una posible implementación de un sistema que utilice estas técnicas ya que estos errores instantáneos no se disponen en la práctica. Ahora, una vez comprobado que de media las estimaciones del error proporcionan valores cercanos a los reales, se pretende demostrar que con la utilización de estas estimaciones propuestas se podría obtener un sistema de localización híbrido funcionando con errores instantáneos y obtener resultados próximos a los obtenidos con los errores reales utilizados en el **Capítulo 5**.

Del estudio de las técnicas híbridas y de los resultados experimentales que proporcionan se sabe que su precisión no se basa en lo buena o mala que sea una de las estimaciones básicas si no que depende más bien de la relación que hay entre las precisiones de las estimaciones básicas. En el presente capítulo se ha propuesto una técnica para obtener errores instantáneos del posicionamiento RSS, pero no se ha propuesto nada para la obtención de errores GPS. Se ha mencionado la dependencia del error GPS con varios factores a partir de los que se podría extraer el error cometido por GPS, sin embargo en los resultados presentados a continuación se utiliza como error GPS el error promedio obtenido en cada uno de los escenarios de localización, dejando como línea futura la propuesta de alguna técnica que permita obtener el error del posicionamiento GPS con el receptor disponible en la WSN.

- **Escenario de localización 1 (E1).**

En el apartado anterior se han obtenido las estimaciones de los errores del posicionamiento basado en medidas RSS para cada una de las posiciones en las que se han tomado medidas. Utilizando las estimaciones obtenidas en el escenario de localización 1 (**Tabla 23**) en las técnicas de localización híbridas estudiadas a lo largo del PFC se obtienen los resultados de posicionamiento mostrados en la **Tabla 28**, donde se indican los errores cometidos en cada una de las posiciones por las distintas técnicas básicas y las técnicas híbridas instantáneas utilizando como errores instantáneos las estimaciones obtenidas en el apartado anterior para el escenario E1. Además de los errores cometidos en cada una de las posiciones se incluye el error promedio de cada una de las técnicas de localización analizadas.

ERROR INSTANTANEO						ERROR MEDIO
TÉCNICA	P1	P2	P3	P4	P5	PROMEDIO
RSS	0.99	0.27	0.56	4.07	9.68	3.11
GPS	47.19	3.76	1.83	2.53	4.04	11.87
Ponderación Inst.	0.99	0.27	0.56	3.94	4.66	<b>2.08</b>
Ponderación Mod. Inst.	0.43	0.29	0.57	3.94	8.02	2.65
BLUHE Inst.	1.03	0.36	0.57	3.83	7.48	2.65

**Tabla 28.** Error de posicionamiento con los errores estimados (E1).

Antes de pasar a analizar los resultados obtenidos comentar que en este caso para la técnica de ponderación simple se utiliza el error RSS estimado para obtener los parámetros de ponderación y se invierte el orden de los productos, es decir, la expresión para obtener la estimación híbrida sería, siendo  $\mu = 1/\hat{e}_{RSS}$ :

$$\hat{P}_H = \mu \cdot \hat{P}_{RSS} + (1 - \mu) \cdot \hat{P}_{GPS}$$

Además en los casos que se obtenga un valor de  $\mu$  mayor que 1 se utilizará  $\mu = 1$ , para mantener la definición teórica establecida en el **Capítulo 3** del parámetro. Por este mismo motivo en las tres primeras posiciones el error de la técnica de ponderación simple tiene los mismos errores que los obtenidos con la técnica RSS, ya que en esas posiciones se utiliza  $\mu = 1$  con lo que únicamente se utilizan las coordenadas RSS para obtener el posicionamiento híbrido. Este hecho es uno de los que hace que con la ponderación simple se obtengan los mejores resultados ya que en esas tres posiciones el posicionamiento RSS es extremadamente preciso y con BLUHE la utilización de las medidas GPS se comporta como ruido en las estimaciones RSS y por eso el empeoramiento del error RSS en esas posiciones por su parte. En cuanto a la ponderación modificada en esas tres primeras posiciones el error es parecido, hasta incluso mejor en algunos casos (P1), pero el problema aparece en la posición P5 donde la diferencia entre los errores de la ponderación simple y modificada es aproximadamente el doble. Esto es debido, igual que en el caso de la técnica BLUHE, al hecho de utilizar el error promedio GPS en todas las posiciones para la estimación de los parámetros de combinación correspondientes a cada una de las técnicas.

En el caso de P5 se utiliza como medida de error GPS un valor de unos 11 metros cuando realmente en esa posición el error GPS es de unos 4 metros, de esta manera ambas técnicas (Ponderación modificada y BLUHE) asumen que en esa posición la mejor estimación es la RSS cuando realmente no es así, mientras que la ponderación simple al utilizar únicamente el error RSS estimado en esa posición.

En la **Tabla 29** se recogen los errores promedios obtenidos en el **Capítulo 5** para las distintas técnicas de localización utilizadas en el mismo escenario que se está analizando (E1). Comparando esos resultados con los obtenidos utilizando los errores estimados en las técnicas híbridas instantáneas se observa que no se consigue las mismas prestaciones que con las técnicas instantáneas teóricas (utilizando errores instantáneos reales) pero que se consigue mejorar considerablemente los resultados obtenidos por las técnicas híbridas promediadas con las que no se conseguía mejorar el menor de los errores básicas, consiguiendo ahora reducirlo en un 33%.

TÉCNICA	ERROR PROMEDIO
RSS	3.11
GPS	11.87
Ponderación	<b>3.42</b>
PM	3.82
BLUHE	4.48
Pond. Inst.	2.73
PMI	<b>1.61</b>
BLUHE Inst.	1.89

**Tabla 29.** Errores para las técnicas promedio e instantáneas teóricas (E1).



- **Escenario de localización 2 (E2).**

Como en el caso anterior, utilizando las estimaciones del error de posicionamiento RSS obtenidas en el escenario de localización 2 (**Tabla 27**) en las técnicas de localización híbridas se obtienen los resultados de la **Tabla 30** que incluye los valores del error de posicionamiento cometido en cada una de las posiciones además del error promedio para cada una de las técnicas de localización utilizadas, siendo la técnica BLUHE la que mejores resultados proporciona utilizando los errores RSS estimados en este escenario básicamente por que en este caso no hay estimaciones RSS tan precisas como en el caso anterior que puedan provocar que las estimaciones GPS actúen como ruido en las estimaciones híbridas, además en este caso el promedio del error GPS que se utiliza como medida de precisión de las estimaciones GPS es lo suficientemente correcta como para evitar incoherencias entre la relación entre errores básicos real y la obtenida utilizando el error GPS medio, tal y como pasaba en la posición P5 del escenario anterior.

TÉCNICA	ERROR INSTANTANEO									ERROR PROMEDIO
	P1	P2	P3	P4	P5	P6	P7	P8	P9	
RSS	12.93	5.98	4.65	1.17	2.93	2.57	5.21	8.16	3.95	5.28
GPS	8.18	2.89	6.7	6.52	2.27	6.02	5.39	8.16	10.7	6.31
Pond. Inst.	7.55	3.08	4.65	3.06	2.93	2.57	4.18	4.61	3.84	4.05
**PMI	8.18	4.11	4.40	1.83	2.69	2.20	3.33	2.22	3.71	3.63
BLUHE Inst.	6.67	4.05	4.26	2.22	2.49	1.97	3.27	1.28	3.67	<b>3.32</b>
*PM: Ponderación Modificada; **PMI: PM Instantánea.										

**Tabla 30.** Error de posicionamiento con los errores estimados (E2).

Con los resultados obtenidos se puede comprobar que con la técnica BLUHE instantánea utilizando los errores RSS estimados se consigue una reducción del 37% en el menor de los errores básicos, que en este caso es el error RSS de 5.28 m. Esta reducción en el error no llega a ser tan grande como la reducción del 48% obtenida con las técnicas instantáneas teóricas (utilizando errores reales) pero si que consigue reducir un 5% más de lo que se conseguía reducir con las técnicas de ponderación promedio. Realmente la mejora obtenida respecto los resultados con las técnicas híbridas promedio no es muy grande o al menos tan grande como en el caso anterior, básicamente por que en el caso anterior las estimaciones promedio no funcionaban correctamente por lo que ya se comentó. Estos últimos porcentajes se pueden comprobar en la **Tabla 31**.

TÉCNICA	ERROR PROMEDIO
RSS	5.28
GPS	6.31
Ponderación	4.41
PM	3.59
BLUHE	<b>3.54</b>
Pond. Inst.	4.26
PMI	<b>2.73</b>
BLUHE Inst.	2.84

**Tabla 31.** Errores para las técnicas promedio e instantáneas teóricas (E2).

En el **Capítulo 5** ya se comentó que la medida del error medio GPS en el escenario E1 no era muy precisa mientras que la obtenida en E2 sí que lo era. El motivo de decidir si una medida del error es buena o mala residía en la diferencia con el valor del error medio con el valor del error cometido en cada una de las posiciones medidas, en el caso de E1 esta diferencia era grande en la mayoría de las posiciones mientras que en E2 esta diferencia era considerablemente pequeña en la mayoría de las posiciones. La diferencia en la calidad de la medida del error GPS promedio en el escenario E1 y en el escenario E2 es la que provoca que en el primer escenario la mejor técnica sea la ponderación simple, que únicamente utiliza las estimaciones del error RSS, y en el segundo la mejor técnica sea BLUHE tal y como teóricamente debería ser por la propia definición de BLUHE. El hecho que en E1 sólo se utilice como medida de precisión los errores RSS estimados y proporcione unos resultados satisfactorios permite concluir que la estimación del error RSS propuesta en este capítulo es una buena medida para la precisión del posicionamiento RSS, esta conclusión queda corroborada en el momento que en E2, donde se utiliza una buena medida del error GPS promedio, tanto la técnica BLUHE como la ponderación modificada funcionan correctamente.

### **6.3.3 Conclusiones de resultados experimentales.**

En este apartado se han tratado los resultados obtenidos por el posicionamiento híbrido implementado en el sistema de localización diseñado en el PFC utilizando la técnica de estimación de errores RSS propuesta en el presente capítulo. Con estos resultados experimentales obtenidos se ha podido comprobar tanto la técnica de estimación de errores en el posicionamiento RSS como el funcionamiento de las técnicas híbridas utilizando estas estimaciones. Entre los resultados analizados se encuentran los obtenidos en los dos primeros escenarios de localización estudiados en el **Capítulo 5** adjuntando los resultados del tercer escenario de localización (E3) en el **ANEXO 3**, donde además se incluyen los resultados obtenidos al utilizar en las estimaciones híbridas el error instantáneo GPS real en vez del error GPS promedio.

Una vez analizados los resultados obtenidos por las estimaciones del error RSS y el posicionamiento obtenido con las técnicas híbridas utilizando estas estimaciones junto el error promedio GPS se puede extraer ciertos comentarios relacionados con ambos aspectos:

- Para la estimación del error en el posicionamiento RSS son necesarios los dos parámetros definidos teóricamente en el apartado anterior:
  - $F$ : Indica una medida de la diferencia entre las distancias estimadas y las **distancias posición** (distancia entre posición estimada y motas anclas con las que se localiza). Experimentalmente se ha comprobado que el parámetro se corresponde con la definición teórica, de tal manera que si la diferencia entre distancias es pequeña éste será próximo a 0 y se corresponde con un error en el posicionamiento pequeño.

- $r$ : Teóricamente se define para representar el efecto del error en función de la distancia, por ello se realiza un proceso de “curve fitting” mediante las medidas de error RSS real y el factor  $F$  obtenido en cada una de las posiciones que determinan una distancia estimada media. Experimentalmente se comprueba que  $r$  tiene cierta dependencia con la distancia pero no es la dependencia lineal asumida teóricamente, sino que se observa la dependencia con otro tipo de factor que no se está teniendo en cuenta y provoca que la estimación del parámetro  $r$  según la distancia sea aproximadamente constante y por lo tanto, en general, la estimación de  $r$  no se corresponde con el factor teórico que se debería aplicar.
- A pesar de la falta de coherencia entre la definición teórica del factor  $r$  y su efecto real las estimaciones del error RSS suelen ser correctas en la mayoría de las posiciones obteniendo unas diferencias respecto los errores reales pequeñas, de media se obtiene una diferencia de 1.6 m en los dos escenarios analizados obteniendo errores máximos del orden de 3 metros en el 40% de posiciones mientras que en el 60% restante de los casos se obtienen errores entre 0.19 y 1 metro, obteniendo por lo tanto de media unas estimaciones más que satisfactorias. Se dice que estas estimaciones son satisfactorias en el sentido que al utilizarlas en las técnicas híbridas se obtiene un correcto funcionamiento ya que:
- Se consigue mejorar los resultados de las técnicas básicas (RSS, GPS) consiguiendo reducir el menor de los errores básicos aproximadamente un 35% obteniendo unos errores medios entre 2 y 3 metros, en los casos estudiados.
  - Se obtiene una mejora respecto los resultados obtenidos al utilizar las técnicas híbridas promedio (utilizando errores promedio de las técnicas básicas). En uno de ellos se consigue mejorar un 40% mientras que en el otro únicamente se mejora un 6% por lo ya comentado con anterioridad sobre las medidas GPS promedio. De esta manera, ya sea mucho o poco se consigue una mejora respecto las técnicas promedio en todos los escenarios de localización analizados obteniendo errores promedio menores a 3 metros.
- Todo y conseguir el objetivo de mejorar siempre las técnicas básicas, hecho que no ocurre con las técnicas híbridas promedio (consultar E1), no se consigue obtener la precisión teórica que se obtendría al utilizar los errores instantáneos reales de cada una de las técnicas básicas. Se puede pensar que no se llega a la precisión teórica por que se está utilizando el error promedio GPS en vez de unos errores instantáneos, sin embargo en el **ANEXO 3** se adjuntan los resultados obtenidos al utilizar los errores instantáneos GPS reales y aun así no se consigue la precisión teórica.

- Con estos resultados queda presente que el hecho de utilizar estimaciones del error RSS, que no son los errores reales, limita la precisión de las técnicas híbridas debida a las imprecisiones que puedan tener las estimaciones y no permite obtener la precisión teórica.

Con los resultados obtenidos en este capítulo ya se puede dar por concluidos todos los objetivos presentados en la introducción de la presente memoria ya que ahora si que se dispone de la capacidad de obtener unas medidas aceptables de la precisión de las técnicas básicas que permitan obtener unos parámetros de combinación híbrida correctos y útiles para obtener estimaciones híbridas válidas cuyo error sea menor que el más pequeño de los errores de las técnicas básicas. Concretamente se consigue obtener una medida correcta de la precisión del posicionamiento RSS en cada una de las posiciones estimadas dejando información para que en proyectos futuros se pueda obtener algo parecido con el posicionamiento GPS.

Para terminar comentar que a falta de una posible revisión del factor  $r$  en líneas futuras, la obtención de los parámetros que definirán la estimación de este parámetro ( $k$  y  $r_0$ ) se debería realizar en la fase de entrenamiento igual que para el caso de los parámetros de propagación necesarios para tomar las estimaciones de distancia a partir de las medidas RSS. De esta manera, sería conveniente utilizar una tabla de parámetros de factor  $r$  con el mismo objetivo con el que se utiliza la tabla de parámetros de propagación, es decir, disponer de un gran número de parámetros para poder escoger aquellos que proporcionen la mejor descripción del factor  $r$  experimental.



## 7 Conclusiones.

Tras la finalización del proyecto, se puede afirmar que se han superado los objetivos marcados al inicio de la memoria. El principal objetivo del presente trabajo de final de carrera ha sido el diseño, la implementación y validación de un sistema de localización híbrido basado en una red de sensores inalámbricos. El sistema de localización híbrido debe brindar la posibilidad de combinar técnicas de localización para permitir el posicionamiento tanto en entornos exteriores como interiores. Para el cumplimiento del objetivo principal se debe seguir una serie de pasos que permitan la obtención de los objetivos secundarios a partir del cual se consigue llegar al principal, los pasos seguidos a lo largo del proyecto y mencionados en la presente memoria son los siguientes:

- En primer lugar se debe estudiar las capacidades y limitaciones básicas de las WSN para poder escoger correctamente las técnicas de localización adecuadas. Estas técnicas de localización son las técnicas basadas en medidas RSS y el posicionamiento GPS ya que las WSN permiten realizar medidas RSS fácilmente y dan la posibilidad de utilizar receptores GPS en los nodos que la constituyen, respectivamente.
  - Estas dos técnicas de localización permiten obtener dos tipos de posicionamiento básico con unas precisiones dadas por las ventajas y limitaciones de cada una de las técnicas básicas, que básicamente son:
    - GPS permite un posicionamiento global aceptable en entornos exteriores.
    - El posicionamiento RSS permite un posicionamiento aceptable en entornos interiores además de la posibilidad de dar apoyo a GPS en los entornos exteriores donde éste no funcione correctamente.
- Una vez escogidas las dos técnicas básicas a utilizar en el sistema se puede combinar las ventajas de cada una para combatir las respectivas limitaciones y obtener así la capacidad de posicionamiento en entornos tanto exteriores como interiores incluyendo aquellos en los que GPS falla por alguno de los motivos comentados en su debido momento (Introducción **Capítulo 3**).
  - Para ello se podría escoger una de las técnicas básicas según el tipo de escenario en el que se encuentre el usuario a localizar, sin embargo el sistema propuesto va aún más allá y en el caso que se puedan utilizar ambas técnicas básicas las combina de forma inteligente para obtener una localización más precisa que la obtenida con las técnicas básicas.

- Estas combinaciones inteligentes de las técnicas básicas son las denominadas técnicas híbridas propuestas en el **Capítulo 3** que pueden clasificarse en dos tipos:
  - **Intuitivas:** Técnicas que valoran más una técnica básica que otra según la precisión de una de ellas, para el caso de la ponderación simple, o según la precisión relativa de una respecto la otra para el caso de la ponderación modificada.
  - **Formal:** Técnica obtenida a partir de un desarrollo matemático que permite resolver el problema de la localización híbrida (mejorar precisión básica). Concretamente la solución del problema se basa en la obtención del estimador insesgado y de menor varianza posible dado el modelo de señal presentado en (16) y el estimador lineal en (17).
- El principio de funcionamiento de las técnicas híbridas se basa en la obtención de unos parámetros que permitan la combinación lineal entre las dos estimaciones obtenidas con las técnicas básicas. Estos parámetros de combinación se obtienen a partir de medidas de la precisión o del error de cada una de las técnicas de posicionamiento básico, la obtención de los parámetros se puede realizar a partir de:
  - **Medidas instantáneas:** Errores cometidos en cada una de las posiciones estimadas con la localización básica lo que permite obtener un parámetro de combinación distinto en cada una de las posiciones estimadas. Este tipo de medidas se obtienen en una fase de prueba del sistema en la que se conocen las posiciones medidas ya que actualmente no existen métodos de obtención de este tipo de medidas.
  - **Medidas promediadas:** Errores promedios obtenidos con la media de los errores cometidos en varias posiciones por la localización básica, esto obliga utilizar el mismo parámetro de combinación en las distintas posiciones estimadas.
- Dada la limitada capacidad de cálculo de los nodos constituyentes de la WSN y la necesidad de hacer ciertas operaciones complejas como la transformación de las coordenadas GPS a unas coordenadas adecuadas para la combinación híbrida con las coordenadas del posicionamiento RSS, se decide implementar un sistema centralizado en el sentido que no es la mota a localizar quien calcula su posición sino que es otro elemento disponible en la WSN, concretamente el PC utilizado como estación base.

