UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN SISTEMAS

COORDINACIÓN DE ROBOTS MÓVILES EN TIEMPO REAL

TESIS
QUE PARA OPTAR POR EL GRADO DE
MAESTRO EN CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

PRESENTA:
GERARDO ARIEL CASTILLO GARCIA

TUTOR O TUTORES PRINCIPALES
DR. HÉCTOR BENÍTEZ PÉREZ
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN SISTEMAS

CIUDAD UNIVERSITARIA, CD. MX., ENERO 2019

Agradecimientos

A la Universidad Nacional Autónoma de México, porque todo lo que me ha brindado, gracias a esta institución me ha ayudado a formar mi camino en la vida.

Al Consejo Nacional de Ciencia y Tecnología por el apoyo entregado para la realización de mis estudios de maestría mediante la beca nacional para estudios de posgrado (agosto 2016 a julio 2018).

A mi familia, a Graciela, Greko y Sasha que siempre estuvieron conmigo en todo este proceso, su cariño y afecto. A mis padres y hermana, por su soporte y consejos.

Al Dr. Héctor Benítez Pérez, por su tiempo, dedicación y la confianza depositada en mi para desarrollar este trabajo.

Agradezco a los miembros del jurado por su tiempo e interés para revisar mi tesis, así como sus valiosas observaciones

Contenido

Símbolos Principales y Nomenclatura

FDIE-C	Filtro disperso de información extendida por consenso
N_{obs}	Número de observaciones realizadas por el scanner
M	Número de puntos de referencia en el mapa global
$\boldsymbol{m}({}^{I}k)$	Vector de puntos de referencia encontrados respecto al sistema global en el instante ${}^{I}k$. Son los puntos que son vistos por el sensor en ${}^{I}k$
$\boldsymbol{m}^{+}({}^{I}k)$	Vector de puntos de referencia encontrados respecto al sistema global en el instante ${}^{I}k$.
$\boldsymbol{m}^{\boldsymbol{0}}({}^{I}k)$	Vector de puntos de referencia activos a pasar a pasivos respecto al sistema global en el instante ${}^{I}k$. Son puntos que en un tiempo ${}^{I}k-1$ eran activos, y en ${}^{I}k$ son pasivos.
$\boldsymbol{m}^{-}({}^{I}k)$	Vector de puntos de referencia pasivos respecto al sistema global en el instante ${}^{I}k$. Son los puntos que vistos por el sensor antes de ${}^{I}k-1$
G	Red de conexión de sistema multiagente
I, J	Nodos en red de conexión
$N(I)$	Vecindario del nodo I
t	Tiempo continuo global
k	Tiempo discreto global (estampa de tiempo)
$\{I\}, \{J\}$	Sistema de referencia del sistema (robot) $\{I\}$, $\{J\}$
${}^{I}t_k$, ${}^{I}k$	Discretización del tiempo en el instante k, respecto al robot $\{I\}$
${}^{I}X_O$, ${}^{I}Y_O$	Posición del punto O en los ejes X y Y,respecto al robot $\{I\}$.
O, P, Q	Puntos de referencia.
${}^{I}_{J}\theta, {}^{I}_{J}\varphi$	Orientación θ y φ del sistema $\{J\}$, respecto al robot$\{I\}$.
${}^{I}\alpha$, ${}^{I}\beta$, ${}^{I}\gamma$	Ángulos α, β, γ medidos respecto al eje X del robot $\{I\}$.
${}^{I}\tau_i(r, C, D, \delta)$	i-ésima tarea, respecto al sistema $\{I\}$
${}^{I}\mathrm{T} = \{ {}^{I}\tau_1, {}^{I}\tau_2 \dots {}^{I}\tau_n\}$	Conjunto de tareas, respecto al sistema $\{I\}$
$r({}^{I}\tau_i)$	Tiempo de liberación de la i-ésima tarea, respecto al sistema $\{I\}$
$s({}^{I}\tau_i)$	Tiempo de inicio de la i-ésima tarea, respecto al sistema $\{I\}$

