

5. Adaptación a Experimentación Real

El Simulador que ha sido desarrollado, implementa una gran cantidad de parámetros con la intención de asemejarse al máximo a la realidad. Sin embargo, es casi imposible prever los miles de factores ambientales y aleatorios a los que un sistema de dichas características puede llegar a ser sometido en condiciones reales.

Es por ello que se ha decidido realizar una adaptación del código generado hasta ahora, para que pueda ser usado en un entorno de experimentación real sobre el que éstas variables si influyan.

El entorno seleccionado para realizar dicho cometido ha sido el Integrated TestBed de CONET⁴⁹, debido a que el mismo se ha diseñado específicamente para albergar sistemas distribuidos como el que en el presente documento se expone.

5.1 El TestBed

El creciente interés de la comunidad académica e industrial en el estudio de la cooperación entre diferentes dispositivos con sensores y sistemas informáticos comunicados entre sí, unido a la baja disponibilidad de herramientas experimentales para la evaluación y comparación de algoritmos de cooperación, ha llevado por parte AICIA⁵⁰, en el marco del proyecto CONET, al desarrollo de un banco de pruebas remoto con robots móviles y redes de sensores inalámbricos (WSN⁵¹), denominado Integrated TestBed (IT).

Dicho sistema cuenta con una arquitectura abierta y modular que permite una amplia integración de los algoritmos a los que es sometido, ofreciendo sobre dicho contexto una red de sensores distribuida de bajo coste y una serie de robots móviles con los que el usuario puede interaccionar gracias a un completo protocolo de comunicaciones bidireccional.

El Integrated TestBed ha sido desarrollado para la UE⁵², financiado por la Red de Cooperación de Excelencia Europea (CONET) y está disponible actualmente en la Escuela de Ingenieros de Sevilla (España), bajo la supervisión del Grupo de Robótica, Visión y Control (GRVC) del Departamento de Ingeniería de Sistemas y Automática

⁴⁹ CONET: Cooperating Objects NETWORK of excellence [<http://www.cooperating-objects.eu/>]

⁵⁰ AICIA: Asociación de Investigación y Cooperación de Andalucía [<http://www.aicia.es/>]

⁵¹ WSN: Wireless Sensor Network

⁵² UE: Unión Europea

Capítulo 5. Adaptación a Experimentación Real

En concreto, el GRVC se encarga de las tareas de integración y gestión del TestBed, poniendo a disposición del usuario que lo solicite un sistema de control remoto a través de Internet que permite la completa manipulación de los robots y la red de sensores desplegada.

Para ello, el laboratorio utiliza una serie de librerías desarrolladas en C++, soportadas sobre el Sistema Operativo Ubuntu 10.04 y apoyadas por el sistema Player; una herramienta OpenSource diseñada para el control de objetos cooperantes.

5.1.1 Descripción del TestBed

El TestBed ha sido diseñado para permitir una amplia gama de experimentos sobre los que puedan ser integrados tanto los robots móviles como la red de sensores inalámbrica (WSN), compuesta por nodos estáticos y dinámicos.

La distribución básica del banco de pruebas consta de una sala rectangular de $528m^2$ (22m x 24m) con únicamente tres columnas situadas en el centro de la sala, sobre la línea que atraviesa transversalmente la misma.