- Siendo el PC el elemento encargado de calcular la posición se tiene la posibilidad de utilizar *software* como MATLAB para el procesamiento de los datos obtenidos en la WSN. De esta manera se divide la implementación del sistema en dos fases de programación:
  - **Programación de las motas:** La programación dedicada a la adquisición de las medidas RSS y GPS tomadas por el nodo a localizar así como la programación de las motas anclas y del *Gateway* que recolecta los datos para pasarlos al PC.
  - **Programación en MATLAB:** Archivos que dan la funcionalidad de localización híbrida al sistema implementado. El archivo más importante es el archivo *Main.m* que se encarga de ejecutar el resto de funciones creadas para obtener los resultados que permitan validar las técnicas híbridas y el sistema de localización híbrido diseñado.
- Una vez diseñado e implementado el sistema híbrido se procede a la validación del sistema a partir de medidas experimentales que permitan obtener los resultados del posicionamiento híbrido diseñado. Para la validación del sistema se simula el funcionamiento del posicionamiento híbrido en varios escenarios mediante el archivo *Main*. Con los resultados proporcionados por el *Main* en los distintos escenarios de medida se pueden recoger las características básicas del sistema híbrido tras su implementación:
  - Existe una gran dependencia en el posicionamiento RSS diseñado con los parámetros de propagación utilizados. Se ha comprobado que en la mayoría de los casos los parámetros que describen mejor el ambiente de un escenario de localización son los parámetros extraídos en el propio escenario de localización.
    - Debido a esta dependencia y la importancia de las medidas realizadas para la obtención de los parámetros de propagación, se decide crear una tabla de parámetros de propagación que reúna una gran cantidad de parámetros de propagación extraídos en una gran variedad de escenarios y de esta manera poder elegir aquellos parámetros que mejor se adapten al propio escenario de localización.
  - Comprobadas las prestaciones de las dos técnicas básicas queda presente la gran capacidad de posicionamiento en entornos locales ya sean exteriores o interiores, obteniendo de media un error menor que el obtenido en GPS en todos los escenarios analizados.



- Para validar el sistema de localización híbrido se analizan los resultados obtenidos por las técnicas híbridas utilizando medidas promediadas, ya que son medidas de las que se disponen. Sin embargo, para validar el correcto funcionamiento de las técnicas híbridas propuestas además se analizan los resultados obtenidos al utilizar medidas instantáneas, que en este caso son los errores reales presentes en las estimaciones de las distintas posiciones.
  - Para que las técnicas híbridas funcionen correctamente con medidas promediadas es necesario que estas medidas sean buenas en el sentido de no diferir considerablemente con los errores cometidos en cada una de las posiciones. Si la medida no es correcta en este sentido las técnicas híbridas no tendrán un funcionamiento aceptable y el sistema híbrido en tal escenario no consigue reducir el menor de los errores básicos.
  - Utilizando medidas instantáneas se demuestra el correcto funcionamiento del desarrollo teórico realizado para obtener las distintas técnicas híbridas, obteniendo la principal capacidad deseada de mejorar las precisiones de las dos técnicas básicas en cualquier tipo de escenario de localización.
- Al ver que las técnicas híbridas necesitan medidas instantáneas para el correcto funcionamiento en cualquier escenario de localización, todo y que obteniendo buenas medidas promediadas se podrían aceptar los resultados, se decide buscar algún método que permita identificar los errores cometidos por cada una de las técnicas básicas en las distintas posiciones. En el PFC se ha buscado un método para la identificación de los errores en el posicionamiento RSS ya que este posicionamiento ha sido diseñado y creado desde cero con lo que se dispone de más información que en el caso del posicionamiento GPS en el que únicamente se dispone de la información que el receptor utilizado proporcione.
  - Para obtener medidas del error del posicionamiento RSS se estudia la información disponible sobre el error GPS para realizar una analogía con el error del posicionamiento RSS. Concretamente la analogía se basa en obtener dos parámetros relacionados con el impacto de las medidas de distancia erróneas y con los efectos del entorno que provocan estas medidas erróneas. Estos parámetros son:
    - ***F***: Indica el efecto de las medidas de distancia erróneas en el cálculo del posicionamiento.
    - ***r***: Representa el efecto del error en función de la distancia.

- Una vez definida la manera de identificar los errores o de estimarlos es posible comprobar su funcionamiento a partir de los resultados obtenidos en la validación previa del sistema híbrido. Con esos resultados se comprueba el funcionamiento de la técnica de errores RSS y el funcionamiento de las técnicas híbridas utilizando este tipo de medidas instantáneas junto las medidas promediadas de GPS:
  - Antes de analizar los resultados de las estimaciones de errores, se comprueba experimentalmente la definición teórica de los dos factores utilizados:
    - $F$ : Experimentalmente se corresponde con su definición teórica de tal manera que si en una posición se obtiene un error de posicionamiento pequeño este factor toma un valor próximo a 0.
    - $r$ : En este caso experimentalmente se comprueba que en cierta medida si que depende de la distancia pero no tiene la correspondencia lineal asumida teóricamente. Por lo tanto, se debería realizar una revisión teórica de este parámetro para que incluyera los efectos que actualmente no se están teniendo en cuenta y provocan la incoherencia entre la definición teórica y los resultados experimentales.
  - A pesar de la falta de correspondencia entre el factor  $r$  teórico y el obtenido experimentalmente, las estimaciones del error de posicionamiento RSS son en general unas buenas estimaciones obteniendo una diferencia media de poco más de 1 metro con los errores reales. Estas estimaciones suelen proporcionar valores menores que los reales dejando presente la falta de algún factor adicional que actualmente no se está teniendo en cuenta, aun así se puede concluir que las estimaciones del error RSS son unas medidas instantáneas satisfactorias.
    - Estas estimaciones se consideran unas medidas instantáneas satisfactorias por el hecho que al utilizarlas en las técnicas híbridas, junto el error promedio GPS, se obtiene un correcto funcionamiento consiguiendo mejorar los resultados de las técnicas básicas en todos los escenarios de localización analizados.
    - El hecho de utilizar estimaciones del error RSS en vez de los errores reales limita la precisión de las técnicas híbridas debido a las imprecisiones de las estimaciones y por lo tanto no permite obtener la precisión teórica obtenida al utilizar las medidas instantáneas reales de ambas técnicas básicas.

- Antes de poner en marcha el sistema son necesarios ciertos parámetros y criterios que se deben obtener con anterioridad. Entre éstos se encuentra la extracción de parámetros de propagación, representación de las estimaciones del factor  $r$  o la propia obtención de las medidas promediadas de las técnicas básicas. Para obtener estos parámetros es necesario la realización de medidas con la WSN en los escenarios de localización en los que se utilice el sistema de localización híbrido, para ello se define la fase de entrenamiento como la etapa previa a la implementación final o puesta en marcha del sistema en la que se obtienen todos estos parámetros y criterios necesarios para el correcto funcionamiento del sistema de localización híbrido diseñado, implementado y validado en el presente PFC.

Tras este repaso de los puntos tratados en la memoria quedan concluidos todos los objetivos presentados al inicio de la memoria y que daban sentido y forma de ser al proyecto, sin embargo el desarrollo expuesto puede ser la base de un desarrollo más amplio y complejo que representaría las posibles líneas de continuación de este proyecto. Estas líneas futuras pasan obviamente por utilizar el sistema de localización híbrido implementado o al menos la idea de combinación híbrida, dejando líneas de investigación abiertas y ciertas mejoras en aspectos como:

- Estudio de técnicas híbridas destinadas exclusivamente a escenarios interiores, para poder obtener posicionamiento en interiores equiparables a los que se puedan obtener en exteriores, por ejemplo con la combinación del posicionamiento RSS obtenido por una WSN con el obtenido por una red Wi-Fi.
- Estudio de modificaciones en el factor  $r$  para obtener estimaciones del error en el posicionamiento RSS más precisas.
- Obtener estimaciones de los errores obtenidos con el posicionamiento GPS proporcionado por la WSN a partir de los conceptos teóricos repasados.

Este posible trabajo puede llegar a representar una gran cantidad de tiempo que lo deja fuera del alcance de este proyecto de final de carrera. A pesar de esto, podría ser interesante afrontarlo en posteriores trabajos que partan como base del presente PFC y tengan como objetivo las mejoras o líneas de investigación abiertas en la presente memoria.





## 8 Bibliografía.

- [1] Kazem Sohraby; Daniel Minoli & Taiem Znati, Wireless Sensor Networks: Technology, Protocols, and Applications, Pag. 27. Editorial Willey Interscience. Canadá (2.007).
- [2] Roberto Fernández Martínez... [et al.], Redes inalámbricas de sensores: teoría y aplicación práctica, Pag. 16. Universidad de La Rioja, Servicio de Publicaciones, 2009.
- [3] Roberto Fernández Martínez... [et al.], Redes inalámbricas de sensores: teoría y aplicación práctica, Pag. 18. Universidad de La Rioja, Servicio de Publicaciones, 2009.
- [4] Página web oficial del fabricante de sensores inalámbricos Crossbow en la que se puede consultar los distintos productos ofrecidos para formar WSN: <http://bullseye.xbow.com:81/Products/wproductsoverview.aspx>.
- [5] Kazem Sohraby; Daniel Minoli & Taiem Znati, Wireless Sensor Networks: Technology, Protocols, and Applications, Pag. 18. Editorial Willey Interscience. Canadá (2.007).
- [6] datasheet de las motas utilizadas en el PFC:  
<http://bullseye.xbow.com:81/Products/productdetails.aspx?sid=264> .
- [7] *datasheet* de la placa de sensores utilizada en el PFC, disponible en:  
<http://www.memsic.com/products/wireless-sensor-networks/sensor-boards.html>.
- [8] Manual de usuario de la placa de sensores utilizada (MTS420/400CC), disponible en:  
<http://www.memsic.com/support/documentation/wireless-sensor-networks/category/6-user-manuals.html>.
- [9] *datasheet* del receptor GPS utilizado (ublox LEA-4A), disponible en:  
[http://jasresearch.com/downloads/LEA-4A\\_Data\\_Sheet.pdf](http://jasresearch.com/downloads/LEA-4A_Data_Sheet.pdf).
- [10] Kazem Sohraby; Daniel Minoli & Taiem Znati, Wireless Sensor Networks: Technology, Protocols, and Applications, Pag. 277-278. Editorial Willey Interscience. Canadá (2.007).
- [11] Página web con información del sistema operativo GalsC:  
<http://galsc.sourceforge.net/>.
- [12] Información sobre el sistema operativo TinyOS:  
[http://docs.tinyos.net/tinywiki/index.php/Main\\_Page](http://docs.tinyos.net/tinywiki/index.php/Main_Page).

- [13] Philip Levis & David Gay, TinyOS Programming. Editorial Cambridge Univesity Press 2009.
- [14] Roberto Fernández Martínez... [et al.], Redes inalámbricas de sensores: teoría y aplicación práctica, Capítulo 5 (APLICACIONES). Universidad de La Rioja, Servicio de Publicaciones, 2009.
- [15] Kazem Sohraby; Daniel Minoli & Taiem Znati, Wireless Sensor Networks: Technology, Protocols, and Applications, Capítulo 2 (*Applications of WSN*). Editorial Willey Interscience, 2.007.
- [16] Alan Mainwaring, Joseph Polastre, Robert Szewczyk... [et al.], Wireless Sensor Networks for Habitat Monitoring, 2002.
- [17] K. Martinez, P.Padhy, A. Riddoch... [et al.], Glacial Environment Monitoring using Sensor Networks, 2004.
- [18] Margaret Morris & Jay Lundell, Ubiquitous Computing for Cognitive Decline: Findings from Intel's Proactive Health Research, 2003.
- [19] The Economist, In Dust We Trust, Technology Quarterly, Vol.371, No.8379, pp.10-12.
- [20] L. Evers, M.J.J. Bijil, M. Marin-Perianu... [et al.], Wireless Sensor Networks and Beyond: A Case Study on Transport and Logistics, 2005.
- [21] Alan Bensky, Wireless Positioning Technologies and Applications, Pag. 162-163, Editorial Artech House Inc, 2008.
- [21'] Alan Bensky, Wireless Positioning Technologies and Applications, Pag. 8, Editorial Artech House Inc, 2008.
- [22] Alan Bensky, Wireless Positioning Technologies and Applications, Capítulo 7 (TOA & TDOA), Editorial Artech House Inc, 2008.
- [23] Alan Bensky, Wireless Positioning Technologies and Applications, Capítulo 8 (*Angle of Arrival*), Editorial Artech House Inc, 2008.
- [24] Alan Bensky, Wireless Positioning Technologies and Applications, Pag. 46, Editorial Artech House Inc, 2008.
- [25] Elliott Kaplan & Christopher Hegarty, Understanding GPS: principles and applications 2nd edition, Pag. 29-32, Editorial Artech House Inc, 2006.
- [26] Andrea Goldsmith, Wireless Communications, Pag. 25, Editorial Stanford Univesrity, 2005.

- [26'] Andrea Goldsmith, Wireless Communications, Capítulo 2 y 3 (*Path Loss and Shadowing & Statistical Multipath Channel Models*), Editorial Stanford University, 2005.
- [27] David Tse & Pramod Viswanath, Fundamentals of Wireless Communication, Capítulo 2 (*The Wireless Channel*), Editorial Cambridge University Press, 2005.
- [28] T. Okumura, E. Ohmori, and K. Fukuda, "Field strength and its variability in VHF and UHF land mobile service," *Review Electrical Communication Laboratory*, Vol. 16, No. 9-10, pp. 825–873, Sept.-Oct. 1968.
- [29] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Trans. Vehic. Technol.*, Vol VT-29, No. 3, pp. 317–325, Aug. 1980.
- [30] European Cooperative in the Field of Science and Technical Research EURO-COST 231, "Urban transmission loss models for mobile radio in the 900 and 1800 MHz bands," Revision 2, The Hague, Sept. 1991.
- [31] Steven M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory, Capítulo 8 (*Least Squares*) Pag. 225, Editorial Pearson, 1993.
- [32] Alan Bensky, Wireless Positioning Technologies and Applications, Pag. 163, Editorial Artech House Inc, 2008.
- [32'] Elliott Kaplan & Christopher Hegarty, Understanding GPS: principles and applications 2nd edition, Pag. 25, Editorial Artech House Inc, 2006.
- [33] Alan Bensky, Wireless Positioning Technologies and Applications, Pag. 165, Editorial Artech House Inc, 2008.
- [34] Steven M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory, Pag. 136, Editorial Pearson, 1993.
- [35] Steven M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory, *Appendix 6A*, Editorial Pearson, 1993.
- [36] Página Web donde se informa del estándar NMEA:  
<http://www.gpsinformation.org/dale/nmea.htm>.
- [37] Página Web con información sobre las características de las coordenadas UTM:  
[http://www.elgps.com/documentos/utm/coordenadas\\_utm.html](http://www.elgps.com/documentos/utm/coordenadas_utm.html).
- [38]: Función de MATLAB para realizar el cambio de coordenadas geográficas a coordenadas UTM:  
<http://www.mathworks.com/matlabcentral/fileexchange/10915-deg2utm/content/deg2utm.m>.



[39]: Crossbow, (CD Wireless Sensor Network Kit, MoteView/MoteWorks 2.0F) SLIDES\_MOTES\_XBOW: 6\_WirelessComms\_Design\_Considerations.pdf, Pag. 28, 2007.

[39']: Daniel Egea Roca, Proyecto de Final de Carrera: Posicionamiento con una red de sensores inalámbricos (WSN): implementación para una aplicación real, Pag. 71-73, UAB 2010. Disponible en:

[http://www.recercat.net/bitstream/handle/2072/97337/PFC\\_DanielEgeaRoca.pdf?sequence=1](http://www.recercat.net/bitstream/handle/2072/97337/PFC_DanielEgeaRoca.pdf?sequence=1).

[40]: Daniel Egea Roca, Proyecto de Final de Carrera: Posicionamiento con una red de sensores inalámbricos (WSN): implementación para una aplicación real, UAB 2010.

[41]: Benito González Caballero, Proyecto de Final de Carrera: Estudio e implementación de un sistema de *tracking* de personas mediante WSN, UAB 22 de Septiembre de 2011.

[42]: Elliott Kaplan & Christopher Hegarty, Understanding GPS: principles and applications 2nd edition, Pag. 322, Editorial Artech House Inc, 2006.

[43]: Elliott Kaplan & Christopher Hegarty, Understanding GPS: principles and applications 2nd edition, Pag. 323, Editorial Artech House Inc, 2006.

[44]: Elliott Kaplan & Christopher Hegarty, Understanding GPS: principles and applications 2nd edition, Capítulo 7 (*Performance of Stand-Alone GPS*), Editorial Artech House Inc, 2006.





## A ANEXO 1: Programación en NesC.

En el presente **ANEXO** se muestran los códigos utilizados para la programación de las motas que forman la red de sensores inalámbricos. Estos códigos están escritos en el lenguaje de programación NesC presentado en la memoria y las funcionalidades de los códigos se explican en el **Capítulo 4**. Básicamente se presentan los códigos utilizados en las motas anclas y el código utilizado en las motas desconocidas, para cada uno de los códigos se presentan los distintos archivos que forman la aplicación NesC, es decir, se muestra la configuración, el módulo y el formato del mensaje enviado por las motas.

### A.1 Programación de las motas anclas.

A continuación se muestra el código de las motas anclas, cuya posición es conocida y transmiten los mensajes para que las motas a localizar puedan medir la potencia recibida y poder obtener el posicionamiento RSS.