$f(\ {}^{I}\tau_i)$	Tiempo de finalización de la i-ésima tarea, del nodo $\{I\}$
$R(\ {}^{I}\tau_i)$	Tiempo de respuesta de la i-ésima tarea, del nodo $\{I\}$
$C(\ {}^{I}\tau_i)$	Tiempo de completitud de la i-ésima tarea, del nodo $\{I\}$
$D(\ {}^{I}\tau_i)$	Fin de plazo de la i-ésima tarea, respecto del nodo $\{I\}$
$\delta_C(\ {}^{I}\tau_i)$	Variabilidad de tiempo de completitud de la i-ésima tarea, del nodo $\{I\}$
$\delta_R(\ {}^{I}\tau_i)$	Variabilidad de tiempo de respuesta de la i-ésima tarea, del nodo $\{I\}$
${}^{I}_{J}\mu_i(r, L, C, \delta)$	i-ésimo mensaje, enviado del nodo $\{J\}$ al nodo $\{I\}$
$r({}^{I}_{J}\mu_i)$	Tiempo de liberación del i-ésimo mensaje, enviado del nodo $\{J\}$ al nodo $\{I\}$
$s({}^{I}_{J}\mu_i)$	Tiempo de inicio de transmisión del i-ésimo mensaje, enviado del nodo $\{J\}$ al nodo $\{I\}$
$f({}^{I}_{J}\mu_i)$	Tiempo de finalización de transmisión del i-ésimo mensaje, enviado del nodo $\{J\}$ al nodo $\{I\}$
$L({}^{I}_{J}\mu_i)$	Latencia del i-ésimo mensaje, enviado del nodo $\{J\}$ al nodo $\{I\}$
$\delta_L({}^{I}_{J}\mu_i)$	Variabilidad de latencia del i-ésimo mensaje, enviado del nodo $\{J\}$ al nodo $\{I\}$
${}^{I}v$	Rapidez tangencial del sistema $\{I\}$.
${}^{I}\omega$	Rapidez rotacional del sistema $\{I\}$.
${}^{I}v_R$	Rapidez tangencial de la rueda derecha del robot $\{I\}$.
${}^{I}v_L$	Rapidez tangencial de la rueda izquierda del robot $\{I\}$.
${}^{I}l$	Distancia entre la rueda izquierda y derecha del robot $\{I\}$.
${}^{I}\Delta d(\ {}^{I}k)$	Distancia recorrida por el robot $\{I\}$ en el instante ${}^{I}k$.
${}^{I}\Delta\theta(\ {}^{I}k)$	Giro realizado por robot $\{I\}$ en el instante ${}^{I}k$.
${}^{I}\Delta\theta_R(\ {}^{I}k)$	Giro realizado por la rueda derecha del robot $\{I\}$ en el instante ${}^{I}k$.
${}^{I}\Delta\theta_L(\ {}^{I}k)$	Giro realizado por la rueda izquierda del robot $\{I\}$ en el instante ${}^{I}k$.
${}^{I}\delta\theta_R(\ {}^{I}k)$	Error de medición en el giro realizado por la rueda derecha del robot $\{I\}$ en el instante ${}^{I}k$.
${}^{I}\delta\theta_L(\ {}^{I}k)$	Error de medición en el giro realizado por la rueda izquierda del robot $\{I\}$ en el instante ${}^{I}k$.
$\boldsymbol{f}$	Vector de función de transición.
$\boldsymbol{r}_I(\ {}^{I}k)$	Vector de pose del robot $\{I\}$ en el instante ${}^{I}k$.
$\boldsymbol{x}_I(\ {}^{I}k)$	Vector de estado del robot $\{I\}$ en el instante ${}^{I}k$.
${}^{I}\boldsymbol{u}(\ {}^{I}k)$	Vector de odometría respecto al robot $\{I\}$ en el instante ${}^{I}k$.