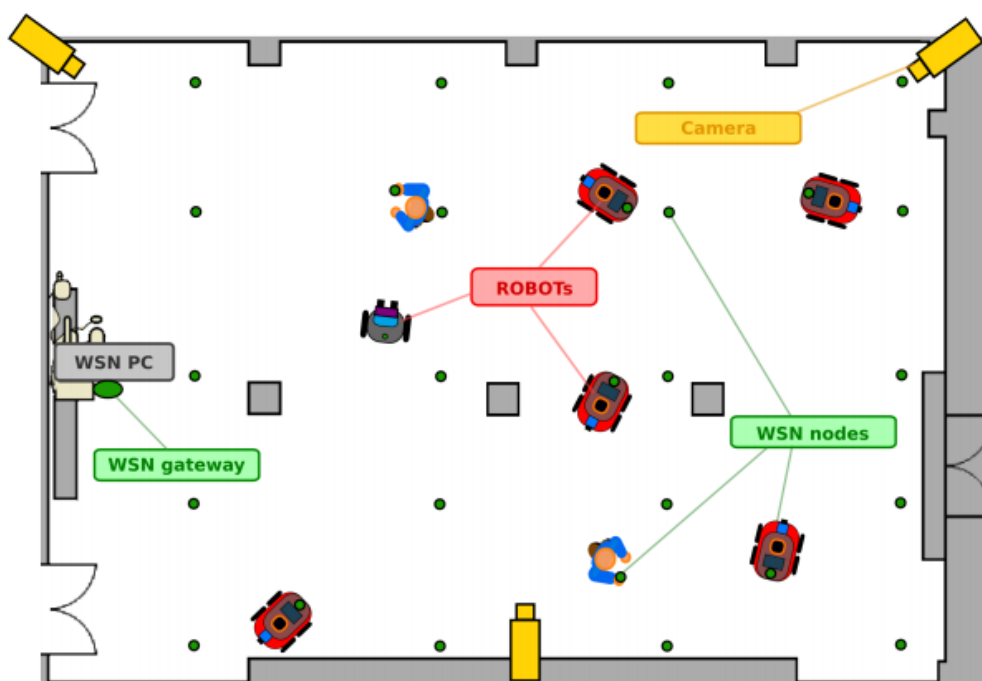


Figura 5.1: Distribución general del Integrated TestBed

La figura 5.1 muestra la distribución general de la sala sobre la que el entorno de experimentación se desarrolla, los robots móviles, y la disposición de los nodos WSN (puntos verdes).

Capítulo 5. Adaptación a Experimentación Real

Como puede observarse, cada robot cuenta con un nodo dinámico acoplado y alimentado por su infraestructura, destinado a los posibles estudios en el campo de la WSN. Así mismo, se admite la inclusión de otros nodos de similares características, portados por personas o por otros tipos de sistemas autónomos.

Todos los nodos estáticos han sido situados a una altura, tal que los robots puedan navegar por el entorno sin necesidad de esquivarlos.

El laboratorio cuenta también con cámaras IP (representadas en amarillo), destinadas a proporcionar a los usuarios remotos una experiencia inmersiva sobre el experimento realizado.

5.1.2 Robots Móviles

El TestBed cuenta con cinco robots tipo Pioneer 3-AT todoterreno de Mobilerobots con configuración Skid-Steer⁵³. Dicho sistema de tracción a efectos prácticos puede ser usado como un sistema diferencial, por tanto éstos agentes pueden ser perfectamente adaptados al código implementado sobre el simulador, sin necesidad de incluir sobre su Reserved Disk zonas adicionales para maniobras.

Adicionalmente, cada robot dispone de un Netbook PC con un procesador Atom Intel con 1024Mb de SDRAM sobre el que corre un servidor de Player. Esto nos brinda la posibilidad de adaptar nuestro código al uso como cliente de Player, empleando de esta forma parte de la potencia computacional de dicho ordenador embarcado.

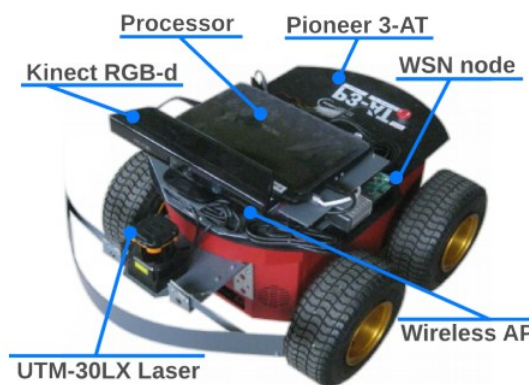


Figura 5.2: Robot Pioneer 3-AT

Como sistema de localización principal los robots cuentan con un Láser de Rango⁵⁴ 2D tipo Hokuyo sobre el que se aplica un Filtro de Partículas⁵⁵ apoyado

⁵³ Skid-Steer: configuración tractora basada en dos ejes perpendiculares a la dirección de avance del robot, dotada de cuatro ruedas motrices dos a dos.

⁵⁴ Láser de Rango: sensor que realiza barridos láser de tiempo de vuelo y que nos permite conocer la distancia hacia los obstáculos adyacentes al mismo.

⁵⁵ Filtro de Partículas: método estadístico de estimación de estados que se usa para el seguimiento de la posición.

Capítulo 5. Adaptación a Experimentación Real

en un mapa preciso del entorno de funcionamiento y que permite una localización del sistema con una alta precisión⁵⁶.

El agente permite adicionalmente el acceso a un sistema de odometría, una Unidad de Medición Inercial (IMU⁵⁷) y a un GPS. Dichos métodos han sido descartados para la localización por la acumulación de errores en el tiempo que suelen caracterizarlos además de por tratarse el laboratorio de una zona carente de señal GPS.