#### A.1.1 Configuración.

```
includes sensorboardApp;

/* This configuration shows how to use the Timer, LED, ADC and Messaging
components.
 * Sensor messages are broadcast single-hop over the RF radio
 */

configuration MyApp {
}
implementation {
  components Main, MyAppM, TimerC, LedsC, GenericComm as Comm,
HPLRF230M;

  Main.StdControl -> TimerC.StdControl;
  Main.StdControl -> MyAppM.StdControl;
  Main.StdControl -> Comm.Control;

  MyAppM.Timer -> TimerC.Timer[unique("Timer")];
  MyAppM.Leds -> LedsC.Leds;
  MyAppM.HPLRF230 -> HPLRF230M.HPLRF230;

  MyAppM.SendMsg -> Comm.SendMsg[AM_XSMSG];
  MyAppM.ReceiveMsg -> Comm.ReceiveMsg[AM_XSMSG];
}
```

### **A.1.2 Módulo.**

```
includes sensorboardAppp;  
includes TosTime;
```

```
/* This module shows how to use the Timer, LED, ADC and Messaging  
components.
```

```
* Sensor messages are broadcast single-hop over the RF radio  
*/
```

```
module MyAppppM {  
  provides {  
    interface StdControl;  
  }  
  uses {  
    interface Timer;  
    interface Leds;  
    interface SendMsg;  
    interface ReceiveMsg;  
  
    interface HPLRF230;  
  }  
}
```

```
implementation {  
  bool sending_packet = FALSE;  
  //tos_time_t temps;  
  TOS_Msg msg_buffer;  
  TOS_MsgPtr msgptr;  
  XDataMsg *pack;  
  XDataMsg *precv;  
  uint8_t node_rx;  
  int8_t rssi;  
  
  /**  
   * Initialize the component.  
   * @return Always returns <code>SUCCESS</code>  
   **/
```

```
command result_t StdControl.init() {  
  call Leds.init();
```

```
  atomic {  
    pack = (XDataMsg *)&(msg_buffer.data);  
  
    pack->xSensorHeader.board_id = SENSOR_BOARD_ID;  
    pack->xSensorHeader.packet_id = 3;  
    pack->xSensorHeader.node_id = TOS_LOCAL_ADDRESS;  
    pack->xSensorHeader.rsvd = 0;
```

```

    node_rx = 0;
    rssi = 42;
    atomic pack->xData.datap1.rssi = 89;

}

return SUCCESS;
}

/**
 * Start things up. This just sets the rate for the clock component.
 * @return Always returns <code>SUCCESS</code>
 */
command result_t StdControl.start() {
    // Start a repeating timer that fires every 1000ms
    return call Timer.start(TIMER_REPEAT, 3000);
}

/**
 * Halt execution of the application.
 * This just disables the clock component.
 * @return Always returns <code>SUCCESS</code>
 */
command result_t StdControl.stop() {
    return call Timer.stop();
}

void task SendData()
{
    if (sending_packet) return;
    atomic sending_packet = TRUE;

    // broadcast message over radio
    if (call SendMsg.send(TOS_BCAST_ADDR, sizeof(XDataMsg), &msg_buffer) !=
SUCCESS)
        sending_packet = FALSE;
    return;
}

/**
 * Toggle the red LED in response to the <code>Timer.fired</code> event.
 * Start the Light sensor control and sample the data
 * @return Always returns <code>SUCCESS</code>
 */
event result_t Timer.fired()
{
    //call Leds.redToggle();
    post SendData();
    call Leds.yellowToggle();
}

```

```

    return SUCCESS;
}

/**
 * Stop the Light sensor control, build the message packet and send
 */

/**
 * Sensor data message has been sucessfully sent over the radio
 * Toggle green LED to signal message sent
 * @return Always returns <code>SUCCESS</code>
 */
event result_t SendMsg.sendDone(TOS_MsgPtr msg, result_t success) {
    //call Leds.greenToggle();
    atomic sending_packet = FALSE;

    return SUCCESS;
}

event TOS_MsgPtr ReceiveMsg.receive(TOS_MsgPtr m) {
    //precv = (XDataMsg *)m->data;
    //atomic node_rx = precv->xSensorHeader.node_id;
    //atomic pack->xData.datap1.Txnodeid = node_rx;
    //atomic rssi = call HPLRF230.readReg(RG_PHY_ED_LEVEL);
}
}

```

### **A.1.3 Estructura del mensaje.**

```

typedef struct XSensorHeader{
    uint8_t board_id;
    uint8_t packet_id; // 3
    uint8_t node_id;
    uint8_t rsvd;
}__attribute__((packed)) XSensorHeader;
typedef struct PData1 {
    int8_t rssi;
} __attribute__((packed)) PData1;

typedef struct XDataMsg {
    XSensorHeader xSensorHeader;
    union {
        PData1 datap1;
    }xData;
} __attribute__((packed)) XDataMsg;

enum {
    AM_XSXMSG = 0,
};

```

## A.2 Programación de motas a localizar.

En este apartado se muestra el código de las motas cuya posición es desconocida y se pretende localizar a partir de la combinación del posicionamiento RSS y la localización GPS. La funcionalidad del código permite medir las RSS de los mensajes transmitidos por las motas anclas y permite obtener la localización GPS dada por el receptor GPS en la placa de sensores utilizada (*sensor-board*).

### A.2.1 Componente.

```
/* Copyright (c) 2004-2007 Crossbow Technology, Inc.
 * All rights reserved.
 * See license.txt file included with the distribution.
 */
/* $Id: TestMTS420cc.nc,v 1.1.2.2 2007/04/26 20:35:23 njain Exp $
 */

/* XSensor single-hop application for MTS420 sensorboard.
 *
 * @author Alan Broad, David M. Doolin, Hu Siquan, Mao Shifeng
 */

#include "appFeatures.h"
#include sensorboardApp;

configuration TestMTS420cc
{
    // this module does not provide any interface
}
implementation
{
    components Main, TestMTS420ccM, SensirionHumidity,
        IntersemaPressure,MicaWbSwitch,GenericComm as Comm,
        TimerC, Voltage, LedsC, Accel, TaosPhoto,HPLRF230M,
#ifdef MTS420
        UARTGpsPacket,
#endif
#ifdef FEATURE_EEPROM_TEST
        MTS420EEPROMC,
#endif
        ADCC;

    Main.StdControl -> TestMTS420ccM;
    Main.StdControl -> TimerC;

    TestMTS420ccM.HPLRF230 -> HPLRF230M.HPLRF230;

    TestMTS420ccM.CommControl -> Comm;
    TestMTS420ccM.Receive -> Comm.ReceiveMsg[AM_XSMSG];
```



```

    TestMTS420ccM.Send -> Comm.SendMsg[AM_XSXMSG];

// Wiring for gps
#ifdef MTS420
    TestMTS420ccM.GpsControl -> UARTGpsPacket;
    //TestMTS420ccM.GpsSend -> UARTGpsPacket;
    TestMTS420ccM.GpsReceive -> UARTGpsPacket;
    TestMTS420ccM.GpsCmd          ->          UARTGpsPacket.GpsCmd;
//UARTGpsPacket.GpsCmd;
#endif

    // Wiring for Battery Ref
    TestMTS420ccM.BattControl -> Voltage;
    TestMTS420ccM.ADCBATT -> Voltage;

// Wiring for Taos light sensor
    TestMTS420ccM.TaosControl -> TaosPhoto;
    TestMTS420ccM.TaosCh0 -> TaosPhoto.ADC[0];
    TestMTS420ccM.TaosCh1 -> TaosPhoto.ADC[1];

// Wiring for Accelerometer
    TestMTS420ccM.AccelControl->Accel.StdControl;
    TestMTS420ccM.AccelCmd -> Accel.AccelCmd;
    TestMTS420ccM.AccelX -> Accel.AccelX;
    TestMTS420ccM.AccelY -> Accel.AccelY;

// Wiring for Sensirion humidity/temperature sensor
    TestMTS420ccM.TempHumControl -> SensirionHumidity;
    TestMTS420ccM.Humidity -> SensirionHumidity.Humidity;
    TestMTS420ccM.Temperature -> SensirionHumidity.Temperature;
    TestMTS420ccM.HumidityError -> SensirionHumidity.HumidityError;
    TestMTS420ccM.TemperatureError -> SensirionHumidity.TemperatureError;

// Wiring for Intersema barometric pressure/temperature sensor
    TestMTS420ccM.IntersemaCal -> IntersemaPressure;
    TestMTS420ccM.PressureControl -> IntersemaPressure;
    TestMTS420ccM.IntersemaPressure -> IntersemaPressure.Pressure;
    TestMTS420ccM.IntersemaTemp -> IntersemaPressure.Temperature;

    TestMTS420ccM.Leds -> LedsC;
    TestMTS420ccM.Timer -> TimerC.Timer[unique("Timer")];

// Wiring for EEPROM, chrl 20060724
#ifdef FEATURE_EEPROM_TEST
    TestMTS420ccM.MTS420EEPROMControl -> MTS420EEPROMC.StdControl;
    TestMTS420ccM.MTS420EEPROM          ->          MTS420EEPROMC.MTS420EEPROM[0x55];
#endif
}

```

## A.2.2 Módulo.

```
/*
 * Copyright (c) 2004-2007 Crossbow Technology, Inc.
 * All rights reserved.
 * See license.txt file included with the distribution.
 *
 * $Id: TestMTS420ccM.nc,v 1.1.2.3 2007/04/26 20:35:31 njain Exp $
 */

/**
 * XSensor single-hop application for MTS420 sensorboard.
 *
 * Measures MTS400/420 weatherboard sensors & gps and converts to engineering
units
 * were possible.
 *-----
 * Output results through mica2 uart port and radio.
 * Use Xlisten.exe program to view data from either port:
 * uart: mount mica2 on mib510 with MTS400/420
 *      connect serial cable to PC
 *      run xlisten.exe at 57600 baud
 * radio: run mica2 with MTS400/420,
 *        run mica2 with TOSBASE
 *        run xlisten.exe at 57600 baud
 *-----
 * NOTES:
 * -Intersema pressure sensor control lines are shared with gps control lines
 * -Cannot enable gps rx/tx and intersema at same time
 *
 * - gps is always enabled, work for both MTS420 and MTS400 sensor boards.
 * - if gps not present (MTS400) then additional ~2sec gps timeout will occur
 *
 * Strategy:
 * 1. Turn on gps power and leave on
 * 2. sequentially read all weather sensors (green led on).
 *    - xmit weather sensor data
 * 3. get gps packet (red led on):
 *    - enable gps Rx,Tx lines to cpu
 *    - wait up to 1 sec to receive a packet (toggle yellow if no pkt)
 *    - xmit gps packet
 *    - disable gps Rx,Rx lines
 * 4. repeat 2,3
 * NOTE:
 * No real power strategy; just turns sensors on sequentially, gps always on.
 * Need I2C BusArbitration routines for better power control
 *
 * @author Alan Broad, David M. Doolin, Hu Siquan, Mao Shifeng
 */
```

```

/*****
* Data packet structure :
* * PACKET #1 (of 2)
* msg->data[0] : sensor id, MTS400 = 0x85,MTS420 = 0x86
* msg->data[1] : packet id = 1
* msg->data[2] : node id
* msg->data[3] : reserved
* msg->data[4,5] : battery ADC data
* msg->data[6,7] : humidity data
* msg->data[8,9] : temperature data
* msg->data[10,11] : cal_word1
* msg->data[12,13] : cal_word2
* msg->data[14,15] : cal_word3
* msg->data[16,17] : cal_word4
* msg->data[18,19] : intersematemp
* msg->data[20,21] : pressure
* msg->data[22,23] : taosch0
* msg->data[24,25] : taosch1
* msg->data[26,27] : accel_x
* msg->data[28, 3] : accel_y  TOS packet is 29 bytes 0..28
* PACKET #2 (of 2)
* msg->data[0] : sensor id, MTS400 = 0x85,MTS420 = 0x86
* msg->data[1] : packet id = 2
* msg->data[2] : node id
* msg->data[3] : reserved
* msg->data[4] : Hours
* msg->data[5] : Minutes
* msg->data[6] : Latitude degrees
* msg->data[7] : Longitude degrees
* msg->data[8,9,10,11] : Decimal seconds
* msg->data[12,13,14,15] : Latitude decimal minutes
* msg->data[16,17,18,19] : Longitude decimal minutes
* msg->data[20] : NSEWind
* msg->data[21] : whether the packet is valid
*****/

includes gps;
#include "appFeatures.h"

module TestMTS420ccM {
    provides interface StdControl;
    uses {

        //communication
        interface StdControl as CommControl;
        interface SendMsg as Send;
        interface ReceiveMsg as Receive;
        interface HPLRF230;

    }

#ifdef MTS420

```

```

//gps
    interface GpsCmd;
    interface StdControl as GpsControl;
    interface ReceiveMsg as GpsReceive;
#endif
// Battery
    interface ADC as ADCBATT;
    interface StdControl as BattControl;

//Accels
    interface StdControl as AccelControl;
    interface I2CSwitchCmds as AccelCmd;
    interface ADC as AccelX;
    interface ADC as AccelY;

//Intersema
    interface SplitControl as PressureControl;
    //interface StdControl as PressureControl;
    interface ADC as IntersemaTemp;
    interface ADC as IntersemaPressure;
    interface Calibration as IntersemaCal;

//Sensirion
    interface SplitControl as TempHumControl;
    interface ADC as Humidity;
    interface ADC as Temperature;
    interface ADCError as HumidityError;
    interface ADCError as TemperatureError;

//Taos
    interface SplitControl as TaosControl;
    interface ADC as TaosCh0;
    interface ADC as TaosCh1;

    interface Timer;
    interface Leds;

#if FEATURE_EEPROM_TEST
// EEPROM
    interface MTS420EEPROM;
    interface StdControl as MTS420EEPROMControl;
#endif
}
}

implementation
{
    enum {START,
        BUSY,
        EEPROM_WRITE,

```

```

EEPROM_READ,
GPS_BUSY,
BATT_DONE,
HUMIDITY_DONE,
PRESSURE_DONE,
LIGHT_DONE,
ACCEL_DONE,
GPS_DONE,
};
enum {SENSOR_NONE = 0,
      SENSOR_BATT_START = 10,

      SENSOR_HUMIDITY_START = 20,
      SENSOR_HUMIDITY_GETHUMDATA = 21,
      SENSOR_HUMIDITY_GETTEMPDATA = 22,
      SENSOR_HUMIDITY_STOP = 23,

      SENSOR_PRESSURE_START = 30,
      SENSOR_PRESSURE_GETCAL = 31,
      SENSOR_PRESSURE_GETPRESSDATA = 32,
      SENSOR_PRESSURE_GETTEMPDATA = 33,
      SENSOR_PRESSURE_STOP = 34,

      SENSOR_LIGHT_START = 40,
      SENSOR_LIGHT_GETCH0DATA = 41,
      SENSOR_LIGHT_GETCH1DATA = 42,
      SENSOR_LIGHT_STOP = 43,

      SENSOR_ACCEL_START = 50,
      SENSOR_ACCEL_GETXDATA = 51,
      SENSOR_ACCEL_GETYDATA = 52,
      SENSOR_ACCEL_STOP = 53
};

#define TIMER_PERIOD 2000           // timer period in msec
char count;

uint16_t calibration[4];           // intersema calibration words
norace uint8_t state;              //
uint8_t sensor_state;              // debug only

#ifdef MTS420
uint8_t gps_wait_cnt;              // cnts wait periods for gps pkt to arrive
uint16_t gps_sleep_cnt;
bool gps_pwr_on;                  // true if gps power on
uint8_t gps_work_factor;
char gga_fields[GGA_FIELDS][GPS_CHAR_PER_FIELD]; // = {{0}};
#endif
TOS_Msg msg_buf;
TOS_MsgPtr msg_ptr;

```

```

    bool sending_packet, WaitingForSend, IsUART;
    norace XDataMsg *pack;
    norace uint8_t iNextPacketID;
    XDataMsg *precv;
    uint8_t node_rx;
    uint8_t rssi;

    #if FEATURE_EEPROM_TEST
        // test EEPROM, chrl 20060724
        uint8_t WData[10];
        uint8_t test;
        uint8_t EEPROMwf;
        uint8_t EEPROMchkover;
        // uint16_t counter;
    #endif
    //RSSI DATA
    void task SendData()
    {
        if (sending_packet) return;
        atomic sending_packet = TRUE;

        // broadcast message over radio
        if (call Send.send(TOS_BCAST_ADDR,sizeof(XDataMsg),msg_ptr) !=
SUCCESS)
            sending_packet = FALSE;
        return;
    }
    /*****
    * Task to send uart and rf message
    *****/
    task void send_msg()
    {
        if (sending_packet) return;
        atomic sending_packet = TRUE;
        pack->xSensorHeader.board_id = SENSOR_BOARD_ID;
        pack->xSensorHeader.packet_id = iNextPacketID;
        pack->xSensorHeader.node_id = TOS_LOCAL_ADDRESS;
        // pack->xSensorHeader.rsvd = 0;

        node_rx = 0;
        rssi = 0;

        call Leds.yellowOn();
        if (IsUART)
        {
            if(call Send.send(TOS_UART_ADDR,sizeof(XDataMsg)-
1,msg_ptr)!=SUCCESS)
            {
                atomic sending_packet = FALSE;
                call Leds.greenToggle();
            }
        }
    }

```

```

    }
}
else
{
    if(call Send.send(TOS_BCAST_ADDR,sizeof(XDataMsg)-
1,msg_ptr)!=SUCCESS)
    {
        atomic sending_packet = FALSE;
        call Leds.greenToggle();
    }
}
return;
}

```

```

task void stopPressureControl()
{
    atomic sensor_state = SENSOR_PRESSURE_STOP;
    call PressureControl.stop();
    return;
}

```

```

task void stopTempHumControl()
{
    atomic sensor_state = SENSOR_HUMIDITY_STOP;
    call TempHumControl.stop();
    return;
}

```

```

task void stopTaosControl()
{
    atomic sensor_state = SENSOR_LIGHT_STOP;
    call TaosControl.stop();
    return;
}

```

```

task void powerOffAccel()
{
    atomic sensor_state = SENSOR_ACCEL_STOP;
    call AccelCmd.PowerSwitch(0); //power off
    return;
}

```

```

command result_t StdControl.init()
{
    atomic
    {
        msg_ptr = &msg_buf;
        sending_packet = FALSE;
        WaitingForSend = FALSE;

```

```

#ifdef MTS420

```

```

        gps_pwr_on = FALSE;
        gps_work_factor = 0;
    #endif
    }
    pack = (XDataMsg *)msg_ptr->data;
    // usart1 is also connected to external serial flash
    // set usart1 lines to correct state
    TOSH_MAKE_FLASH_OUT_OUTPUT();           //tx output
    TOSH_MAKE_FLASH_CLK_OUTPUT();           //usart clk

    call BattControl.init();
    call CommControl.init();
    call Leds.init();
    call Leds.greenOn();
    #ifdef MTS420
        call GpsControl.init();
    #endif

    call TaosControl.init();
    call AccelControl.init();    //initialize accelerometer
    call TempHumControl.init();  //init Sensirion
    call PressureControl.init(); //init Intersema

    #if FEATURE_EEPROM_TEST
        call MTS420EEPROMControl.init();

        // test EEPROM, chrl 20060724
        for(test=0; test<10;test++)
        {
            WData[test]=test;
        }
        test = 1;
        EEPROMwf=0;
        EEPROMchkover = 0;
    //
    #endif
        counter =0;
        return SUCCESS;
    }

    command result_t StdControl.start()
    {
        call HumidityError.enable();           //in case Sensirion doesn't
respond
        call TemperatureError.enable();        // same as above
        call CommControl.start();
        call BattControl.start();
    #ifdef MTS420
        call GpsControl.start();
        atomic gps_wait_cnt = 0;
        atomic gps_sleep_cnt = 0;
    #endif

```



```

        atomic state = START;
        atomic sensor_state= SENSOR_NONE;

        IsUART = TRUE;
        call Timer.start(TIMER_REPEAT, TIMER_PERIOD); //start up sensor
measurements
        return SUCCESS;
    }

    command result_t StdControl.stop()
    {
        call BattControl.stop();
#ifdef MTS420
        call GpsControl.stop();
        call GpsCmd.TxRxSwitch(0);
#endif
        call Timer.stop();
        call CommControl.stop();
        return SUCCESS;
    }

/*****
* Timer fired, test GPS, humidity/temp
* async for test only
* If gps_wait_cnt > 0 then gps is active, waiting for a packet
*****/
    event result_t Timer.fired()
    {
        uint8_t l_state;
        //call Leds.redToggle();

        atomic l_state = state;
        if (sending_packet || (l_state == BUSY))
            return SUCCESS; //don't overrun buffers

#ifdef MTS420
        if (!gps_pwr_on)
        {
            if (gps_sleep_cnt > GPS_SLEEP_INTERVAL * gps_work_factor)
            {
                if (GPS_WORK_INTERVAL>0)
                {
                    //turn on GPS power
                    return call GpsCmd.PowerSwitch(1);
                }
                else
                {
                    atomic gps_sleep_cnt = 0;
                }
            }
            return SUCCESS;
        }

```

```

    }
    else
    {
        gps_sleep_cnt++;
    }
}
else
{
    if (gps_sleep_cnt > GPS_WORK_INTERVAL * gps_work_factor)
    {
        if (GPS_SLEEP_INTERVAL > 0)
        {
            //turn off GPS power
            return call GpsCmd.PowerSwitch(0);
        }
        else
        {
            atomic gps_sleep_cnt = 0;
        }
        return SUCCESS;
    }
    else
    {
        gps_sleep_cnt++;
    }
}
#endif

#if FEATURE_EEPROM_TEST
    // chrl 20060810
    if (EEPROMchkover == 0)
    {
        call MTS420EEPROMControl.start();
        atomic EEPROMchkover = 1;
        return SUCCESS;
    }
    TOSH_uwait(10);
    if (test != 0)
    {
        test = 0;
        if (EEPROMwof == 0)
        {
            atomic state = BUSY;
            call
MTS420EEPROM.writePacket(0, 10, (char*)(WData), 0x01);
            return SUCCESS;
        }
    }
    else
    {
        test = 1;
    }
}