$^{I}\delta\boldsymbol{u}(\ ^{I}k)$	Vector de error de odometría respecto al robot $\{I\}$ en el instante ^{I}k.
$\hat{\boldsymbol{x}}_I(\ ^{I}k\mid\ ^{I}k)$	Estimación de vector de estado del sistema $\{I\}$ respecto al sistema global.
$\hat{\boldsymbol{x}}_I(\ ^{I}k\mid\ ^{I}k-1)$	Predicción de vector de estado del sistema $\{I\}$ respecto al sistema global.
$\hat{\boldsymbol{y}}_I(\ ^{I}k\mid\ ^{I}k)$	Estimación de vector de información del sistema $\{I\}$ respecto al sistema global.
$\boldsymbol{I}$	Matriz de identidad.
$\boldsymbol{P}_I(\ ^{I}k\mid\ ^{I}k)$	Matriz de covarianza de estado del sistema $\{I\}$ respecto al sistema global.
$\boldsymbol{Y}_I(\ ^{I}k\mid\ ^{I}k)=\{\boldsymbol{P}_I(\ ^{I}k\mid\ ^{I}k)\}^{-1}$	Matriz de información del sistema $\{I\}$ respecto al sistema global.
$\boldsymbol{x}(\ ^{I}k)_{,\boldsymbol{x}(\ ^{I}k-1)}=\dfrac{\partial\boldsymbol{x}(\ ^{I}k)}{\partial\boldsymbol{x}(\ ^{I}k-1)}$	Matriz Jacobiana del estado $\boldsymbol{x}(\ ^{I}k)$ respecto el vector de estado anterior $\boldsymbol{x}(\ ^{I}k-1)$
$\boldsymbol{r}(\ ^{I}k)_{,\boldsymbol{r}(\ ^{I}k-1)}=\dfrac{\partial\boldsymbol{r}(\ ^{I}k)}{\partial\boldsymbol{r}(\ ^{I}k-1)}$	Matriz Jacobiana del vector de la función de transición $\boldsymbol{r}(\ ^{I}k)$ respecto el vector de la pose anterior $\boldsymbol{r}(\ ^{I}k-1)$
$\boldsymbol{m}(\ ^{I}k)_{,\boldsymbol{m}(\ ^{I}k-1)}=\dfrac{\partial\boldsymbol{m}(\ ^{I}k)}{\partial\boldsymbol{m}(\ ^{I}k-1)}$	Matriz Jacobiana del vector de mapa actual $\boldsymbol{m}(\ ^{I}k)$ respecto el vector del mapa anterior $\boldsymbol{m}(\ ^{I}k-1)$
$\boldsymbol{r}(\ ^{I}k)_{,\boldsymbol{u}(\ ^{I}k)}=\dfrac{\partial\boldsymbol{r}(\ ^{I}k)}{\partial\boldsymbol{u}(\ ^{I}k)}$	Matriz Jacobiana del vector de pose actual $\boldsymbol{r}(\ ^{I}k)$ respecto el vector de odometría $\boldsymbol{u}(\ ^{I}k)$**.**
$^{I}\delta\boldsymbol{U}(\ ^{I}k)=E\left[\left(\ ^{I}\delta\boldsymbol{u}(\ ^{I}k)\right)^2\right]$	Matriz de covarianza de error de odometría del robot $\{I\}$ en el instante ^{I}k.
$^{\mathrm{I}}\rho_P(\ ^{I}k)$	Distancia radial del punto de referencia P, respecto respecto al sistema $\{I\}$, en el instante ^{I}k.
$^{\mathrm{I}}\alpha_P(\ ^{I}k)$	Ángulo del punto de referencia P, observado por el robot $\{I\}$ en el instante ^{I}k.
$\boldsymbol{m}_P(\ ^{I}k)=\left[X_P(\ ^{I}k),Y_P(\ ^{I}k)\right]$	Posición del punto P en los ejes X y Y observado por el robot $\{I\}$ en el instante ^{I}k.
$^{\boldsymbol{I}}\boldsymbol{z}_P(\ ^{I}k)$	Vector de observación del punto de referencia P observado por el robot $\{I\}$ en el instante ^{I}k.
$^{\boldsymbol{I}}\delta\boldsymbol{z}_{\boldsymbol{P}}(\ ^{I}k)$	Vector de error de observación del punto de referencia P observado por el robot $\{I\}$ en el instante ^{I}k.
$^{\boldsymbol{I}}\boldsymbol{z}(\ ^{I}k)_{,\boldsymbol{x}(\ ^{I}k-1)}=\dfrac{^{I}\partial\boldsymbol{z}(\ ^{I}k)}{\partial\boldsymbol{x}(\ ^{I}k-1)}$	Matriz Jacobiana del vector de observación $^{\boldsymbol{I}}\boldsymbol{z}(\ ^{I}k)$ respecto el vector de estado $\boldsymbol{x}(\ ^{I}k-1)$**.**
$^{I}\delta\boldsymbol{Z}(\ ^{I}k)=E\left[\left(\ ^{I}\delta\boldsymbol{z}_m(\ ^{I}k)\right)^2\right]$	Matriz de covarianza de error de observación de punto de referencia P del robot $\{I\}$ en el instante ^{I}k.
$^{I}\phi$	Función de navegación local del sistema $\{I\}$
$^{I}\kappa(\ ^{I}X_O,\ ^{I}Y_O)$	Curvatura en el punto O, de la función de flujo del sistema $\{I\}$