Las comunicaciones del equipo se sustentan principalmente en un Puente Wireless bajo el estándar IEEE 802.11 a/b/g/n capaz de proveer de velocidades máximas teóricas de transmisión de 54Mbps, aunque en la práctica se ha detectado que dicho valor no suele ascender de 36Mbps. La cobertura de dicho puente Wireless contempla todo el laboratorio.

Estas especificaciones son suficientes para tener a cada robot informado sobre la posición de los otros cuatro en todo momento. Por ello, y dado que los obstáculos estáticos son completamente fijos, se obviará la necesidad de un mecanismo capaz de detectar la posición de otros agentes, y se informará previamente al sistema de la situación de los obstáculos, relajando de ésta forma aún más las necesidades de cómputo asociadas.

5.1.3 Arquitectura del Software Implicado

La plataforma software del Integrated TestBed ha sido diseñada para permitir una máxima interacción entre sensores, haciendo uso de una plataforma abierta y heterogénea, capaz de apoyarse en los algoritmos y códigos proporcionados por los usuarios de la misma. Su modularidad, facilidad de uso y extensibilidad ha sido aportada principalmente por una capa de integración de software denominado Player.

Player es un middleware⁵⁸ de código abierto ampliamente utilizado en la red de investigación de Robótica. Se basa en una arquitectura cliente/servidor en la que el servidor interactúa con los elementos de hardware, en nuestro caso los robots, y proporciona a los clientes una interfaz abstracta de fácil uso, capaz de

⁵⁶ La precisión de dicho sistema depende de las mediciones que el mismo puede hacer sobre su entorno. Los robots al enfrentarse suelen crear zonas de sombras para el láser que aumenta el error de la medición del filtro.

⁵⁷ IMU: Inertial Measurement Unit. Unidad diseñada para la adquisición de datos de carácter inercial. Los sensores que suelen caracterizar éstas unidades son los acelerómetros, giróscopos y compases.

⁵⁸ Middleware: Software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y/o sistemas operativos.

Capítulo 5. Adaptación a Experimentación Real

trabajar en múltiples plataformas y coordinarse con diversos lenguajes tales como C, C++, Python, Java y GNU Octave y Matlab entre otros.

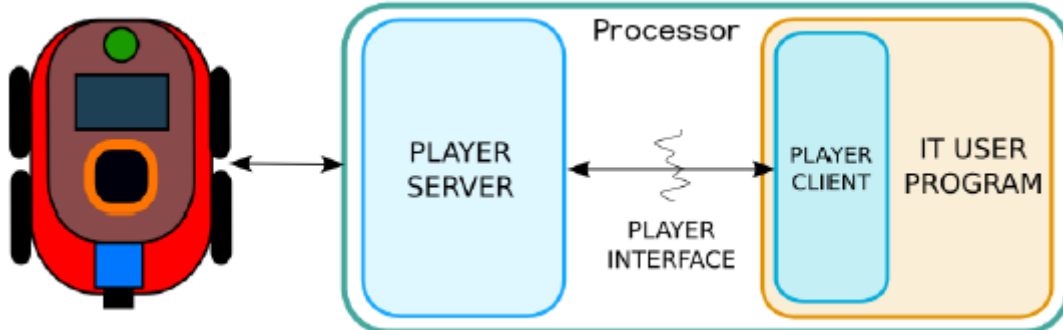


Figura 5.3: Esquema general de Player para el uso en los robots móviles

5.2 Sistema Reactivo aplicado sobre el TestBed

Para la correcta utilización del entorno de experimentación por parte de nuestro sistema se ha optado por realizar una adaptación del código desarrollado al lenguaje C++, debido a que aunque Matlab presenta una interfaz muy potente de representación de datos, su velocidad de ejecución no es comparable con la del código en dicho lenguaje.

Al realizar la adaptación, se ha tenido en todo momento en cuenta la necesidad de compatibilizar el código resultante con la arquitectura Cliente/Servidor que proporciona Player y que rige el laboratorio de experimentación. Para ello, se ha usado la herramienta de simulación virtual del laboratorio proporcionada por el GRVC.

La interfaz ha sido implementada de tal forma que cada sistema embarcado realice una suscripción a la interfaz de control de los motores de su propio robot a la vez que se suscribe a la posición de todos los agentes además de a la de él mismo.

Con ello, y definiendo la posición de las columnas como obstáculos circulares, las paredes como obstáculos circulares de radio semiinfinito, y adaptando el código para el uso de un sistema de medición de la posición global y no local (basado en un sistema de coordenadas no centrado en el propio robot) ha sido posible el uso completo del algoritmo sobre el entorno de experimentación remoto.