```

```

        if(EEPROMwf==1)
        {
            atomic state = BUSY;
            call MTS420EEPROM.readPacket(0,10,0x03);
            return SUCCESS;
        }
    }
#endif

    if (WaitingForSend)
    {
/*
* The GPS modules DC-DC booster can interfere with radio communication.
* If the GPS module must be continually powered and monitored during radio
* communication, then 3.3-3.6 volt lithium batteries are recommended to
* power the mote. Normal alkaline batteries are not recommended unless
* the GPS module is powered down during radio communication.
*
* If GPS module is to be powered down during radio communication,
*   uncomment following GpsCmd.PowerSwitch(0) line
* If the GPS module must be continually powered and monitored during radio
*   communication, comment following GpsCmd.PowerSwitch(0) line
*/

        /*
        if(gps_pwr_on)
        {
            // if(gps_sleep_cnt > GPS_WORK_INTERVAL*gps_work_factor)
            //   call GpsCmd.PowerSwitch(0);
            // else
            //   gps_sleep_cnt++;
        }*/

        post send_msg();
        return SUCCESS;
    }

    switch(l_state)
    {
        case START:
#ifdef FEATURE_GPS_ONLY
            call Leds.greenOff();
#endif
#ifdef MTS420
            atomic state = GPS_BUSY;
            return call GpsCmd.TxRxSwitch(1); //enable gps tx/rx
#else
            atomic state = START;
            return SUCCESS;
#endif
    }
#endif
#endif
//#else
/*atomic{

```

```

        state = BUSY;
        sensor_state = SENSOR_BATT_START;
    }
    call Leds.greenToggle();
    return call ADCBATT.getData(); //get vref data;
#endif

    break;
case BATT_DONE:
    atomic
    {
        state = BUSY;
        sensor_state = SENSOR_HUMIDITY_START;
    }
    return call TempHumControl.start();
case HUMIDITY_DONE:
    atomic
    {
        state = BUSY;
        sensor_state = SENSOR_PRESSURE_START;
    }
    return call PressureControl.start();
case PRESSURE_DONE:
    atomic
    {
        state = BUSY;
        sensor_state = SENSOR_LIGHT_START;
    }
    return call TaosControl.start();
case LIGHT_DONE:
    atomic
    {
        state = BUSY;
        sensor_state = SENSOR_ACCEL_START;
    }
    return call AccelCmd.PowerSwitch(1); //power on
case ACCEL_DONE:*/
    call Leds.greenOff();
#ifdef MTS420
    if (gps_pwr_on)
    {
        atomic state = GPS_BUSY;
        return call GpsCmd.TxRxSwitch(1); //enable gps tx/rx
    }
    else
    {
        atomic state = START;
        return SUCCESS;
    }
#else
    atomic state = START;

```

```

        return SUCCESS;
    #endif
        break;

#ifdef MTS420
    case GPS_BUSY:
        if (gps_wait_cnt >= GPS_MAX_WAIT)
        { // gps rcvd pkt before time out?
            call Leds.greenOn();
            call GpsCmd.TxRxSwitch(0); // no,disable gps tx/rx
switches
            atomic state = START;
            return SUCCESS;
        }
        else
        {
            call Leds.yellowToggle();
            gps_wait_cnt++; //keep waiting for gps pkt
            return SUCCESS;
        }
        break;
    case GPS_DONE:
        atomic state = START;
        return call GpsCmd.TxRxSwitch(0);
        break;
#endif
    }
    return SUCCESS;
}

```

*#ifdef MTS420*

*/\*\*\*\*\**

*\* Packet received from GPS - ASCII msg*

*\* 1st byte in pkt is number of ascii bytes*

*\* async used only for testing*

*GGA - Global Positioning System Fix Data*

*GGA,123519,4807.038,N,01131.324,E,1,08,0.9,545.4,M,46.9,M, , \*42*

*123519 Fix taken at 12:35:19 UTC*

*4807.038,N Latitude 48 deg 07.038' N*

*01131.324,E Longitude 11 deg 31.324' E*

*1 Fix quality: 0 = invalid*

*1 = GPS fix*

*2 = DGPS fix*

*08 Number of satellites being tracked*

*0.9 Horizontal dilution of position*

*545.4,M Altitude, Metres, above mean sea level*

*46.9,M Height of geoid (mean sea level) above WGS84*

*ellipsoid*

*(empty field) time in seconds since last DGPS update*

*(empty field) DGPS station ID number*

```

*****/
event TOS_MsgPtr GpsReceive.receive(TOS_MsgPtr data)
{
    uint8_t i,j,m,length,NS,EW;
    uint16_t k;
    uint32_t temp;
    bool end_of_field;
    char *packet_format;
    char *pdata;
    float dec_secs,dec_min;

    //change to GPS packet!!
    GPS_MsgPtr gps_data = (GPS_MsgPtr)data;

    // if gps have been scanned then stop receiving gps uart packet
    if (state == GPS_DONE)
        return data;

    // check for NMEA format, gga_fields[0]
    packet_format = gps_data->data;
    //
    if ( !((packet_format[3] == 'G') &&
        (packet_format[4] == 'G') &&
        (packet_format[5] == 'A')))
    {
        //SODbg(DBG_USR2, "No NEMA format, gps packet parse
failed!! \n");
        return data;
    }

    call GpsCmd.TxRxSwitch(0);

    // parse comma delimited fields to gga_filed[][]
    end_of_field = FALSE;
    i=0;
    k=0;
    length = gps_data->length;
    memset(gga_fields, '0', GGA_FIELDS*GPS_CHAR_PER_FIELD);
    memset(pack,0,sizeof(XDataMsg));
    while (i < GGA_FIELDS)
    {
        // assemble gga_fields array
        end_of_field = FALSE;
        j = 0;
        while ((!end_of_field) & (k < length))
        {
            if (gps_data->data[k] == GPS_DELIMITER)
            {
                end_of_field = TRUE;
            }

```

```

        else
        {
            gga_fields[i][j] = gps_data->data[k];
        }
        j++;
        k++;
    }
    /*
    // two commas (,,) indicate empty field
    // if field is empty, set it equal to 0
    if (j <= 1)
    {
        for (m=0; m<10; m++)
            gga_fields[i][m] = '0';
    }
    */
    i++;
}

// gga_msg.hours = call extract_hours(gga_fields[1]);
pdata=gga_fields[1];
pack->xData.dataGps.hour=10*(pdata[0]-'0') + (pdata[1]-'0');

// gga_msg.minutes = call extract_minutes(gga_fields[1]);
pack->xData.dataGps.minute=10*(pdata[2]-'0') + (pdata[3]-'0');

// uint32_t
// gga_msg.dec_sec = call extract_dec_sec(gga_fields[1]);
dec_secs = 10*(pdata[4]-'0') + (pdata[5]-'0') + 0.1*(pdata[7]-'0')
            + 0.01*(pdata[8]-'0') + 0.001*(pdata[9]-'0');
temp = (uint32_t)(dec_secs * 1000);
pack->xData.dataGps.dec_sec = temp;

// gga_msg.Lat_deg = call extract_Lat_deg(gga_fields[2]);
pdata=gga_fields[2];
pack->xData.dataGps.lat_deg=      (uint16_t)(10*(pdata[0]-'0')      +
(pdata[1]-'0'));

// gga_msg.Lat_dec_min = call extract_Lat_dec_min(gga_fields[2]);
dec_min = 10*(pdata[2]-'0') + (pdata[3]-'0') + 0.1*(pdata[5]-'0')
            + 0.01*(pdata[6]-'0') + 0.001*(pdata[7]-'0') +
0.0001*(pdata[8]-'0');
temp = (uint32_t)(dec_min * 10000);
pack->xData.dataGps.lat_dec_min = temp;

// gga_msg.Long_deg = call extract_Long_deg(gga_fields[4]);
pdata = gga_fields[4];
pack->xData.dataGps.long_deg = (100*(pdata[0]-'0') + 10*(pdata[1]-
'0') + (pdata[2]-'0'));

```

```

// gga_msg.Long_dec_min = call extract_Long_dec_min(gga_fields[4]);
dec_min = 10*(pdata[3]-'0') + (pdata[4]-'0') + 0.1*(pdata[6]-'0')
          + 0.01*(pdata[7]-'0') + 0.001*(pdata[8]-'0') +
0.0001*(pdata[9]-'0');
temp = (uint32_t)(dec_min * 10000);
pack->xData.dataGps.long_dec_min= temp;

NS = (gga_fields[3][0] == 'N') ? 1 : 0;
EW = (gga_fields[5][0] == 'W') ? 1 : 0;
pack->xData.dataGps.nsewind=  EW | (NS<<4); // eg. Status=
000N000E = 00010000

// uint8_t
pack->xData.dataGps.fixed=(uint8_t)(gga_fields[6][0]-'0'); // invalid

pdata = gga_fields[7];
// pack->xData.dataGps.SVs = 10*(pdata[0]-'0') + (pdata[1]-'0');

/*
if (gps_pwr_on)
{
    if (gps_sleep_cnt > GPS_WORK_INTERVAL*gps_work_factor)
        call GpsCmd.PowerSwitch(0);
    else
        gps_sleep_cnt++;
}
*/
atomic state = GPS_DONE;
if( (pack->xData.dataGps.fixed==0) ||
(pack->xData.dataGps.fixed==1) || (pack->xData.dataGps.fixed==2))
{
    iNextPacketID = 2; // issue gga packet xmit
    WaitingForSend = TRUE;
}
else
{
    memset(pack,0,sizeof(XDataMsg));
}
return data;
}

event result_t GpsCmd.PowerSet(uint8_t PowerState)
{
    if(PowerState)
    {
        atomic gps_pwr_on = TRUE;
        if (gps_work_factor == 0)
            atomic gps_work_factor = FIRST_WORK_FACTOR;
    }
}

```



```

        else
        {
            atomic gps_pwr_on = FALSE;
            if (gps_work_factor != NORMAL_WORK_FACTOR)
                atomic gps_work_factor = NORMAL_WORK_FACTOR;
        }

        atomic gps_sleep_cnt = 0;
        return SUCCESS;
    }

    event result_t GpsCmd.TxRxSet(uint8_t rtstate)
    {
        // gps tx/rx switches set to on or off
        if (rtstate) //reinit gps uart since its shared with pressure sensor
        {
            call GpsControl.start();
            //start counting time intervals, waiting for gps pkt
            atomic gps_wait_cnt = 0;
        }
        return SUCCESS;
    }
#endif

/*****
* Battery Ref or thermistor data ready
*****/
    async event result_t ADCBATT.dataReady(uint16_t data)
    {
        pack->xData.data1.vref = data ;
        atomic state = BATT_DONE;
        return SUCCESS;
    }

/*****
* Intersema MS5534A barometric pressure/temperature sensor
* - 6 cal coefficients (C1..C6) are extracted from 4,16 bit, words from sensor
* - Temperature measurement:
*    $UT1 = 8 * C5 + 20224$ 
*    $dT = data - UT1$ 
*    $Temp = (degC \times 10) = 200 + dT(C6 + 50) / 1024$ 
* - Pressure measurement:
*    $OFF = C2 * 4 + ((C4 - 512) * dT) / 1024$ 
*    $SENS = C1 + (C3 * dT) / 1024 + 24576$ 
*    $X = (SENS * (PressureData - 7168)) / 16384 - OFF$ 
*    $Press(mbar) = X / 32 + 250$ 
*****/
    async event result_t IntersemaPressure.dataReady(uint16_t data)
    {

```

```

        pack->xData.data1.pressure = data ;
        atomic sensor_state = SENSOR_PRESSURE_GETTEMPDATA;
        return call IntersemaTemp.getData();
    }

    async event result_t IntersemaTemp.dataReady(uint16_t data)
    {
        pack->xData.data1.intersematemp = data ;
        post stopPressureControl();
        return SUCCESS;
    }

    event result_t IntersemaCal.dataReady(char word, uint16_t value)
    {
        // make sure we get all the calibration bytes
        count++;

        calibration[word-1] = value;

        if (count == 4)
        {
            pack->xData.data1.cal_word1 = calibration[0];
            pack->xData.data1.cal_word2 = calibration[1];
            pack->xData.data1.cal_word3 = calibration[2];
            pack->xData.data1.cal_word4 = calibration[3];

            atomic sensor_state = SENSOR_PRESSURE_GETPRESSDATA;
            call IntersemaPressure.getData();
        }
        return SUCCESS;
    }

    event result_t PressureControl.initDone()
    {
        return SUCCESS;
    }

    event result_t PressureControl.stopDone()
    {
        atomic state = PRESSURE_DONE;
        return SUCCESS;
    }

    event result_t PressureControl.startDone()
    {
        uint16_t i;
        for (i=0; i<300;i++)
        {
            TOSH_uwait(1000);
        }
    }

```

```

        count = 0;
        atomic sensor_state = SENSOR_PRESSURE_GETCAL;
        call IntersemaCal.getData();
        return SUCCESS;
    }

/*****
* Sensirion SHT11 humidity/temperature sensor
* - Humidity data is 12 bit:
*   Linear calc (no temp correction)
*    $fRH = -4.0 + 0.0405 * data - 0.0000028 * data^2$    'RH linear
*   With temperature correction:
*    $fRH = (fTemp - 25) * (0.01 + 0.00008 * data) + fRH$    'RH true
* - Temperature data is 14 bit
*    $Temp(degC) = -38.4 + 0.0098 * data$ 
*****/
    async event result_t Temperature.dataReady(uint16_t data)
    {
        pack->xData.data1.temperature = data ;
        post stopTempHumControl();
        return SUCCESS;
    }

    async event result_t Humidity.dataReady(uint16_t data)
    {
        pack->xData.data1.humidity = data ;
        atomic sensor_state = SENSOR_HUMIDITY_GETTEMPDATA;
        return call Temperature.getData();
    }

    event result_t TempHumControl.startDone()
    {
        atomic sensor_state = SENSOR_HUMIDITY_GETHUMDATA;
        call Humidity.getData();
        return SUCCESS;
    }

    event result_t TempHumControl.initDone()
    {
        return SUCCESS;
    }

    event result_t TempHumControl.stopDone()
    {
        atomic state = HUMIDITY_DONE;
        return SUCCESS;
    }

    event result_t HumidityError.error(uint8_t token)
    {

```

```

        call Temperature.getData();
        return SUCCESS;
    }

    event result_t TemperatureError.error(uint8_t token)
    {
        call TempHumControl.stop();
        atomic state = HUMIDITY_DONE;
        return SUCCESS;
    }

/*****
* Taos- tsl2250 light sensor
* Two ADC channels:
*   ADC Count Value (ACNTx) = INT(16.5*[CV-1]) + S*CV
*   where CV = 2^C
*       C = (data & 0x7) >> 4
*       S = data & 0xF
*   Light level (lux) = ACNT0*0.46*(e^-3.13*R)
*       R = ACNT1/ACNT0
*****/
    async event result_t TaosCh1.dataReady(uint16_t data)
    {
        pack->xData.data1.taoch1 = data & 0x00ff;
        post stopTaosControl();
        return SUCCESS;
    }

    async event result_t TaosCh0.dataReady(uint16_t data)
    {
        pack->xData.data1.taoch0 = data & 0x00ff;
        atomic sensor_state = SENSOR_LIGHT_GETCH1DATA;
        return call TaosCh1.getData();
    }

    event result_t TaosControl.startDone()
    {
        atomic sensor_state = SENSOR_LIGHT_GETCH0DATA;
        return call TaosCh0.getData();
    }

    event result_t TaosControl.initDone()
    {
        return SUCCESS;
    }

    event result_t TaosControl.stopDone()
    {
        atomic state = LIGHT_DONE;
        return SUCCESS;
    }

```

```

    }

/*****
* ADXL202E Accelerometer
* At 3.0 supply this sensor's sensitivity is ~167mv/g
* 0 g is at ~1.5V or ~VCC/2 - this varies alot.
* For an accurate calibration measure each axis at +/- 1 g and
* compute the center point (0 g level) as 1/2 of difference.
* Note: this app doesn't measure the battery voltage, it assumes 3.2 volts
* To get better accuracy measure the battery voltage as this effects the
* full scale of the Atmega128 ADC.
* bits/mv = 1024/(1000*VBATT)
* bits/g = 1024/(1000*VBATT)(bits/mv) * 167(mv/g)
* = 171/VBATT (bits/g)
* C = 0.171/VBATT (bits/mg)
* Accel(mg) ~ (ADC DATA - 512) /C
*****/

    async event result_t AccelY.dataReady(uint16_t data)
    {
        pack->xData.data1.accel_y = data & 0xff;
        pack->xSensorHeader.rsvd = data >> 8;
        post powerOffAccel();
        return SUCCESS;
    }

/*****
    async event result_t AccelX.dataReady(uint16_t data)
    {
        pack->xData.data1.accel_x = data;
        atomic sensor_state = SENSOR_ACCEL_GETYDATA;
        call AccelY.getData();
        return SUCCESS;
    }

/*****
    event result_t AccelCmd.SwitchesSet(uint8_t PowerState)
    {
        // power on/off
        if (PowerState)
        {
            call AccelX.getData(); //start measuring X accel axis
            atomic sensor_state = SENSOR_ACCEL_GETXDATA;
        }
        else
        {
            atomic state = ACCEL_DONE;
            iNextPacketID = 1; // issue 1st sensors packet xmit
            atomic WaitingForSend = TRUE;
        }
    }

```

```

        return SUCCESS;
    }

/*****
* Radio msg xmitted.
*****/
event result_t Send.sendDone(TOS_MsgPtr msg, result_t success)
{
    call Leds.yellowOff();
    if(IsUART)
    {
        msg_ptr = msg;
        IsUART = !IsUART;    // change to radio send
        WaitingForSend = TRUE; // uart sent, issue radio send
        sending_packet = FALSE;
    }
    else
    {
        IsUART = !IsUART; // change to uart send
        atomic
        {
            WaitingForSend = FALSE; // both uart and radio sent,
done for current msg
            sending_packet = FALSE;
        }
    }
    return SUCCESS;
}

/*****
* Uart msg rcvd.
* This app doesn't respond to any incoming uart msg
* Just return
*****/
event TOS_MsgPtr Receive.receive(TOS_MsgPtr m)
{
    precv = (XDataMsg *)m->data;
    atomic node_rx = precv->xSensorHeader.node_id;
    atomic pack->xData.dataGps.Txnoid = node_rx;
    atomic rssi = call HPLRF230.readReg(RG_PHY_ED_LEVEL);
    atomic pack->xData.dataGps.rssi = rssi;

    post SendData();
    call Leds.redToggle();
}

#if FEATURE_EEPROM_TEST

/*****

```

```

* test EEPROM, chrl 20060724
*****/
event result_t MTS420EEPROM.writePacketDone(bool result)
{
    if(result)
    {
        atomic EEPROMwf=1;
        atomic EEPROMchkover=1;
        atomic state = EEPROM_READ;
        return SUCCESS;
    }
    return FAIL;
}

event result_t MTS420EEPROM.readPacketDone(char length, char* data)
{
    int i;
    if(length!=10)
        return FAIL;
    if(EEPROMwf>1)
        return SUCCESS;
    EEPROMwf=EEPROMwf+1;
    if(sending_packet)
        return SUCCESS;

    call MTS420EEPROMControl.stop();
    atomic pack->xSensorHeader.rsvd = 1;
    for(i=0;i<10;i++)
    {
        atomic {((uint8_t*)pack)[i+4]=WData[i];}
        atomic {((uint8_t*)pack)[i+14]=data[i];}
        if((uint8_t)data[i] != WData[i])
            pack->xSensorHeader.rsvd = 0;

        WData[i] = WData[i]+1;
    }
    // ((uint8_t*)pack)[25] = ((counter>>8)&0xff);
    // ((uint8_t*)pack)[24] = (counter&0xff);

    // counter++;
    atomic iNextPacketID = 3; // issue 3rd sensors packet xmit
    atomic WaitingForSend = TRUE;
    atomic EEPROMchkover=1;
    atomic state = START;
    return SUCCESS;
}
#endif
}