Capítulo 1. Introducción

1.1 Motivación

Uno de los problemas más desafiantes en la navegación de un robot es la localización autónoma y la creación de un mapa de manera confiable. En muchas aplicaciones robóticas, como búsqueda y rescate, vigilancia y exploración planetaria, se requiere una localización precisa en entornos desconocidos. Cuando los robots operan en áreas no asignadas o denegadas por GPS, lograr una localización precisa requiere crear y mantener un mapa de su entorno. El problema de construir un mapa mientras se explora un territorio desconocido se conoce comúnmente como localización y mapeo simultáneo (SLAM) [1]. Para aumentar la precisión y la eficacia al mapear áreas extensas, se puede utilizar dos o más robots participen en esta tarea, los robots cooperan para construir un mapa único y conjunto del área que exploran [2]. Este proceso se conoce como Multi-robot SLAM o SLAM cooperativo (C-SLAM). Sin embargo, esta estrategia presenta un compromiso, al mejorar la eficacia, se aumenta la complejidad de C-SLAM.

Las técnicas SLAM utilizadas para un solo robot se desarrollaron desde mediados de los 80s, pero fue hasta 2004 en el primer DARPA Grand Challenge que se le dio más atención [3]. Por otro lado, la extensión de estas técnicas a múltiples robots, con el fin de realizar tareas cooperativas SLAM en entornos desconocidos o dinámicos, sigue siendo un gran desafío y tema de investigación [4].

En los sistemas multi-robots, la precisión con la que los robots pueden modelar su entorno tiene un profundo impacto en el rendimiento individual y del equipo. Se espera que un equipo de robots construya una representación consistente del espacio, de una manera mucho más rápida que un solo robot.

En general se contemplan tres etapas para realizar SLAM en un solo robot. La primera es la predicción del estado. El estado se compone por la pose (posición y orientación del robot) y los puntos de referencia del ambiente, en la predicción se usa el modelo de transición del estado, es decir, se utiliza la información previa. La segunda etapa es la corrección del estado, aquí se utilizan las observaciones realizadas, para corregir la predicción, se hace uso de un modelo de

observación. Por último, se realiza la asociación de los puntos de referencia, respecto a las observaciones.

En el SLAM multi-robot el mapa se puede realizar de dos maneras distintas. Mediante un mapa global, el cual debe ser actualizado por cada miembro del equipo, todos tienen el mismo mapa. La otra alternativa es con mapas parciales, cada robot crea su propio mapa y en algún momento de la exploración, fusionan sus mapas en uno global.