Capítulo 5. Adaptación a Experimentación Real

Además de ello, el código ha sido programado para que durante su ejecución, guarde sobre el computador embarcado un archivo .txt, preparado para ser convertido a .m y para ser leído por Matlab.

Dicho archivo contendrá una definición de una matriz de datos Nx12 con la que será posible reconstruir el experimento sobre el computador una vez realizado.

En concreto, cada fila de dicha matriz contendrá la siguiente información:

$$[t_{act} \quad N_{robot} \quad x_{act} \quad y_{act} \quad \varphi_{act} \quad v_{act} \quad x_{dest} \quad y_{des} \quad d_{des} \quad v_{ref} \quad \mathcal{G}_{ref} \quad S]$$

Las variables de dicho vector se corresponden con los siguientes datos:

t_{act}	Tiempo en el que se salva el vector de datos
N_{robot}	Número del agente
x_{act}	Coordenada x del agente
y_{act}	Coordenada y del agente
φ_{act}	Ángulo de orientación del agente
v_{act}	Distancia al objetivo del agente
x_{dest}	Coordenada x objetivo del agente
y_{des}	Coordenada y objetivo del agente
d_{des}	Distancia al objetivo del agente
v_{ref}	Referencia en velocidad del agente
\mathcal{G}_{ref}	Referencia en orientación del agente
S	Estado de la máquina de estados en el instante actual

Tabla 5.1: Relación Símbolo-Dato para el vector de información de un agente en un instante puntual

Capítulo 5. Adaptación a Experimentación Real

Adicionalmente se generará una segunda matriz con la información de la localización de los obstáculos que el agente observa en todo momento. Dicha matriz será de la siguiente forma:

$$[t_{act} \quad N_{obs1} \quad x_{obs1} \quad y_{obs1} \quad \varphi_{obs1} \quad v_{obs1} \quad \dots \quad N_{obsN} \quad x_{obsN} \quad y_{obsN} \quad \varphi_{obsN} \quad v_{obsN}]$$

Las variables de dicho vector se corresponden con los siguientes datos:

t_{act}	Tiempo en el que se salva el vector de datos
N_{obsN}	Número n del obstáculo
x_{obsN}	Coordenada x del obstáculo n
y_{obsN}	Coordenada y del obstáculo n
φ_{obsN}	Orientación del obstáculo n si procede
v_{obsN}	Velocidad del obstáculo n si procede

Tabla 5.2: Relación Símbolo-Dato para el vector de datos de obstáculos registrado por los agentes

Téngase en cuenta que los tiempos del sistema se referencian en segundos reales desde la orden de comienzo del experimento.

Por último, se ha programado un PID para la orientación del Pioneer 3-AT basado en un sistema similar al que utiliza el simulador, pero adaptado y programado para su uso en el experimento real.

Sobre el sistema de capas se utilizarán las mismas suposiciones realizadas sobre el simulador para el Sistema de Movimiento Global⁵⁹.

Los valores aplicados sobre el sistema para realizar los experimentos han sido los siguientes:

K_P^{Speed}	(Player ⁶⁰)	K_P^φ	0.5	R_S	0.4m
K_I^{Speed}	(Player)	K_I^φ	0	R_R	0.7m
K_D^{Speed}	(Player)	K_D^φ	0	$d_{llegada}$	0.25m

Tabla 5.3: Parámetros de configuración del sistema

⁵⁹ Véase Apartado 3.3.1: El Sistema de Movimiento Global

⁶⁰ Control de velocidad proporcionado por la interfaz de Player

Capítulo 5. Adaptación a Experimentación Real

Los obstáculos programados sobre el sistema han sido los siguientes:

Nombre Obstáculo	N_{obsN}	x_{obsN}	y_{obsN}	R_R
Pioneer I	1	-7	-6	0.7
Pioneer II	2	-6	-6	0.7
Pioneer III	3	-5	-6	0.7
Pioneer IV	4	-4	-6	0.7
Pioneer V	5	-3	-6	0.7
Columna 1	6	-2.25	0	0.7
Columna 2	7	2.75	0	0.7
Columna 3	8	7.75	0	0.9
Pared Delantera	9	0	1332	1332.5
Pared Trasera	10	0	-1329	1322.5
Pared Derecha	11	687.15	1.15	675.70
Pared Izquierda	12	-687.15	1.15	675.70

Tabla 5.4: Parámetros de configuración de Obstáculos del sistema

En ella los valores de las posiciones de los robots sólo son válidos en caso de que éstos se encuentren desactivados y en su posición de reposo. La nomenclatura de las paredes se debe a cómo están dispuestas con respecto a la zona de control del Integrated TestBed, siendo así mismo numeradas las columnas de izquierda a derecha.