```

### A.2.3 Estructura del mensaje.

```
/*
 * Copyright (c) 2004-2007 Crossbow Technology, Inc.
 * All rights reserved.
 * See license.txt file included with the distribution.
 *
 * $Id: sensorboardApp.h,v 1.1.2.2 2007/04/26 20:35:48 njain Exp $
 */

#define MAKE_GPS_ENA_OUTPUT() sbi(DDRE,8)//6)
#define SET_GPS_ENA() cbi(PORTE,8)//6)
#define CLR_GPS_ENA() sbi(PORTE,8)//6)

#define GPS_MSG_LENGTH 100
#define GPS_CHAR 11
#define GGA_FIELDS 8
#define GPS_CHAR_PER_FIELD 10
#define GPS_DELIMITER ','
#define GPS_END_MSG '*'

#define GPS_MAX_WAIT 5 // max wait time for gps packet =
GPS_MAX_WAIT*TIMER_PERIOD
#define FIRST_WORK_FACTOR 8
#define NORMAL_WORK_FACTOR 1
// REAL_SLEEP_TIME(ms) = SLEEP_INTERVAL * gps_work_factor *
TIMER_PERIOD, gps_work_factor = 0,NORMAL_WORK_FACTOR
#define GPS_SLEEP_INTERVAL 150
// REAL_WORK_TIME(ms) = WORK_INTERVAL * gps_work_factor *
TIMER_PERIOD, gps_work_factor = FIRST_WORK_FACTOR,NORMAL_WORK_FACTOR =
#define GPS_WORK_INTERVAL 30

typedef struct XSensorHeader
{
    uint8_t board_id;
    uint8_t packet_id; // 3
    uint8_t node_id;
    uint8_t rsvd;
}__attribute__((packed)) XSensorHeader;

typedef struct GGAMsg
{
    uint8_t hour;
    uint8_t minute;
    uint8_t lat_deg;
    uint8_t long_deg;
    uint32_t dec_sec;
    uint32_t lat_dec_min;
    uint32_t long_dec_min;
```



```

    uint8_t nsewind;
    uint8_t fixed;
    uint8_t Txnodeid;
    uint8_t rssi;
//  uint8_t SVs;
} __attribute__((packed)) GGAMsg;

typedef struct XSensorMTS400DataMsg
{
    uint16_t vref;
    uint16_t humidity;
    uint16_t temperature;
    uint16_t cal_word1;
    uint16_t cal_word2;
    uint16_t cal_word3;
    uint16_t cal_word4;
    uint16_t intersematemp;
    uint16_t pressure;
    uint16_t taoch0;
    uint16_t taoch1;
    uint16_t accel_x;
    uint16_t accel_y;
} __attribute__((packed)) XSensorMTS400DataMsg;

typedef struct EEPROMData
{
    uint8_t EEPROMData[10];
} __attribute__((packed)) EEPROMData;

enum
{
    AM_XSXMSG = 0,
};

typedef struct XDataMsg
{
    XSensorHeader xSensorHeader;
    union
    {
        XSensorMTS400DataMsg data1;
        GGAMsg dataGps;
        EEPROMData dataE2prom;
    }xData;
} __attribute__((packed)) XDataMsg;

```





## B ANEXO 2: Programación en MATLAB.

Este **ANEXO** incluye todas las funciones creadas en MATLAB y el archivo principal (Main) que permite simular el sistema de posicionamiento híbrido a partir de los archivos XSniffer generados con los datos recogidos en la WSN. Se muestra el código detalladamente comentado para cada uno de los ficheros de código MATLAB generados. Todos estos ficheros han sido explicados en el **Capítulo 4** y tienen como objetivo proporcionar la funcionalidad de posicionamiento básico además de implementar las técnicas híbridas propuestas en la memoria. Además se incluye el código necesario para obtener las estimaciones del error RSS y el posicionamiento híbrido obtenido al utilizar estas estimaciones del error, este código queda reflejado en el archivo *pruebas.m*.

### B.1 GPS.m

```
% -----  
% [x,y] = GPS(data)  
%  
% Descripción: Función para convertir lat/lon obtenida del GPS de la WSN y  
% almacenadas en archivos XSniffer en coordenadas UTM. Una vez tratados los  
% datos de XSniffer se utiliza la función deg2utm de Rafael Palacios.  
%  
% Entradas:  
%  
% data: variable creada a partir de los datos de XSniffer que no  
% contiene su cabecera y únicamente dispone de los mensajes de la mota  
% 0, que es la que obtiene el posicionamiento GPS y cuyos mensajes  
% tienen el siguiente formato:  
%  
%     uint8_t board_id;(INDEX=1)  
%     uint8_t packet_id;  
%     uint8_t node_id;  
%     uint8_t rsvd;  
%     uint8_t hour;(INDEX=5)  
%     uint8_t minute;  
%     uint8_t lat_deg;  
%     uint8_t long_deg;  
%     uint32_t dec_sec;  
%     uint32_t lat_dec_min;  
%     uint32_t long_dec_min;  
%     uint8_t nsewind;  
%     uint8_t fixed;(INDEX=22)  
%     uint8_t Txnodeid;  
%     uint8_t rssi;  
%
```

```

% Salidas:
%   x, y: Coordenadas UTM de las lat/lon disponibles en data.
%
% Autor: Daniel Egea
%   Daniel Egea
%   Universidad Autónoma de Barcelona (UAB)
%   Barcelona, España
%-----

function [x,y]=GPS(data)

    GGAMsg=data(:,5:22);%Selecciona únicamente el mensaje GPS (GGAMsg)

    %%%%%%%%%%% Formato de GGAMsg %%%%%%%%%%%
    %
    % uint8_t hour;(INDEX=1)      %
    % uint8_t minute;(INDEX=2)   %
    % uint8_t lat_deg;(INDEX=3)   %
    % uint8_t long_deg;(INDEX=4)  %
    % uint32_t dec_sec;(INDEX=5:8) %
    % uint32_t lat_dec_min;(INDEX=9:12) %
    % uint32_t long_dec_min;(INDEX=13:16)%
    % uint8_t nsewind;(INDEX=17)  %
    % uint8_t fixed;(INDEX=18)    %
    %
    %%%%%%%%%%%

    lat_deg=GGAMsg(:,3);%latitud en grados (N/S)
    long_deg=GGAMsg(:,4);%longitud en grados (E/W)

    %Ojo que los campos siguientes son uint32-> 4 números de 1 Byte.
    %HAY QUE PASAR LOS 4 TERMINOS A BINARIO CON 8 BITS Y TENER EN
    CUENTA
    %QUE EL "MSB" ES EL ÚLTIMO (+ A LA DERECHA)SEGÚN EL FORMATO
    %LITTLE-ENDIAN

    %Paso a binario los 4 términos
    lat_0=dec2bin(GGAMsg(:,9),8);%Conversión binario con 8 bits
    lat_1=dec2bin(GGAMsg(:,10),8);%Conversión binario con 8 bits
    lat_2=dec2bin(GGAMsg(:,11),8);%Conversión binario con 8 bits
    lat_3=dec2bin(GGAMsg(:,12));

    long_0=dec2bin(GGAMsg(:,13),8);%Conversión binario con 8 bits
    long_1=dec2bin(GGAMsg(:,14),8);%Conversión binario con 8 bits
    long_2=dec2bin(GGAMsg(:,15),8);%Conversión binario con 8 bits
    long_3=dec2bin(GGAMsg(:,16));

    %FORMATO LITTLE-ENDIAN

```

```

%Concateno los 4 términos teniendo en cuenta el "MSB"
lat=strcat(lat_3,lat_2,lat_1,lat_0);
long=strcat(long_3,long_2,long_1,long_0);

%Obtengo el valor en decimal
lat_dec_min=bin2dec(lat)/10000;%latitud "minutos" (N/S),,divido por 10000 para
pasar a decimal (racional)
lat_deg_min=lat_dec_min/60;%Conversión minutos a grado

long_dec_min=bin2dec(long)/10000;%longitud "minutos" (E/W),,divido por 10000
para pasar a decimal (racional)
long_deg_min=long_dec_min/60;%Conversión minutos a grado

%Latitud y Longitud en unidades de grados
Lat=lat_deg+lat_deg_min;
Lon=long_deg+long_deg_min;

%AHORA TENEMOS COORDENADAS ESFÉRICAS (LAT/LONG)
REFERENCIADAS SEGÚN
%N/S Y E/W, PARA PODER HACER EL CAMBIO A COORDENADAS UMTS
(X/Y) NECESITAMOS
%CAMBIAR LAS REFERENCIAS HACIENDO LAT/LONG NEGATIVAS SI ES
EL CASO (S/W -).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CAMBIO DE REFERENCIA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% nsewid=>000x000y, donde x:N/S e y:E/W,ej:000N000E=00010000
% Posibles estados: 000N000E=00010000=16
%           000N000W=00010001=17
%           000S000E=00000000=0
%           000S000W=00000001=1
% Si S=> Lat*(-1); Si W=> Long*(-1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

nsewid=GGAMsg(:,17);%referencia n/s-e/w

for i=1:length(nsewid)

    if(nsewid(i) == 17)%N-W=>Long. negativa
        Lon(i)=Lon(i)*(-1);

    end

    if(nsewid(i) == 0)%S-E=>Lat. negativa
        Lat(i)=Lat(i)*(-1);

    end

    if(nsewid(i) == 1)%S-W=>Lat. y Long. negativas

```

```

        Lat(i)=Lat(i)*(-1);
        Lon(i)=Lon(i)*(-1);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TRANSFORMACIÓN DE
COORDENADAS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Se utiliza la función deg2utm, en el fichero deg2utm.m del autor:
% Rafael Palacios
% Universidad Pontificia Comillas
% Madrid, Spain

[x,y,~] = deg2utm(Lat,Lon);%Lat y Lon en grados
End

```

## B.2 triangulación.m

```

% -----
% [poscx_est,poscy_est,distm,nodo]=triangulacion(info,x_anchor,y_anchor,n,gama)
%
% Descripción: Función para obtener el posicionamiento a partir de medidas
% RSS tomadas en la WSN y almacenadas en archivos XSniffer.
%
% Entradas:
%
% info: variable creada a partir de los datos de XSniffer que no
% contiene su cabecera y únicamente dispone de los mensajes de la mota
% 0, cuya posición se quiere estimar mediante la información en los
% mensajes transmitido con el siguiente formato:
%
%   uint8_t board_id;
%   uint8_t packet_id; // 3
%   uint8_t node_id;
%   uint8_t rsvd;
%   uint8_t hour;
%   uint8_t minute;
%   uint8_t lat_deg;
%   uint8_t long_deg;
%   uint32_t dec_sec;
%   uint32_t lat_dec_min;
%   uint32_t long_dec_min;
%   uint8_t nsewind;
%   uint8_t fixed;
%   uint8_t Txnodeid;(INDEX=23)
%   uint8_t rssi;(INDEX=24)
%
% x_anchor/y_anchor: Coordenadas de las motas anclas que forman la WSN.
%
% n/gama: Parámetros de propagación utilizados para la estimación de

```

```

% distancia.
%
% Salidas:
%
% poscx_est, poscy_est: Vectores con las coordenadas estimadas a partir
% del conjunto de RSS medidas. Hay que tener en cuenta que para cada
% valor de RSS en los archivos XSniffer se realiza el cálculo de la
% posición con lo que se obtendrán tantas posiciones como el número de
% medidas RSS disponibles.
%
% distm: Distancias estimadas entre la mota a localizar y las tres motas
% anclas con las que se obtiene la localización.
%
% nodo: Variable que contiene los tres nodos con los que se obtiene la
% localización en cada momento.
%
% Autor:
% Daniel Egea
% Universidad Autónoma de Barcelona (UAB)
% Barcelona, España
%-----

function
[poscx_est,poscy_est,distm,nodo]=triangulacion(info,x_anchor,y_anchor,n,gama)
% Dispondremos de dos vectores, uno con varios valores de rssi medidas y
% otro con los correspondientes nodos, es decir, los nodos respecto los que
% se ha medido la rssi. Estos vectores saldrán de los datos de XSniffer.
% De los paquetes que nos pasa XSniffer sólo nos interesan los que nos envíe
% la mota a posicionar, en nuestro caso será nodeid=0.
%
% Formato de info
%
Tx_id=info(:,23);%identificador de nodo correspondiente a la RSS medida.
rssi_values=info(:,24)-45;% Transformación RSS
N=length(rssi_values);%Número de valores de rssi disponibles.

% Se irán calculando las posiciones con las tres mejores rssi a medida que
% se recorre el vector de rssi_values. Hay que controlar que una vez se
% tiene 3 rssi de nodos DISTINTOS empiece a triangular y a partir de ahí
% en cada iteración comprobar si obtenemos una rssi de otro nodo y si es
% mayor cambiarla por la menor de las rssi actuales y si no hay nodos
% nuevos cambiar el valor de la rssi que toque según el nodo del que se
% recibe.

inicio=1;
rssi=zeros(1,3);% Vector con las 3 rssi mayores.
Tx=zeros(1,3);% Vector con los nodoid de las rssi mayores.

% Vectores con las correspondientes posiciones de los nodeid.

```



```

x=zeros(1,3);
y=zeros(1,3);

%Posiciones estimadas
poscx_est=zeros(1,N);
poscy_est=zeros(1,N);
Pe=zeros(2,N);%Matriz con las estimaciones de posición.

%Estimación de distancias.
dism=zeros(3,N);%Matriz con las distancias estimadas utilizadas para la
localización.
nodo=zeros(3,N);%Matriz con el nodo_id de las motas anclas con las que se localiza.

for i=1:length(rssi_values)

    newid=0;
    idnew=Tx_id(i);
    new_rssi=rssi_values(i);
    %Miramos si hay nuevo id
    if((idnew ~= Tx(1)) && (idnew ~= Tx(2)) && (idnew ~= Tx(3)))
        newid=1;
        %Si no es nuevo miramos en que posición de Tx está
    else
        for j=1:length(Tx)
            if(idnew == Tx(j))
                Tx_newid=j;
            end
        end
    end
    %Inicialización de la localización
    if(inicio < 4)
        if(newid == 1)
            Tx(inicio)=idnew;
            rssi(inicio)=new_rssi;
            x(inicio)=x_anchor(idnew);
            y(inicio)=y_anchor(idnew);
            inicio=inicio+1;
        else
            rssi(Tx_newid)=new_rssi;
        end
        %Ahora si hay un nuevo id miramos si la rssi es mayor que alguna de las
        %disponibles y en tal caso se intercambian las rssi, el nodeid y las
        %coordenadas de la mota. Si no hay nuevo id poner las rssi dónde toquen.
    else
        if(newid == 1)
            [rssi_min Tx_min]=min(rssi);%Buscamos el mínimo y la posición

            %Si hay un nuevo nodo con rssi mayor la intercambio, lo mismo
            %con los nodeid y las posiciones.
            if(new_rssi > rssi_min)

```

```

    rssi(Tx_min)=new_rssi;
    Tx(Tx_min)=idnew;
    x(Tx_min)=x_anchor(idnew);
    y(Tx_min)=y_anchor(idnew);

end

%Si no hay un nuevo nodo poner la rssi dónde toque
else
    rssi(Tx_newid)=new_rssi;
end

%Ahora ya tenemos las 3 rssi disponibles para estimar la distancia y
%con las correspondientes posiciones de las motas transmisoras para
%poder utilizar el método de localización.

dist_dB=(rssi-n)/(-gama);%Estimación distancia
dist=10.^(dist_dB/10);%distancia en metros
dism(:,i)=dist';
nodo(:,i)=Tx';

%Estimación de la localización o posicionamiento
A=[x(1)-x(2) y(1)-y(2);x(1)-x(3) y(1)-y(3)];
B=(1/2)*[x(1)^2-x(2)^2+y(1)^2-y(2)^2+dist(2)^2-dist(1)^2;x(1)^2-
x(3)^2+y(1)^2-y(3)^2+dist(3)^2-dist(1)^2];
Pe(:,i)=inv(transpose(A)*A)*transpose(A)*B;

poscx_est(i)=Pe(1,i);
poscy_est(i)=Pe(2,i);

end

end

end

```

## B.3 filtroMsg.m

```
% -----  
% [poscx_est,poscy_est,x,y,dist,nodo]=filtroMsg(x_anchor,y_anchor,n,gama,data)  
%  
% Descripción: Función para seleccionar los mensajes de la mota a localizar  
% y poder utilizar las funciones que permiten obtener la localización RSS y  
% GPS a partir de los distintos archivos de XSniffer. Esta función permite  
% obtener el posicionamiento RSS con varios parámetros de propagación  
%  
% Entradas:  
%  
% data: variable creada a partir de los archivos XSniffer obtenidos en  
% cada una de las posiciones que se toman las medidas. Esta variable  
% contiene todos los datos transmitidos por la mota a localizar, es  
% decir, tiene las medidas RSS y los datos del mensaje GPS.  
%  
% x_anchor/y_anchor: Coordenadas de las motas anclas que forman la WSN.  
%  
% n/gama: Parámetros de propagación utilizados para la estimación de  
% distancia. En este caso estas variables contienen el conjunto de  
% parámetros que forman la tabla de parámetros obtenida a lo largo del  
% PFC.  
%  
% Salidas:  
%  
% poscx_est, poscy_est: Vectores con las coordenadas estimadas a partir  
% del conjunto de RSS medidas. Hay que tener en cuenta que para cada  
% valor de RSS en los archivos XSniffer se realiza el cálculo de la  
% posición con lo que se obtendrán tantas posiciones como el número de  
% medidas RSS disponibles. Además se obtendrán estas posiciones  
% utilizando los distintos parámetros de propagación disponible en la  
% tabla de parámetros.  
%  
% distm: Distancias estimadas entre la mota a localizar y las tres motas  
% anclas con las que se obtiene la localización.  
%  
% nodo: Variable que contiene los tres nodos con los que se obtiene la  
% localización en cada momento.  
%  
% Autor:  
% Daniel Egea  
% Universidad Autónoma de Barcelona (UAB)  
% Barcelona, España  
% -----
```

**function**

```
[poscx_est,poscy_est,x,y,dist,nodo]=filtroMsg(x_anchor,y_anchor,n,gama,data)
```

```

%Buscamos los índices de los paquetes que vienen de la mota a posicionar
%(nodeid=0).
node_id=data(:,3);
index=find(node_id == 0);
L=length(index);
values=length(data(1,:));
data_nodeid0=zeros(L,values);% Vector con la información del nodo0.

%Tomamos los valores que nos interesan (los que vienen de la mota 0)
for k=1:L

    N=index(k);
    data_nodeid0(k,:)=data(N,:);

end

%Estimamos posición con las rssi disponibles, parámetros de nuestra
%propagación y posiciones de las motas anclas
poscx_est_temp=zeros(length(n),L);
poscy_est_temp=zeros(length(n),L);
dist_temp=zeros(3,L,length(n));
for j=1:length(n)

    [poscx_est_temp(j,:),poscy_est_temp(j,:),dist_temp(:,j),nodo_temp]=triangulacion(dat
a_nodeid0,x_anchor,y_anchor,n(j),gama(j));

end

%Quitamos los ceros por falta de motas a triangular, para que a la hora
%de hacer la media no nos afecte.
nulls=length(find(poscx_est_temp(1,:) == 0));%ceros por inicio<3
poscx_est=poscx_est_temp(:,nulls+1:end);
poscy_est=poscy_est_temp(:,nulls+1:end);
dist=dist_temp(:,nulls+1:end,:);
nodo=nodo_temp(:,nulls+1:end);

%Obtenemos la posición proporcionada por el GPS en grados y coordenadas
%UTM
[x,y]=GPS(data_nodeid0);

end