Capítulo 5. Adaptación a Experimentación Real

5.3 Software de Soporte

Aunque el Sistema de Esquiva Reactivo ha sido transcrito al lenguaje C++ para su completa compatibilidad con el entorno TestBed, el software principal de comunicación con el usuario sigue apoyándose en el entorno Matlab, presentando al mismo una interfaz amigable, también basada en el sistema de intercambio de comandos de dicho programa.

Para su ejecución, es necesario que se llame a la función *EjecutaTestBed.m* desde un Matlab embarcado en un sistema operativo completamente adaptado para su uso en el TestBed⁶¹.

La función permite al usuario seleccionar el número de robots a utilizar (de entre los cinco disponibles) para realizar el experimento y haciendo uso de una interfaz sobre la que representa el laboratorio, permite posicionar con la ayuda del ratón las distribuciones inicial y final del mismo.

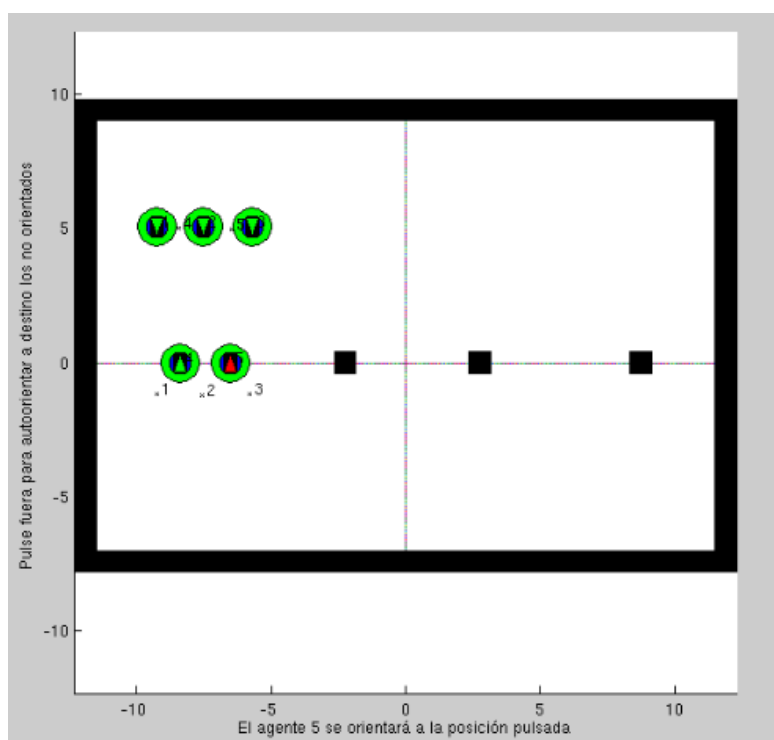


Figura 5.4: Interfaz de Posicionamiento de agentes para la realización de los experimentos

Una vez comprobado que dichas distribuciones son válidas, el sistema se encarga de testear las comunicaciones entre el ordenador, el laboratorio de experimentación y los ordenadores embarcados en los robots a ejecutar.

⁶¹ Típicamente un Sistema Operativo Ubuntu 10.04 con los programas necesarios y un túnel virtual (VPN) habilitado para su uso sobre el Laboratorio.

Capítulo 5. Adaptación a Experimentación Real

Evaluadas favorablemente las comunicaciones el sistema carga el Código de Esquiva Reactivo sobre los agentes, se ejecutan los Servidores de Player en cada uno de ellos y se realizan las suscripciones Cliente/Servidor pertinentes.

El software utiliza el Código de Esquiva Reactivo para emplazar a los agentes de uno en uno sobre la posición que el usuario ha seleccionado como inicial. Esto permite un viaje seguro por parte del agente hacia dicha posición.

Una vez confirmado por el sistema que todos los robots se encuentran situados en su posición inicial y que su velocidad es cero, se procede a dar la orden a los mismos para que comiencen al unísono a buscar el destino, a la vez que a registrar las posiciones propias y ajenas en las matrices de datos pertinentes.

Al concluir el experimento el programa devuelve a los Pioneer 3-AT de forma escalonada a sus respectivas posiciones de reposo, volviendo a hacer uso del algoritmo para evitar las colisiones.

Tras ello se procede a la desconexión de los agentes y a la adquisición de los datos sobre el ordenador central que ha comandado la operación.