```

## B.4 hibrido.m

```
% -----  
%  
[Pgps,Pe_pond,Pe_pondi,Pe_BLUE,Pe_BLUEi,error,errori]=hibrido2(Pe,Pe_GPS,trans,  
px,py,GPS_ON)  
%  
% Descripción: Función para obtener el posicionamiento híbrido a partir del  
% posicionamiento básico (RSS y GPS). Además la función proporciona los  
% errores cometidos en las distintas posiciones así como el error promedio  
% para cada una de las técnicas de localización utilizadas.  
%  
% Entradas:  
%  
%   Pe : Posicionamiento RSS.  
%  
%   Pe_GPS : Posicionamiento GPS transformado a UTM referenciado.  
%  
%   trans : Translación necesaria en las coordenadas del GPS.  
%  
%   px, py : Posiciones de medida reales.  
%  
%   GPS_ON : Posiciones de medida en las que se obtiene localización GPS.  
%  
% Salidas:  
%  
%   Pgps : Coordenadas del posicionamiento básico GPS tras tomar la  
%   translación necesaria para que coincida con el sistema de coordenadas  
%   local.  
%  
%   Pe_pond : Posicionamiento obtenido con la técnica híbrida de  
%   ponderación simple utilizando medidas ponderadas.  
%  
%   Pe_BLUE : Posicionamiento obtenido con la técnica híbrida BLUEHE  
%   utilizando medidas ponderadas.  
%  
%   Pe_pondi y Pe_BLUEi : Posicionamiento obtenido con las técnicas  
%   híbridas anteriores utilizando medidas instantáneas reales.  
%  
%   error : errores medios para los distintos métodos.  
%  
%   errori : errores instantáneos para los distintos métodos.  
%  
%   FORMATO ERROR:  
%       [RSSI GPS POND POND I BLUE BLUEI]  
%  
% Autor:  
%   Daniel Egea  
%   Universidad Autónoma de Barcelona (UAB)  
%   Barcelona, España
```

```

%-----
function
[Pgps,Pe_pond,Pe_pondi,Pe_BLUE,Pe_BLUEi,error,errori]=hibrido2(Pe,Pe_GPS,trans,
px,py,GPS_ON)

%Se aplica la translación de coordenadas necesaria para coincidir con
%el sistema de coordenadas del posicionamiento con RSSI.
Pgps(1,:)=Pe_GPS(1,:)*trans(1);
Pgps(2,:)=Pe_GPS(2,:)*trans(2);

Pref=[px;py];%Posiciones reales .
Np=length(px);%Número de posiciones a estimar.

%ERROR RSSI
e_RSSI=sqrt(sum((Pe-Pref).^2));%Error cuadrático medio obtenido en cada posición
estimada con RSSI.
RSSI_ME=mean(e_RSSI);%Error medio obtenido con posicionamiento RSSI.

%ERROR GPS
e_GPS=sqrt(sum((Pgps-Pref).^2));%Error cuadrático medio obtenido en cada
posición medida con GPS.
L=find(GPS_ON);% Posiciones en las que se recibe GPS.
e_m=zeros(1,L);
for i=1:Np
    if(GPS_ON(i) == 1)
        e_m(i)=e_GPS(i);
    end
end
GPS_ME=mean(e_m);% Se toma la media teniendo en cuenta únicamente las
posiciones en las que hay GPS.

%ESTIMACIONES CON PONDERACION/BLUHE & INSTANTANEO

alpha=1/GPS_ME;%Parámetro promediado para la ponderación simple.
alpha_i=1./e_GPS;%Parámetros instantáneos para la ponderación simple.

%Parámetros promediados para BLUHE.
sig1=RSSI_ME;
sig2=GPS_ME;
%Parámetros promediados para BLUHE.
sig1i=e_RSSI;
sig2i=e_GPS;

cnt=1;
Pe_pond=zeros(2,Np);
Pe_BLUE=zeros(2,Np);
Pe_pondi=zeros(2,Np);
Pe_BLUEi=zeros(2,Np);

```

```

for i=1:Np
    if(GPS_ON(i) == 1)
        Pe_pond(:,i)=(1-alpha)*Pe(:,i)+alpha*Pgps(:,i);% PONDERACIÓN SIMPLE
        PROMEDIADA.
        Pe_BLUE(:,i)=(1/(1/sig1+1/sig2))*(Pe(:,i)/sig1+Pgps(:,i)/sig2);%BLUHE
        PROMEDIADO.
        Pe_pondi(:,i)=(1-alphai(L(cnt)))*Pe(:,i)+alphai(L(cnt))*Pgps(:,i);%PONDERACIÓN
        SIMPLE INSTANTÁNEA.

        Pe_BLUEi(:,i)=(1/(1/sig1i(L(cnt))+1/sig2i(L(cnt))))*(Pe(:,i)/sig1i(L(cnt))+Pgps(:,i)/sig
        2i(L(cnt)));%BLUHE INSTANTÁNEO.
        cnt=cnt+1;
    else
        %Si no se dispone de posicionamiento GPS se toma el
        %posicionamiento RSS directamente.
        Pe_pond(:,i)=Pe(:,i);
        Pe_BLUE(:,i)=Pe(:,i);
        Pe_pondi(:,i)=Pe(:,i);
        Pe_BLUEi(:,i)=Pe(:,i);
    end
end

%ERROR PONDERACIÓN
e_pond=sqrt(sum((Pe_pond-Pref).^2));%Error cuadrático medio obtenido en cada
posición estimada con ponderación.
pond_ME=mean(e_pond);%Error medio obtenido con posicionamiento hibrido
ponderando posiciones GPS/RSSI.

%ERROR BLUHE
e_BLUE=sqrt(sum((Pe_BLUE-Pref).^2));%Error cuadrático medio obtenido en cada
posición estimada con BLUHE.
BLUE_ME=mean(e_BLUE);%Error medio obtenido con posicionamiento hibrido
BLUHE.

%INSTANTANEO
e_pondi=sqrt(sum((Pe_pondi-Pref).^2));%Error cuadrático medio obtenido en cada
posición estimada con ponderacion instantanea.
pondi_ME=mean(e_pondi);%Error medio obtenido con posicionamiento hibrido
ponderando instantaneamente.
e_BLUEi=sqrt(sum((Pe_BLUEi-Pref).^2));%Error cuadrático medio obtenido en
cada posición estimada con BLUHE instantaneo.
BLUEi_ME=mean(e_BLUEi);%Error medio obtenido con posicionamiento hibrido
BLUHE instantaneo.

error=[RSSI_ME GPS_ME pond_ME pondi_ME BLUE_ME BLUEi_ME];%errores
medios para los distintos métodos de estimación
errori=[e_RSSI;e_GPS;e_pond;e_pondi;e_BLUE;e_BLUEi];%errores instantáneos
para los distintos métodos de estimación
end

```

## PONDERACIÓN MODIFICADA

```
% -----  
%  
[Pgps,Pe_pond,Pe_pondi,Pe_BLUE,Pe_BLUEi,error,errori]=hibrido_2(Pe,Pe_GPS,trans,px,py,GPS_ON)  
%  
% Descripción: Función para obtener el posicionamiento híbrido a partir del  
% posicionamiento básico (RSS y GPS). Además la función proporciona los  
% errores cometidos en las distintas posiciones así como el error promedio  
% para cada una de las técnicas de localización utilizadas. Básicamente es  
% una modificación de hibrido2 para obtener las estimaciones híbridas de  
% ponderación modificada.  
%  
% Entradas:  
%  
%   Pe : Posicionamiento RSS.  
%  
%   Pe_GPS : Posicionamiento GPS transformado a UTM referenciado.  
%  
%   trans : Translación necesaria en las coordenadas del GPS.  
%  
%   px, py : Posiciones de medida reales.  
%  
%   GPS_ON : Posiciones de medida en las que se obtiene localización GPS.  
%  
% Salidas:  
%  
%   Pgps : Coordenadas del posicionamiento básico GPS tras tomar la  
%   translación necesaria para que coincida con el sistema de coordenadas  
%   local.  
%  
%   Pe_pond : Posicionamiento obtenido con la técnica híbrida de  
%   ponderación modificada utilizando medidas ponderadas.  
%  
%   Pe_BLUE : Posicionamiento obtenido con la técnica híbrida BLUE  
%   utilizando medidas ponderadas.  
%  
%   Pe_pondi y Pe_BLUEi : Posicionamiento obtenido con las técnicas  
%   híbridas anteriores utilizando medidas instantáneas reales.  
%  
%   error : errores medios para los distintos métodos.  
%  
%   errori : errores instantáneos para los distintos métodos.  
%  
%   FORMATO ERROR:  
%   [RSSI GPS POND PONDI BLUE BLUEI]  
%  
% Autor:  
%   Daniel Egea
```



```

% Universidad Autónoma de Barcelona (UAB)
% Barcelona, España
%-----

function
[Pgps,Pe_pond,Pe_pondi,Pe_BLUE,Pe_BLUEi,error,errori]=hibrido_2(Pe,Pe_GPS,trans,px,py,GPS_ON)

%Se aplica la translación de coordenadas necesaria para coincidir con
%el sistema de coordenadas del posicionamiento con RSSI.
Pgps(1,:)=Pe_GPS(1,:)*trans(1);
Pgps(2,:)=Pe_GPS(2,:)*trans(2);
Pref=[px;py];%Posiciones reales.
Np=length(px);%Número de posiciones a estimar.

%ERROR RSSI
e_RSSI=sqrt(sum((Pe-Pref).^2));%Error cuadrático medio obtenido en cada posición
estimada con RSSI.
RSSI_ME=mean(e_RSSI);%Error medio obtenido con posicionamiento RSSI.

%ERROR GPS
e_GPS=sqrt(sum((Pgps-Pref).^2));%Error cuadrático medio obtenido en cada
posición medida con GPS.
L=find(GPS_ON);% Posiciones en las que se recibe GPS.
e_GPSm=zeros(1,length(L));
for i=1:Np
    if(GPS_ON(i) == 1)
        e_GPSm(i)=e_GPS(i);
    end
end
GPS_ME=mean(e_GPSm);

%ESTIMACIONES HÍBRIDAS.
%Parámetro promediado para la ponderación modificada.
beta=RSSI_ME/GPS_ME;%Identificador de cuanto bueno o malo es el gps respecto
rssi.
alpha=beta/2;
if(alpha > 1)
    alpha=1;
end
%Parámetro instantaneo para la ponderación modificada.
betai=e_RSSI/e_GPS;
alphai=betai/2;
for i=1:length(alphai)
    if(alphai(i) > 1)
        alphai(i)=1;
    end
end
sig1=RSSI_ME;
sig2=GPS_ME;

```

```

sig1i=e_RSSI;
sig2i=e_GPS;

cnt=1;
Pe_pond=zeros(2,Np);
Pe_BLUE=zeros(2,Np);
Pe_pondi=zeros(2,Np);
Pe_BLUEi=zeros(2,Np);

for i=1:Np
    if(GPS_ON(i) == 1)
        Pe_pond(:,i)=(1-alpha)*Pe(:,i)+alpha*Pgps(:,i);%PONDERACIÓN
MODIFICADA PROMEDIO.
        Pe_BLUE(:,i)=(1/(1/sig1+1/sig2))*(Pe(:,i)/sig1+Pgps(:,i)/sig2);
        Pe_pondi(:,i)=(1-
alpha(L(cnt)))*Pe(:,i)+alpha(L(cnt))*Pgps(:,i);%PONDERACIÓN MODIFICADA
INSTANTÁNEA.

Pe_BLUEi(:,i)=(1/(1/sig1i(L(cnt))+1/sig2i(L(cnt))))*(Pe(:,i)/sig1i(L(cnt))+Pgps(:,i)/sig
2i(L(cnt)));
        cnt=cnt+1;
    else
        Pe_pond(:,i)=Pe(:,i);
        Pe_BLUE(:,i)=Pe(:,i);
        Pe_pondi(:,i)=Pe(:,i);
        Pe_BLUEi(:,i)=Pe(:,i);
    end
end

%ERROR PONDERACIÓN
e_pond=sqrt(sum((Pe_pond-Pref).^2));%Error cuadrático medio obtenido en cada
posición estimada con ponderación modificada
pond_ME=mean(e_pond);%Error medio ponderación modificada

%ERROR BLUE
e_BLUE=sqrt(sum((Pe_BLUE-Pref).^2));%Error cuadrático medio obtenido en cada
posición estimada con BLUE
BLUE_ME=mean(e_BLUE);%Error medio obtenido con posicionamiento hibrido
BLUE

%INSTANTANEO
e_pondi=sqrt(sum((Pe_pondi-Pref).^2));%Error cuadrático medio obtenido en cada
posición estimada con ponderacion instantánea
pondi_ME=mean(e_pondi);%Error medio obtenido con posicionamiento hibrido
ponderando instantaneamente
e_BLUEi=sqrt(sum((Pe_BLUEi-Pref).^2));
BLUEi_ME=mean(e_BLUEi);

error=[RSSI_ME GPS_ME pond_ME pondi_ME BLUE_ME BLUEi_ME];%errores
medios para los distintos métodos de estimación

```

```

    errori=[e_RSSI;e_GPS;e_pond;e_pondi;e_BLUE;e_BLUEi];%errores instantáneos
para los distintos métodos de estimación
end

```

## B.5 parametros.m

```

% -----
% [n,gamma]=parametros_2(info,L,x_anchor,y_anchor,px,py)
%
% Descripción: Función para obtener parámetros de propagación a partir de
% medidas experimentales de RSS y los datos del escenario de medida.
%
% Entradas:
%
%   info: variable creada a partir de los archivos XSniffer obtenidos en
%   cada una de las posiciones que se toman las medidas. Esta variable
%   contiene todos los datos transmitidos por la mota a localizar, es
%   decir, tiene las medidas RSS y los datos del mensaje GPS.
%
%   x_anchor/y_anchor: Coordenadas de las motas anclas que forman la WSN.
%
%   px/py: Posiciones de medida reales.
%
% Salidas:
%
%   n/gama : Parámetros de propagación obtenidos en el escenario de
%   localización donde se toman las medidas de RSS utilizadas.
%
% Autor:
%   Daniel Egea
%   Universidad Autónoma de Barcelona (UAB)
%   Barcelona, España
% -----

```

```

function [n,gamma]=parametros_2(info,L,x_anchor,y_anchor,px,py)

```

```

    Np=length(px);%Número de posiciones a estimar.

```

```

    %La primera fila de las variables creadas de los archivos de XSniffer no
    %contiene información necesaria y hay que eliminarla para trabajar con las
    %funciones creadas. Simplemente tenerlo en cuenta.

```

```

    %Se calculan las distancias entre motas anclas y mota a estimar en cada una
    %de las posiciones del escenario a tratar.

```

```

    d_m=0;
    rssi_values=0;
    sum1=1;
    for i=1:Np

```

```

dist=sqrt((px(i)-x_anchor).^2+(py(i)-y_anchor).^2);
data=info(1+sum1:L(i)+sum1-1,:);

%Buscamos los índices de los paquetes que vienen de la mota a
%posicionar(nodeid=0).
node_id=data(:,3);

index=find(node_id == 0);
Li=length(index);
values=length(data(1,:));
data_nodeid0=zeros(Li,values);% Vector con la información del nodo0.

%Tomamos los valores que nos interesan (los que vienen de la mota
%0).
for k=1:Li

    N=index(k);
    data_nodeid0(k,:)=data(N,:);

end

Tx_id=data_nodeid0(:,23);
for j=1:length(Tx_id)
    %Los paquetes con sólo gps no los proceso.
    if(Tx_id(j) ~= 0)
        d_m=[d_m dist(Tx_id(j))];
        rssi_values=[rssi_values (data_nodeid0(j,24)-45)'];
    end
end
sum1=sum1+L(i);

end

%Quitamos el primer 0.
rssi_values=rssi_values(2:end);
d_m=d_m(2:end);

%ESTIMACIÓN DE PARÁMETROS
z=transpose(rssi_values);% Vector [Nx1] de potencia (RSSI) medida.
d_m=transpose(d_m);% Vector [Nx1] de distancia (a la que se ha medio la
correspondiente RSSI) en metros.
d_dB=10*log10(d_m);% Vector [Nx1] de distancia en logarítmico.
A=[ones(length(z),1) d_dB];

b=(A'*A)\A'*z; %Parametros estimados [n -gamma].
n=b(1);
gamma=-b(2);

%rEPRESENTACIÓN DE LA POTENCIA RECIBIDA EN FUNCIÓN DE LA
DISTANCIA.
plot(d_dB,rssi_values,'.');

```

```

grid on;

end

```

## B.6 resultados.

```

%%%RESULTADOS%%%
%Mostramos por pantalla los parámetros utilizados y los resultados
%obtenidos en las estimaciones y los errores obtenidos, comparando las
%estimaciones con las posiciones reales del escenario.

fprintf('\n Los mejores parámetros son:\n \n gama=%f \n n=%f
\n',gama(index),n(index));
fprintf('\n Dando como resultado las siguientes posiciones estimadas con RSSI:\n ');
Pe_best
if(Nm > 0)
    fprintf('Posiciones UTM obtenidas tras la translación:\n');
    Pgps
    fprintf('Las estimaciones híbridas:\n');
    Pe_pond
    Pe_BLUE
    fprintf('Sus respectivas estimaciones instantáneas:\n');
    Pe_pondi
    Pe_BLUEi
end
fprintf('Error en cada una de las posiciones para las distintas estimaciones:\n');
errori
fprintf('Errores medios para cada una de las estimaciones:\n');
error

%Dibujamos el escenario y las posiciones estimadas con los distintos
%métodos.
figure;
plot(x_anchor,y_anchor,'ko');
hold on;
plot(px,py,'x');
grid on;
xlabel('coordenada x (m)');
ylabel('coordenada y (m)');
legend('motas ancla','posiciones reales');

figure;
plot(px,py,'x');
hold on;
plot(x_anchor,y_anchor,'o');
hold on;
plot(Pe_best(1,:),Pe_best(2:),'gx');
grid on;
if(Nm == 0)

```

```

xlabel('coordenada x (m)');
ylabel('coordenada y (m)');
legend('posiciones reales','motas ancla','estimaciones RSSI');
title('Representación del escenario y las estimaciones RSSI');
else
    hold on;
    plot(Pgps(1,:),Pgps(2:),'kx');
    xlabel('coordenada x (m)');
    ylabel('coordenada y (m)');
    legend('posiciones reales','motas ancla','estimaciones RSSI','coordenadas GPS');
    title('Representación del escenario y las estimaciones RSSI & GPS');
    grid on;

    figure;
    plot(px,py,'x');
    hold on;
    plot(x_anchor,y_anchor,'o');
    hold on;
    plot(Pe_pond(1,:),Pe_pond(2:),'gx');
    hold on;
    plot(Pe_pondm(1,:),Pe_pondm(2:),'kx');
    hold on;
    plot(Pe_BLUE(1,:),Pe_BLUE(2:),'rx');
    xlabel('coordenada x (m)');
    ylabel('coordenada y (m)');
    legend('posiciones reales','motas ancla','estimaciones ponderación','ponedación
modificada','estimaciones BLUHE');
    title('Representación del escenario y las estimaciones "promedio"');
    grid on;

    figure;
    plot(px,py,'x');
    hold on;
    plot(x_anchor,y_anchor,'o');
    hold on;
    plot(Pe_pondi(1,:),Pe_pondi(2:),'gx');
    hold on;
    plot(Pe_pondim(1,:),Pe_pondim(2:),'kx');
    hold on;
    plot(Pe_BLUEi(1,:),Pe_BLUEi(2:),'rx');
    xlabel('coordenada x (m)');
    ylabel('coordenada y (m)');
    legend('posiciones reales','motas ancla','estimaciones ponderación','ponedación
modificada','estimaciones BLUHE');
    title('Representación del escenario y las estimaciones "instantáneas"');
    grid on;

    figure;
    plot(Pe_best(1,:),Pe_best(2:),'ko');
    hold on;

```

```

plot(Pgps(1,:),Pgps(2,:),'kx');
hold on;
plot(Pe_pond(1,:),Pe_pond(2,:),'rx');
hold on;
plot(Pe_pondm(1,:),Pe_pondm(2,:),'x');
hold on;
plot(Pe_BLUE(1,:),Pe_BLUE(2,:),'gx');
xlabel('coordenada x (m)');
ylabel('coordenada y (m)');
legend('RSS','GPS','ponderación','ponedación modificada','BLUHE');
title('Estimaciones RSS&GPS vs Estimaciones híbridas');
grid on;

figure;
plot(Pe_best(1,:),Pe_best(2,:),'ko');
hold on;
plot(Pgps(1,:),Pgps(2,:),'kx');
hold on;
plot(Pe_pondi(1,:),Pe_pondi(2,:),'rx');
hold on;
plot(Pe_pondim(1,:),Pe_pondim(2,:),'x');
hold on;
plot(Pe_BLUEi(1,:),Pe_BLUEi(2,:),'gx');
xlabel('coordenada x (m)');
ylabel('coordenada y (m)');
legend('RSS','GPS','ponderación','ponedación modificada','BLUHE');
title('Estimaciones RSS&GPS vs Estimaciones híbridas inst. ');
grid on;

```

end

## B.7 Archivo Main.

```

clear all;
close all;
clc;

a=input('Directorio del escenario: \n');

%%--- DEFINICIÓN DEL ESCENARIO. ---%%

% Se debe estar en el directorio del escenario en cuestión y se deben tener
% guardadas las variables correspondientes a las posiciones de las motas
% anclas y las posiciones reales a estimar.
cd (a);

%Motas anclas en nuestro sistema de referencia, la posición del vector se
%corresponde con el nodeid.