Aunque todas las simulaciones realizadas con éste método, han sido efectuadas de forma presencial en el propio TestBed para aliviar la labor de los organizadores, el ordenador ha sido configurado de tal forma que la ejecución se realizase de forma remota. Siguiendo los mismos pasos que cualquier usuario no presencial.

Capítulo 5. Adaptación a Experimentación Real

5.4 Software de Extracción de Resultados

Para la pertinente extracción de resultados, se ha incluido sobre el propio Simulador una función denominada *ProcesaResultados.m* dentro de la subcarpeta “ProcesadorResultados”, capaz de extraer las matrices generadas por los robots durante los experimentos y realizar un análisis completo de las mismas.

Concretamente el sistema extrae de la carpeta “ResultadosEnBruto” los archivos .txt generados y los convierte a archivos .m sobre la carpeta “ResultadosProcesados”.

Interpolando dichos datos se genera una nueva Matriz de Resultados Reales de características similares a la Matriz de Resultados generada por el Simulador⁶² y que se utiliza para crear un video .avi en el que se representa de forma gráfica a los agentes viajando dentro del laboratorio.

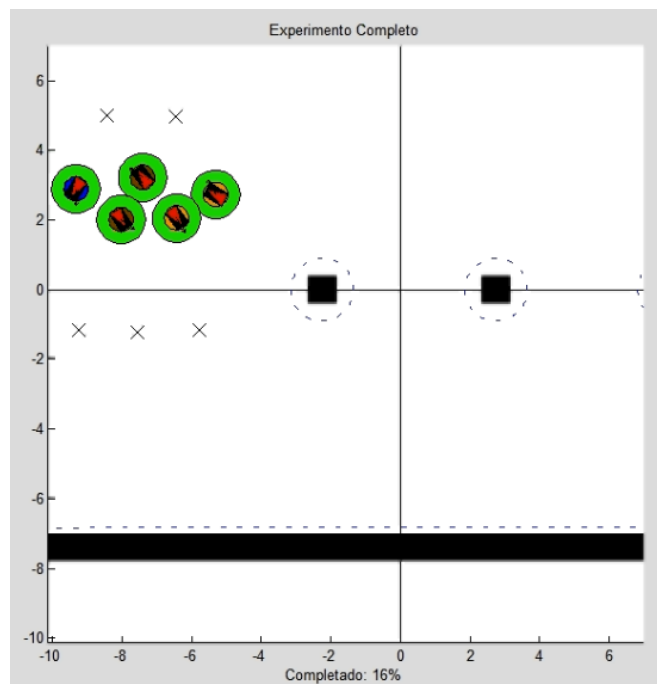


Figura 5.5: Representación de los agentes viajando en el laboratorio

La codificación visual de dicho video es similar a la de los archivos generados por el simulador⁶³, incluyendo adicionalmente unas líneas discontinuas de color azul que representan el final del Reserved Zone de los obstáculos del mapa. Al igual que en el Simulador, la velocidad de ejecución del video es equivalente a la velocidad de ejecución real.

⁶² Véase Apartado 3.2: El Simulador en Matlab

⁶³ Véase Apartado 3.4: Análisis de Resultados en Matlab

Capítulo 5. Adaptación a Experimentación Real

Junto a él, se genera una imagen .jpg denominada “Todoslosagentes” y que contiene tres representaciones. A la izquierda se presenta el recorrido descrito por cada agente durante el experimento, a la derecha el recorrido que los mismos realizarían en caso de no existir otros obstáculos, y en el centro una representación de la distancia de los mismos a su destino con respecto al centro. Por cada agente se ha seleccionado un color y se han marcado los orígenes con una O y los destinos con una X.

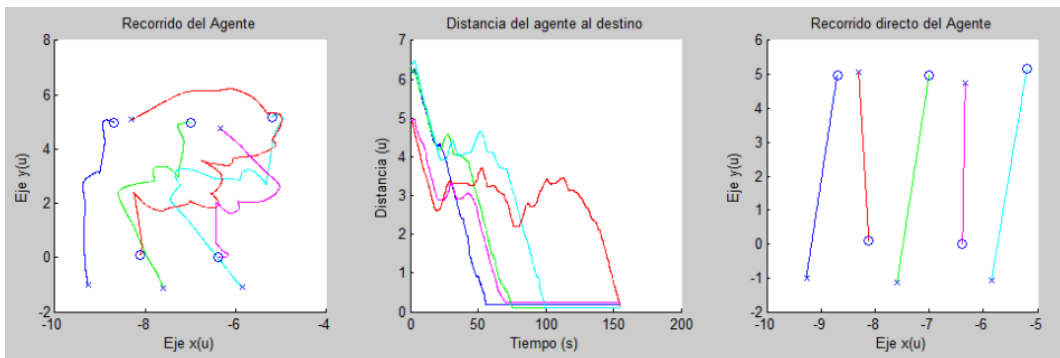


Figura 5.6: Representación del recorrido de los agentes con y sin encuentros y de la distancia al destino

Además de ello, el sistema extrae de la Matriz de Simulación los valores necesarios para rellenar una Matriz de Inicialización, que se pasa al Simulador para que realice una simulación en las mismas condiciones en la que se encontraban los robots sobre el laboratorio.