```

```

load('x_anchor');
load('y_anchor');
load('Nm');

%Posiciones a estimar.
load('px');
load('py');
Np=length(px);%Número de posiciones a estimar.

%Posiciones GPS de las motas anclas, si se disponen. La variable generada
%contiene las distintas posiciones GPS de las motas anclas, además se
%incluye otra variable con el número de filas correspondientes a cada una
%de las motas anclas. También se incluye otra variable que indica las
%posiciones en las que se recibe GPS, básicamente se indican poniendo la
%posición del vector a 1 y las posiciones en las que no se recibe GPS se
%pone a 0.
if(Nm > 0)
    load('anchor_nodes');
    load('Lm');
    load('GPS_ON');
end

%Datos disponibles, capturados con XSniffer de la WSN para distintas
%posiciones. La variable generada contiene las distintas posiciones, además
%se incluye otra variable con el número de filas correspondientes a cada
%una de las posiciones.
load('data');
load('L');

%Cambiamos al directorio dónde se encuentran las funciones a utilizar. El
%directorio de las funciones debe estar en el directorio anterior al que se
%encuentran las distintas variables del escenario en cuestión.
cd ..;

%-- PARÁMETROS DEL MODELO DE PROPAGACIÓN --%

%- Tabla de parámetros disponibles -%
%Se dispone de varios parámetros de propagación almacenados en unas
%variables (n, gama).
load('n');
load('gama');

%Por si incluimos un escenario nuevo llamaremos a la función parámetros que
%nos estima los parámetros de propagación para el escenario en cuestión,
%después se buscarán estos parámetros en la tabla de parámetros disponibles
%y si es un escenario nuevo se incluirá en dicha tabla para proceder a la
%estimación de posición y quedarnos con los parámetros que proporcionan
%mejores resultados.
[ni,gamai]=parametros_2(data,L,x_anchor,y_anchor,px,py);
%Miramos si es un nuevo escenario básicamente mirando si los parámetros

```



```

%obtenidos se encuentran en la tabla de parámetros disponibles.
if(isempty(find(n == ni,1)))
    n=[n ni];
    gama=[gama gamai];
    %Se guardan los nuevos parámetros de propagación:
    save('n','n');
    save('gama','gama');
end
Nn=length(n);%Número de parámetros de propagación disponibles.

%La primera fila de las variables creadas de los archivos de XSniffer no
%contiene información necesaria y hay que eliminarla para trabajar con las
%funciones creadas. Simplemente tenerlo en cuenta.

%-- TRANSFORMACIÓN DE COORDENADAS GPS (grados) A COORDENADAS
UTM(m)(anclas)
if(Nm > 0)
    x_anch_UTM=zeros(1,Nm);
    y_anch_UTM=zeros(1,Nm);
    sum1=1;
    for i=1:Nm

        [x,y]=GPS(anchor_nodes(1+sum1:Lm(i)+sum1-1,:));
        x_anch_UTM(i)=mean(x);
        y_anch_UTM(i)=mean(y);
        sum1=sum1+Lm(i);
        clear x y;
    end
end

%%--- ESTIMACIONES DE POSICIONES CON RSSI Y TRANSFORMACIÓN A
UTM_REF ---%%

%Ahora se calculan las posiciones estimadas con RSSI y las obtenidas de
%GPS, al disponer archivos de posiciones estáticas tomaremos la MEDIA de
%las distintas posiciones estimadas y medidas. Para el caso de la
%aplicación real lo que se haría sería ir proporcionando posiciones cada
%cierto tiempo tomando la media de las posiciones estimadas durante ese
%tiempo.

Pe=zeros(2*Nn,Np);%Matriz con las posiciones estimadas para los distintos
parámetros.
Pe_GPS=zeros(2,Np);%Matriz con las posiciones UTM (referenciadas) transformadas
del GPS.
nodo=zeros(3,Np);
sum1=1;
for i=1:Np

    [poscx_est,poscy_est,x,y,~,nodo_temp]=filtroMsg(x_anchor,y_anchor,n,gama,data(1+s
um1:L(i)+sum1-1,:));

```

```

sum1=sum1+L(i);
nodo(:,i)=mean(nodo_temp,2);
%ESTIMACIÓN_RSSI (con los distintos parámetros de propagación)
x_est=mean(poscx_est,2);% Vector de coordenadas x estimadas [Nn X 1]
y_est=mean(poscy_est,2);% Vector de coordenadas y estimadas [Nn X 1]

%Construimos Pe.
cnt=1;
for ii=1:Nn

    Pe(cnt:cnt+1,i)=[x_est(ii);y_est(ii)];
    cnt=cnt+2;
end

%GPS_UTM_REF
if(Nm > 0)
    Pe_GPS(1,i)=mean(x)-x_anch_UTM(1);
    Pe_GPS(2,i)=mean(y)-y_anch_UTM(1);
end
clear x y poscx_est poscy_est;
end

%%--- ELECCIÓN DE LOS MEJORES PARÁMETROS ---%%

%Miramos cuál de los parámetros disponibles proporciona mejores resultados
%en el escenario analizado, para ello se calculan los errores medios
%obtenidos con cada uno de los parámetros y los que den un error menor
%serán los que se utilicen para proporcionar las estimaciones RSSI.

e=zeros(Nn,Np);
ref=[px;py];
for ii=1:Nn

    rest=sum((Pe((2*ii-1):2*ii,:)-ref).^2);
    e(ii,:)=sqrt(rest);
end
mean_error=sum(e,2)./Np;
 [~,index]=min(mean_error);

Pe_best=Pe(2*index-1:2*index,:);

%Una vez disponemos de los mejores parámetros y de las estimaciones de
%posición correspondientes se procede a obtener las distancias estimadas y
%las motas con las que se triangula. Los nodos ya los tenemos en la
%variable nodo.
sum1=1;
dist=zeros(3,Np);
for i=1:Np
    [~,~,~,~,dism,~]=filtroMsg(x_anchor,y_anchor,n(index),gama(index),data(1+sum1:L(i)
    )+sum1-1,:));

```

```

    dist(:,i)=mean(distm,2);
    sum1=sum1+L(i);
end

%%--- CÁLCULO DE LAS ESTIMACIONES HÍBRIDAS Y LOS ERRORES ---%%

%Antes de llamar a la función que calcula las distintas estimaciones
%híbridas y proporciona los errores obtenidos, se debe calcular la
%translación del sistema de coordenadas GPS respecto el nuestro (RSSI).

if(Nm > 0)
    if(Nm == length(x_anchor))

        ref_anch=[x_anchor;y_anchor];
        ref_anch_UTM=[x_anch_UTM-x_anch_UTM(1);y_anch_UTM-
y_anch_UTM(1)];
        suma=sum(sign(ref_anch)+sign(ref_anch_UTM),2)-1;
        translacion=sign(suma);%translación a aplicar en el sist. coords. del GPS.
    else
        suma=sum((sign([px;py])+sign(Pe_GPS))-1,2);
        translacion=sign(suma);
    end

    %Una vez obtenida la translación a aplicar a las posiciones del GPS
    %(Pe_GPS) y en posesión de las mejores estimaciones (según los parámetros)
    %con RSSI se llama a la función hibrido que proporciona las estimaciones
    %híbridas (ponderación, BLUE y sus respectivos instantáneos),así como los
    %errores correspondientes a estas estimaciones.

    [Pgps,Pe_pond,Pe_pondi,Pe_BLUE,Pe_BLUEi,error,errori]=hibrido2(Pe_best,Pe_GPS,
    translacion,px,py,GPS_ON);

    [~,Pe_pondm,Pe_pondim,~,~,errorm,errorim]=hibrido_2(Pe_best,Pe_GPS,translacion,p
    x,py,GPS_ON);
    error=[error errorm(3:4)];
    errori=[errori;errorim(3:4,:)];
else
    errori=e(index,:);
    error=mean_error(index);
end

%%--- PRESENTACIÓN DE LOS RESULTADOS --%%

%Mostramos los resultados obtenidos y la representación del escenario así
%como de las estimaciones obtenidas, llamando al script resultados.

resultados;

```

## B.8 Pruebas.m

```

Pref=[x_anchor;y_anchor];
% Pref_UTM=[x_anch_UTM;y_anch_UTM];
Preal=[px;py];
Np=size(Preal,2);
Nm=size(Pref,2);
e_RSS_r=errori(1,:);
e_GPS=errori(2,:);
% nodo=[2 2 4 1 2 4 4 4 2;3 1 2 3 1 1 1 1 4;1 3 3 4 3 2 2 2 1];%E5
nodo=[2 2 4 4 3;3 3 2 2 1;1 1 3 3 4];%E4_2
% nodo=[4 1 1 4 2 4 2 1 4;3 3 3 3 3 2 3 2 2;1 4 4 1 4 3 4 4 3];%E6

dist_r=zeros(3,Np);
dist_p=zeros(3,Np);
for i=1:Np
    for ii=1:3
        dist_r(ii,i)=sqrt(sum((Preal(:,i)-Pref(:,nodo(ii,i))).^2));
        dist_p(ii,i)=sqrt(sum((Pe_best(:,i)-Pref(:,nodo(ii,i))).^2));
    end
end

err_d_r=dist-dist_r;
F=sqrt(mean((dist_p-dist).^2,1));

%Curve fitting LS error
z=e_RSS_r/F;
R=[(mean(dist,1)).' ones(Np,1)];
m=(R'*R)\R'*z.'; %Parametros estimados [k r0]

mdist=R(:,1);%Distancias estimadas promediadas en cada una de las posiciones.
r=sqrt(abs(m(1).*mdist+m(2)));

e_RSS_est=F.*r.';

%ESTIMACIONES CON PONDERACION/BLEUE INSTANTANEO
alpha_i=1./e_RSS_est;
% alpha_i=1./e_GPS;
sig1_i=e_RSS_est;
% sig2_i=e_GPS;
sig2_i=error(2)*ones(1,Np);
% beta_i=e_RSS_est./e_GPS;
beta_i=e_RSS_est./error(2);
alphami=beta_i/2;
for i=1:length(alpha_i)
    if(alphaami(i) > 1)
        alphaami(i)=1;
    end
    if(alpha_i(i) > 1)
        alpha_i(i)=1;
    end
end

```

```

end
end

Pe_pondi=zeros(2,Np);
Pe_BLUEi=zeros(2,Np);
Pe_pondmi=zeros(2,Np);
for i=1:Np

    Pe_pondi(:,i)=alpha(i)*Pe_best(:,i)+(1-alpha(i)).*Pgps(:,i);%Con e_RSS
%    Pe_pondi(:,i)=(1-alpha(i))*Pe_best(:,i)+alpha(i).*Pgps(:,i);%Con e_GPS
    Pe_BLUEi(:,i)=(1/(1/sig1i(i)+1/sig2i(i)))*(Pe_best(:,i)/sig1i(i)+Pgps(:,i)/sig2i(i));
    Pe_pondmi(:,i)=(1-alpha(i))*Pe_best(:,i)+alpha(i).*Pgps(:,i);

end

%INSTANTANEO
e_pondi=sqrt(sum((Pe_pondi-Preal).^2));%Error cuadrático medio obtenido en cada
posición estimada con ponderacion instantanea
pondi_ME=mean(e_pondi);%Error medio obtenido con posicionamiento hibrido
ponderando instantaneamente
e_BLUEi=sqrt(sum((Pe_BLUEi-Preal).^2));%Error cuadrático medio obtenido en cada
posición estimada con BLUE instantaneo
BLUEi_ME=mean(e_BLUEi);%Error medio obtenido con posicionamiento hibrido
BLUE instantaneo
e_pondmi=sqrt(sum((Pe_pondmi-Preal).^2));%Error cuadrático medio obtenido en cada
posición estimada con ponderacion instantanea
pondmi_ME=mean(e_pondmi);%Error medio obtenido con ponderación modificada
instantánea.

error_est=[pondi_ME BLUEi_ME pondmi_ME];%errores medios para los distintos
métodos de estimación
errori_est=[e_RSS_r;e_GPS;e_pondi;e_BLUEi;e_pondmi];%errores instantáneos para
los distintos métodos de estimación

```





## C ANEXO 3: Medidas adicionales.

En el siguiente **ANEXO** se muestran todas las medidas adicionales a las presentadas en los capítulos de la memoria, adquiridas a lo largo del transcurso del proyecto.

Entre estos resultados se encuentran:

- Las medidas tomadas para extraer los parámetros de propagación que completan la tabla de parámetros presentada en la memoria. Estas medidas se toman en escenarios ajenos a los tratados en los distintos capítulos.
- Resultados obtenidos del posicionamiento tratado en el PFC en escenarios de localización distintos a los analizados.
- Estimación del error RSS en el escenario de localización 3 (E3) y los resultados del posicionamiento híbrido al utilizar estos errores instantáneos.
- Resultados del posicionamiento híbrido al utilizar las estimaciones del error instantáneo RSS junto con el error GPS instantáneo real.

### C.1 Parámetros de propagación y posicionamiento.

A continuación se presentan los datos relacionados con la extracción de los parámetros de propagación constituyentes de la tabla de parámetros obtenida en el PFC así como resultados de posicionamiento que no se han tratado, prestando especial interés en el posicionamiento en entornos interiores. Igual que en las medidas presentadas en la memoria hay algunas que se toman con las motas en línea recta y otras que se realizan con las motas situadas tal y como si estuvieran en un escenario de localización en concreto.

- Para las medidas en línea recta se presenta:
  - Tabla con la RSS obtenida junto con la distancia a la que se mide.
  - Representación de la potencia obtenida en función de la distancia junto con la aproximación realizada por los parámetros obtenidos.
- Para las medidas en escenarios de localización se presenta:
  - Coordenadas de las posiciones de medida y de las motas anclas.
  - Representación del escenario con el sistema de coordenadas utilizado.



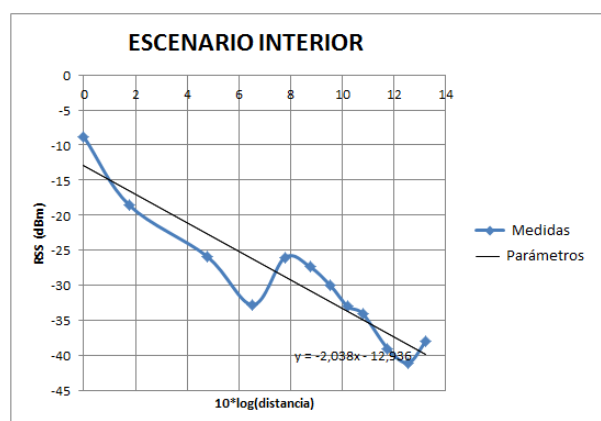
- Representación de las medidas obtenidas en función de la distancia junto con la aproximación realizada por los parámetros de propagación obtenidos.
- Aquellos escenarios de localización en los que se hayan realizado pruebas de posicionamiento se muestran los resultados obtenidos.

### ***C.1.1 Escenario Interior.***

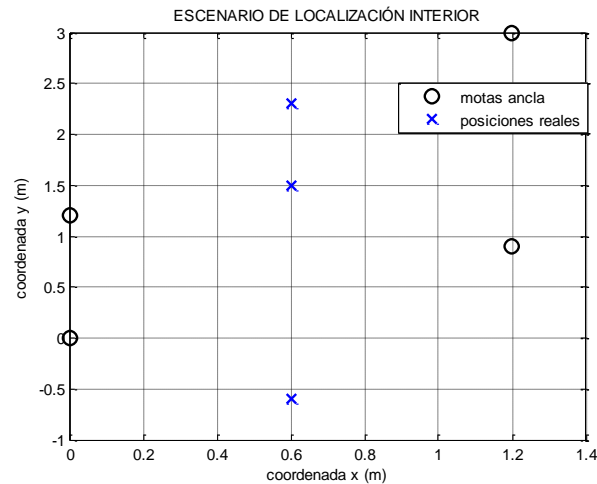
El escenario en el que se toman las medidas mostradas a continuación es un escenario interior localizado en un aparcamiento subterráneo donde se encuentra la presencia de obstáculos como paredes o columnas y vehículos estacionados alrededor de la zona de medida. En la **Tabla 32** indica las medidas de potencia obtenidas en las distintas distancias en las que se han situado la mota para medir la RSS. Con estos resultados se obtiene la descripción de la potencia en función de la distancia mostrada en la **Figura 47** donde además se representa la aproximación obtenida con los parámetros de propagación extraídos de las medidas adquiridas. Estos parámetros de propagación son los correspondientes a los Parámetros 7 de la tabla de parámetros presentada en la memoria.

<b>ESCENARIO INTERIOR</b>		
<b>DISTANCIA</b>	<b>10LOG(dist)</b>	<b>RSSI</b>
1	0	-8,8
1,5	1,76091259	-18,56
3	4,77121255	-25,85
4,5	6,53212514	-32,75
6	7,7815125	-26
7,5	8,75061263	-27,34
9	9,54242509	-30
10,5	10,211893	-33
12	10,7918125	-34
15	11,7609126	-39
18	12,5527251	-41
21	13,2221929	-38

**Tabla 32.** Medidas de potencia en escenario interior.



**Figura 47.** Medidas en escenario interior obteniendo Parámetros 7.



**Figura 48.** Escenario de localización interior.

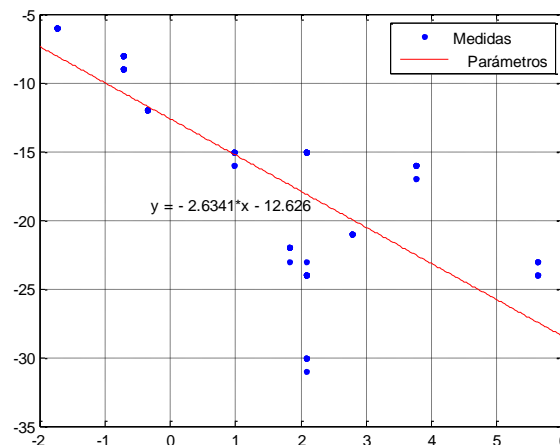
### ***C.1.2 Escenario de localización interior.***

Este escenario de localización se corresponde con el interior de una vivienda, concretamente un pasillo de ésta donde las motas se encuentran considerablemente juntas entre sí. En la **Tabla 33** se indican las posiciones en las que se han tomado las medidas RSS para calcular el posicionamiento y las posiciones de las motas anclas que forman el sistema de coordenadas mostrado en la **Figura 48**

Coordenadas	P1	P2	P3	M1	M2	M3	M4
X	0.6	0.6	0.6	0	0	1.2	1.2
Y	-0.6	1.5	2.3	0	1.2	0.9	0.9

**Tabla 33.** Coordenadas del escenario de localización interior.

Las medidas tomadas en las distintas posiciones que forman el escenario de localización se recogen en la **figura 49**, donde se representan las potencias recibidas en función de la distancia junto la recta representada por los parámetros de propagación extraídos de las medidas que se corresponden con los **Parámetros 8** de la tabla de parámetros presentada.