El resultado de dicha simulación se guarda junto con los resultados analizados en un archivo .m de nombre “ResSimulador”.

Haciendo uso de la Matriz de Resultados de la simulación y de la Matriz de Resultados Reales se representa sobre una imagen .jpg, el recorrido de los agentes reales (como líneas continuas) superpuesto con el recorrido de los agentes virtuales (como líneas discontinuas).

Capítulo 5. Adaptación a Experimentación Real

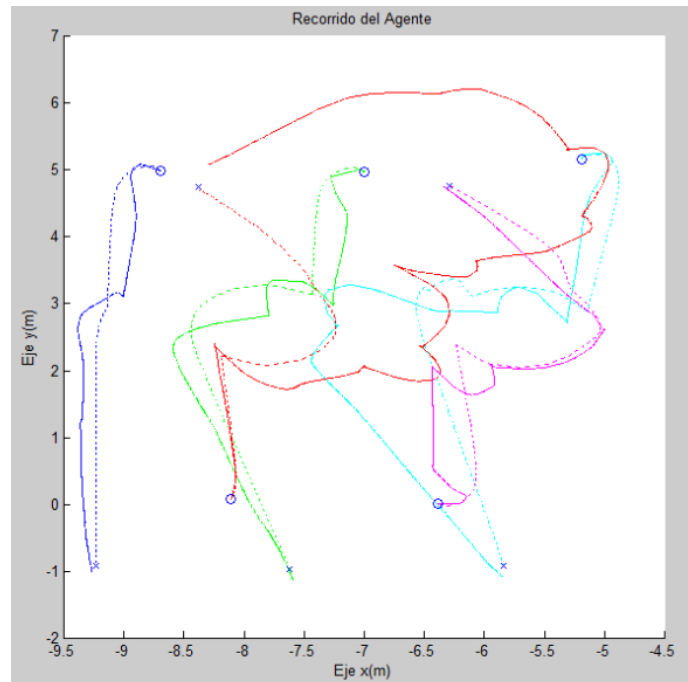


Figura 5.7: Representación del recorrido de los agentes reales y virtuales

Tras ello se realiza un análisis individual de cada agente en la que se crean dos archivos .jpg. El primer tipo representanta individualmente las trayectorias contrapuestas de los agentes reales y virtuales haciendo uso de líneas continuas para los primeros y discontinuas para los segundos, y se guarda bajo el nombre "RealVsVirtualAgenteN" donde N es el número del robot representado.

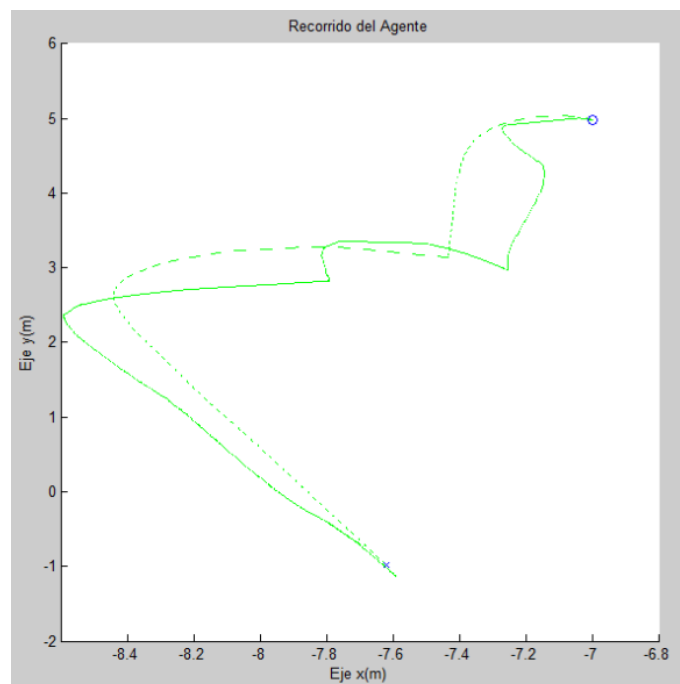


Figura 5.8: Representación del recorrido de un agente real contra uno virtual

Capítulo 5. Adaptación a Experimentación Real

El segundo tipo genera una imagen .jpg, con toda la información relativa a la navegación del mismo, incluyendo gráficamente los siguientes valores:

- Recorrido del robot sobre el laboratorio.
- Referencia en orientación con respecto al tiempo.
- Orientación adquirida con respecto al tiempo.
- Velocidad referencia del robot con respecto al tiempo.
- Velocidad filtrada del robot con respecto al tiempo⁶⁴.
- Distancia del agente al objetivo con respecto del tiempo.