**Figura 49.** Medidas en escenario interior y Parámetros 8.

Coordenadas	P1_RSS	P2_RSS	P3_RSS
X (m)	2.93	0.49	2.85
Y (m)	0.41	0.63	0.61
Error Inst. (m)	2.54	0.87	2.81

**Tabla 34.** Coordenadas RSS estimadas.

Los parámetros obtenidos con las medidas de la **Tabla 33** no son los que proporcionan los mejores resultados de posicionamiento, siendo los óptimos en este sentido los Parámetros **12** de la tabla de parámetros. Utilizando los Parámetros **12** se obtienen los resultados de la **Tabla 34**, donde únicamente se muestran los resultados del posicionamiento RSS por que al tratarse de un escenario interior el receptor GPS es incapaz de recibir señal. Con estos resultados se obtiene un error medio de posicionamiento de 2.07 metros que para tratarse de un escenario interior es más que aceptable.

### **C.1.3** *Escenario de localización interior 2 (EI2).*

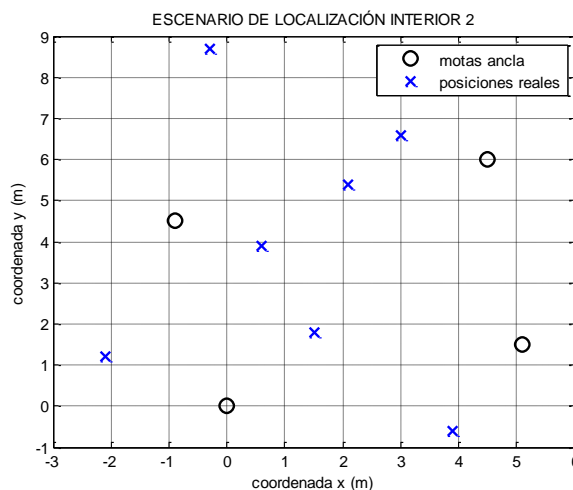
Para corroborar el buen funcionamiento obtenido en el escenario interior anterior se vuelve a realizar otro escenario de localización en el mismo entorno interior pero utilizando un número mayor de posiciones de medida, concretamente estas posiciones de medidas son las indicadas en la **Tabla 35** siendo las coordenadas de las motas anclas que definan el sistema de coordenadas local las indicada en la **Tabla 36** dando lugar al escenario representado en la **Figura 50**.

Coordenadas	P1	P2	P3	P4	P5	P6	P7
X (m)	3.9	-2.1	1.5	0.6	2.1	3	-0.3
Y (m)	-0.6	1.2	1.8	3.9	5.4	6.6	8.7

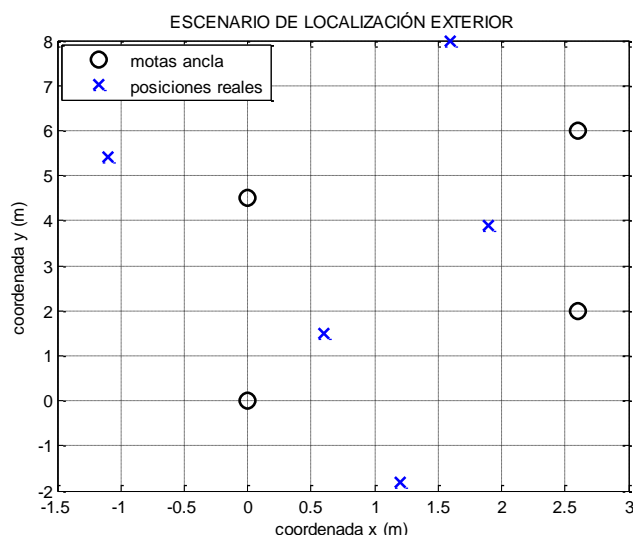
**Tabla 35.** Posiciones de medida del escenario de localización EI2.

Coordenadas	M1	M2	M3	M4
X (m)	0	5.1	-0.9	4.5
Y (m)	0	1.5	4.5	6

**Tabla 36.** Coordenadas de motas anclas en EI2



**Figura 50.** Escenario de localización EI2.



**Figura 51.** Escenario de localización exterior.

Igual que en el caso anterior los parámetros óptimos en el sentido de la precisión de los resultados no son unos parámetros extraídos en un escenario exterior, concretamente los Parámetros **11**, obteniendo los resultados de la **Tabla 37** que proporcionan un error medio del posicionamiento RSS de 3.4 metros que todo y aumentar ligeramente respecto el del anterior escenario sigue siendo satisfactorio para tratarse de un escenario de localización interior.

Coordenadas	P1	P2	P3	P4	P5	P6	P7
X_RSS (m)	1.79	2.02	2.26	2.39	2.10	2.31	0.62
Y_RSS (m)	2.92	2.50	3.20	3.04	4.18	3.58	1.24
Error Inst. (m)	4.10	4.32	1.59	1.99	1.22	3.09	7.51

**Tabla 37.** Coordenadas RSS estimadas.

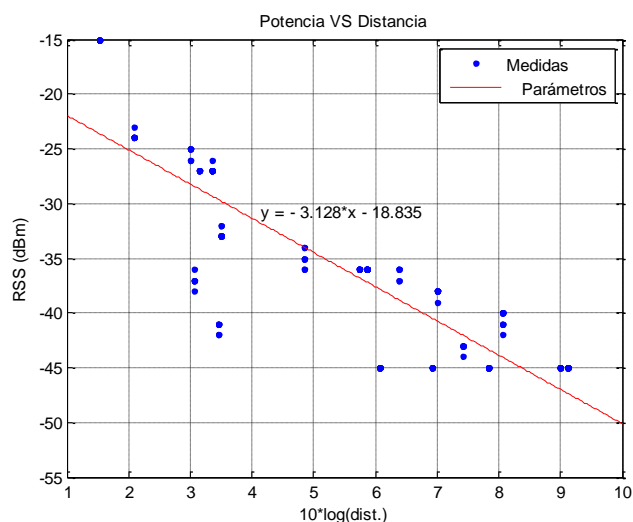
#### ***C.1.4 Escenario de localización exterior.***

El escenario que se presenta a continuación se trata de un escenario exterior parecido a los analizados en la memoria. Concretamente este escenario está situado en la zona del escenario de localización 1 (E1) tratado en la memoria, sin embargo todas las motas de la WSN se encuentran situadas en el suelo. Las coordenadas de las posiciones en que se toman medidas y de las motas anclas se indica en la **Tabla 38**, obteniendo el sistema de coordenadas representado en la **Figura 51**.

Coordenadas	P1	P2	P3	P4	P5	M1	M2	M3	M4
X (m)	1.2	0.6	1.9	-1.1	1.6	0	2.6	0	2.6
Y (m)	-1.8	1.5	3.9	5.4	8	0	2	4.5	6

**Tabla 38.** Coordenadas del escenario de localización exterior.

Las medidas de potencia obtenidas se muestran en la **Figura 52** en función de la distancia donde además se incluye la aproximación de los parámetros extraídos que son los Parámetros **10** de la tabla de parámetros.



**Figura 52.** Medidas en escenario de localización exterior.

En este caso los parámetros óptimos para el posicionamiento RSS son los Parámetros **11** de la tabla de parámetros, que son unos parámetros extraídos en este mismo escenario de localización pero tomando medidas en posiciones diferentes a las utilizadas en este caso. Con estos parámetros se obtienen los resultados de posicionamiento RSS mostrado en la **Tabla 39**, mientras que con GPS se obtienen los resultados de la **Tabla 40**. Para el posicionamiento RSS se obtiene un error medio de 2.66 metros mientras que para el posicionamiento GPS se obtiene un error medio de 3.7 m, tal y como se puede comprobar en la **Tabla 41** donde además se muestran los errores cometidos por las distintas técnicas de localización híbrida tratadas en la memoria.

Coordenadas	P1_RSS	P2_RSS	P3_RSS	P4_RSS	P5_RSS
X (m)	-4.52	0.73	-0.62	-0.16	2.14
Y (m)	0.47	1.45	3.21	4.42	5
Error Inst. (m)	6.16	0.14	2.62	1.36	3.04

**Tabla 39.** Posicionamiento RSS en escenario de localización exterior.

Coordenadas	P1_GPS	P2_GPS	P3_GPS	P4_GPS	P5_GPS
X (m)	2.45	1.92	-3.55	-1.06	-4.31
Y (m)	-2.24	-1.62	2.11	4.33	4.31
Error Inst.	1.33	3.39	5.73	1.06	6.96

**Tabla 40.** Posicionamiento GPS en escenario de localización exterior.

En este escenario el error GPS no es tan grande como el obtenido en los escenarios analizados en la memoria, además la diferencia con el error medio RSS es bastante pequeña. Aun así los resultados obtenidos por las técnicas híbridas promedio (utilizando medidas de precisión promediadas) no proporcionan unos resultados satisfactorios debido básicamente a la medida del error GPS medio obtenida ya que difiere lo suficiente en ciertas posiciones con el error instantáneo cometido como para no permitir el correcto funcionamiento de las técnicas híbridas. Tal y como se ha comprobado a lo largo de la memoria, al utilizar las técnicas híbridas instantáneas siempre se obtiene una mejora del menor de los errores básicos.

ERROR INSTANTANEO						ERROR MEDIO
TÉCNICA	P1	P2	P3	P4	P5	PROMEDIO
RSS	6.16	0.14	2.62	1.36	3.04	2.66
GPS	1.33	3.39	5.73	1.06	6.96	3.7
Ponderación	4.13	0.99	3.46	1.22	3.4	2.64
Ponderación Inst.	0.54	1.07	3.16	1.06	3.11	1.79
*PM	3.46	1.28	3.74	1.18	3.70	2.67
**PMI	1.33	0.19	3.33	1.10	3.26	1.84
BLUE	3.02	1.47	3.92	1.16	3.93	2.7
BLUE Inst.	0.04	0.25	3.6	1.12	3.5	1.7
*PM: Ponderación Modificada; **PMI: PM Instantánea						

**Tabla 41.** Error de posicionamiento para cada técnica de localización.

## C.2 Estimación de error RSS y localización híbrida.

Para completar el **ANEXO 3** se presentan los resultados de la estimación del error RSS en el escenario de localización 3 (E3), presentando además los resultados del posicionamiento híbrido utilizando esta estimación. Por último, para comprobar que la utilización del error GPS medio como medida de precisión del posicionamiento GPS es suficiente, se muestran los resultados del posicionamiento híbrido al utilizar el error GPS instantáneo real y se compara con el posicionamiento obtenido al utilizar el error medio GPS junto con las estimaciones del error RSS.

### C.2.1 Escenario de localización 3 (E3).

En la **Tabla 42** se representan los resultados obtenidos al utilizar la técnica de estimación de errores RSS presentada en el **Capítulo 6**. Además se presenta el error RSS real y el valor promedio, que se comprueba que tanto el error promedio real como el estimado son muy próximos todo y que en ciertas posiciones se obtenga una diferencia considerable (P1 y P9). Utilizando las estimaciones de la **Tabla 42** como medidas de la precisión RSS y el error medio GPS como medida de la precisión del posicionamiento GPS se obtienen los resultados mostrados en la **Tabla 43** del posicionamiento híbrido.

ERROR RSS (m)									PROMEDIO
P1	P2	P3	P4	P5	P6	P7	P8	P9	(m)
8.3	4.21	3.63	2.59	0.84	1.95	8.47	9.61	6.90	5.17
ERROR RSS ESTIMADO (m)									
4.9	5.50	4.75	1.94	0.10	0.82	9.18	8.88	11.26	5.26
DIFERENCIA									
3.40	1.28	1.12	0.66	0.74	1.13	0.71	0.73	4.36	1.57

**Tabla 42.** Error RSS estimado en E3.

ERROR INSTANTANEO										ERROR PROMEDIO
TÉCNICA	P1	P2	P3	P4	P5	P6	P7	P8	P9	
RSS	8.3	4.21	3.63	2.59	0.84	1.95	8.47	9.61	6.90	5.17
GPS	10.57	4.71	5.49	4.98	0.66	5.77	10.29	7.53	7.40	6.38
Pond. Inst.	6.78	3.89	4.34	3.22	0.84	1.95	9.19	7.45	6.36	4.89
*PMI	1.53	3.07	2.94	2.62	0.84	2.19	7.72	7.51	6.04	3.83
**BLUHEI	1.14	3.07	3.01	2.70	0.83	2.37	6.92	7.67	3.64	<b>3.48</b>
PM: Ponderación Modificada; *PMI: PM Instantánea.; **BLUHEI: BLUHE Instantáneo.										

**Tabla 43.** Error de posicionamiento con los errores estimados (E3).

Igual que para los escenarios presentados en la memoria, utilizando las estimaciones del error RSS propuestas en la memoria se consigue reducir el error promedio de las técnicas básicas, en este caso se consigue reducir en un 33% el menor de los errores básicos que en este caso es el error RSS y se corresponde con una reducción de aproximadamente 2 metros.

### ***C.2.2 Localización híbrida con error GPS instantáneo.***

Con los resultados presentados en la memoria se observa que al utilizar las estimaciones del error RSS en las técnicas híbridas no permite obtener el error teórico obtenido al utilizar los errores instantáneos tanto del posicionamiento RSS como del GPS. Se podría pensar que al no disponer de medidas instantáneas del error GPS y utilizar el error promedio provoca el hecho de no conseguir la precisión teórica. Por ello a continuación se presentan los errores obtenidos al utilizar el error GPS instantáneo real junto con las estimaciones del error RSS en las técnicas híbridas, comprándose con los errores obtenidos al utilizar el error GPS promedio y con los errores teóricos (al utilizar errores instantáneos teóricos tanto para GPS como para RSS).

En la **Tabla 44** se muestran los errores promedio para cada uno de los escenarios de localización tratados en la memoria al utilizar el error instantáneo GPS real junto con las estimaciones RSS en las técnicas híbridas. Los resultados de la ponderación simple no se muestran ya que serían los mismos que los mostrados en el **Capítulo 5** (caso de utilizar error instantáneo GPS real) y los del **Capítulo 6** (caso de utilizar las estimaciones del error RSS).

Escenario de localización 1 (E1)	
TÉCNICA	ERROR PROMEDIO (m)
Ponderación Modificada	2.10
BLUHE	2.23
Escenario de localización 2 (E2)	
Ponderación Modificada	3.45
BLUHE	3.31
Escenario de localización 3 (E3)	
Ponderación Modificada	3.87
BLUHE	3.60

**Tabla 44.** Errores promedios utilizando error GPS instantáneo.

En la **Tabla 45** se recogen los errores obtenidos con las técnicas híbridas al utilizar ambos errores instantáneos (RSS y GPS) reales (ERROR PROMEDIO A) y al utilizar las estimaciones del error RSS con el error promedio GPS (ERROR PROMEDIO B). Comparando los errores de la **Tabla 44** con los de la **Tabla 45** se comprueba que el hecho de utilizar los errores instantáneos GPS reales no permite obtener la precisión teórica obtenida al utilizar ambos errores reales (ERROR PROMEDIO A), sin embargo comparando con los errores obtenidos al combinar las estimaciones del error RSS con los errores GPS reales se observa que la diferencia entre los errores no es muy grande, con lo que se puede concluir que no es el hecho de utilizar el error GPS promedio junto las estimaciones del error RSS lo que impide obtener la precisión teórica.

<b>Escenario de localización 1 (E1)</b>		
<b>TÉCNICA</b>	<b>ERROR PROMEDIO A</b>	<b>ERROR PROMEDIO B</b>
Ponderación Modificada	1.61	2.65
BLUHE	1.89	2.65
<b>Escenario de localización 2 (E2)</b>		
Ponderación Modificada	2.73	3.63
BLUHE	2.84	3.32
<b>Escenario de localización 3 (E3)</b>		
Ponderación Modificada	3.39	3.83
BLUHE	3.43	3.48

**Tabla 45.** Comparación de errores promedios según las medidas utilizadas.





**Resum:**

*Aquest projecte estudia la implementació d'un sistema de localització híbrid amb la combinació del posicionament GPS y el posicionament obtingut mitjançant mesures de potència rebuda (RSS) utilitzant una xarxa de sensors sense fils (WSN). Inicialment s'analitzen les característiques principals de les WSN y les tècniques de posicionament GPS y RSS. A continuació es proposen tècniques de localització híbrides que combinen el posicionament bàsic brindat per la WSN (GPS y RSS) per obtenir posicionament tant en escenaris interiors com exteriors a més d'obtenir una precisió major que les precisions bàsiques. Una vegada s'han analitzat els conceptes bàsics y s'han proposat les tècniques a utilitzar en el sistema de localització híbrida s'expliquen els aspectes d'implementació relacionats amb la programació de la WSN. Finalment, després d'analitzar els resultats de diverses mesures, queda present la necessitat de tècniques d'estimació d'error en el posicionament, per això es proposa una tècnica d'estimació d'error per ser utilitzada en les estimacions híbrides y obtenir així el funcionament desitjat de les tècniques híbrides.*

**Resumen:**

*Este proyecto estudia la implementación de un sistema de localización híbrido con la combinación del posicionamiento GPS y el posicionamiento obtenido mediante medidas de potencia recibida (RSS) a partir de una red de sensores inalámbricos (WSN). En primer lugar se analizan las características principales de las WSN y las técnicas de posicionamiento GPS y RSS. A continuación se proponen técnicas de localización híbrida que permitan a partir del posicionamiento básico brindado por la WSN (GPS y RSS) obtener posicionamiento tanto en escenarios interiores como exteriores y obtener una precisión en la localización híbrida mayor que las precisiones básicas. Una vez analizados los conceptos básicos y propuestas las técnicas a utilizar en el sistema de localización híbrida se comentan los aspectos de implementación a tener en cuenta a la hora de la programación de la WSN. Finalmente, tras analizar los resultados de diversas medidas, queda presente la necesidad de técnicas de estimación de errores en el posicionamiento, por ello se propone una técnica de estimación de errores para ser utilizada en las estimaciones híbridas y permita obtener el funcionamiento deseado.*

**Summary:**

*This thesis studies the implementation of a positioning System combining GPS positioning and positioning obtained by the Receive Signal Strength (RSS) measurements from a Wireless Sensor Network (WSN). First of all, the features of the WSN and GPS and RSS positioning techniques are discussed. Then some hybrid localization techniques are proposed which enable indoor and outdoor localization and obtain a hybrid location accurate than the basic accuracies. Finally, after analyze the results of various measures, is needed to obtain estimation techniques for positioning errors, for this a technique is proposed to be used in the hybrid estimations and allow to obtain the desired hybrid techniques performance.*