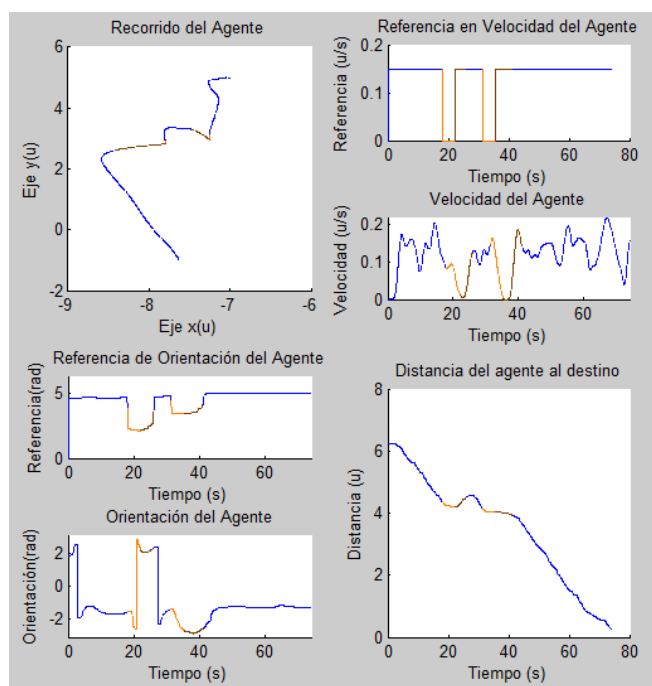


Figura 5.9: Representación .png de los datos del recorrido de un agente

Por último se genera un vídeo por cada agente denominado “PioneerNPrincipal” en el que se representa la visión de los obstáculos que el mismo observa durante su recorrido⁶⁵; éstos son representados por una línea discontinua azul.

El agente sigue el código de colores característico del simulador y deja tras de si durante el viaje la representación de la trayectoria bajo la misma codificación visual.

⁶⁴ La velocidad del robot ha sido filtrada debido a que el sistema calcula la velocidad de la diferencia espacio-temporal de dos muestras consecutivas, heredando de ésta forma el ruido sobre la posición del sistema. El filtro aplicado ha sido Paso Baja (LP).

⁶⁵ El sistema hace uso de la Matriz de Obstáculos salvada durante el experimento.

Capítulo 5. Adaptación a Experimentación Real

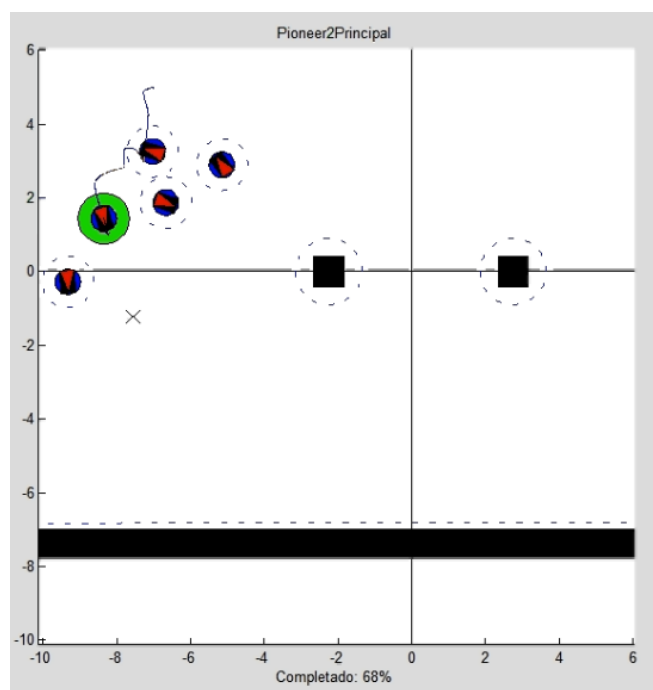


Figura 5.10: Representación de un agente y sus obstáculos

Capítulo 5. Adaptación a Experimentación Real