Esta última metodología es la utilizada en este trabajo investigación. La unión o fusión de los mapas en dos fases. Primero se alinean los mapas, para ello en cada robot se calcula la transformación de coordenadas de cada uno de los marcos de referencia de los robots, a un marco global para todo el equipo. Posterior a que el mapa de cada robot se haya transformado, la segunda fase es la estimación de puntos de referencia coincidentes, los cuales se deben fusionar para crear el mapa global.

1.2 Objetivos

El objetivo principal de este trabajo es la propuesta de un algoritmo para la fusión de mapas locales de múltiples robots móviles en uno global. Cada robot tiene una versión parcial (local) del mapa explorado, conforme avanzan comparten esta información con sus pares, y con ello generan un mapa global con mayores elementos, que los locales, este algoritmo es llamado filtro disperso de información extendida por consenso (o FDIE-C por sus siglas). Este algoritmo consta de dos etapas, en la primera se usa el filtro disperso de información extendida, este es un filtro de Kalman extendido, se utiliza para estimar la posición de cada robot y los puntos de referencia del mapa observado respecto a los puntos de referencia de todo el mapa [5] [6]; las estimaciones se representan mediante el vector de información y la matriz de información, con esta representación las nuevas observaciones de los robots se pueden fusionar de manera aditiva [7]. La segunda etapa del algoritmo es el filtro de consenso, es un algoritmo distribuido con el cual los robots llegan a un valor propuesto en conjunto, en este caso un mapa global, mediante la fusión de los mapas locales. La fusión se logra mediante la adición de la estimación de los puntos de referencia, en su representación de vector de información de cada robot; este algoritmo cuenta con propiedades de robustez al presentarse retrasos en la red o perdida de paquetes.

1.3 Metodología

Para lograr de este objetivo se proponen dos etapas. En la primera, se hará una revisión de los algoritmos utilizados en SLAM en sistemas tanto de un solo robot, como en sistemas multirobot, así como de la fusión de datos en redes de sensores móviles. En la segunda etapa se determinarán las tareas a realizar por los robots de manera individual y en lo colectivo conforme exploran el espacio, así como determinar sus restricciones en el tiempo, prioridad y periodicidad.

1.4 Metas

Para este trabajo se plantean cuatro metas, que se detallan a continuación. La primera meta es utilizar un algoritmo de tiempo lineal para la estimación del mapa de cada robot y cuya representación permita la fusión de los mapas de forma aditiva. La segunda es utilizar el filtro de consenso para la fusión de mapas locales en uno global, este contará con las propiedades de libre de espera y tolerancia a fallos, es decir, en cada robot la fusión del mapa global llega a un valor consensado estable, a pesar de tener una red de comunicaciones con pérdida de paquetes y retrasos. La tercera meta es definir algoritmos con complejidad polinomial para las tareas individuales de los robots como son planificación de movimiento y la extracción de los puntos de referencia del mapa de cada robot, con ello, los robots pueden explorar el ambiente de manera segura, es decir, evitando colisiones, con esto se busca cumplir con las restricciones de tiempo real. La cuarta y última meta es definir un programa para simulación de robots móviles comunicados mediante una red con pérdida de paquetes y retrasos para la experimentación de los algoritmos mencionados

1.5 Estructura de la tesis

El presente trabajo se divide en cuatro capítulos. En el primero y segundo se realiza una introducción a los términos y trabajos relacionados al tema de investigación. En el tercer capítulo, se describe de manera detallada el algoritmo propuesto para la fusión de mapas, mediante el FDIE-C. En el cuarto capítulo se detallan la implementación y los resultados, y como se desarrolló la experimentación mediante la simulación de robots móviles, así como su comparación con los trabajos relacionados. Por último, en las conclusiones se realiza un análisis de ventajas y desventajas respecto al trabajo realizado, así como posibles mejoras y trabajo a futuro.

Capítulo 2. Antecedentes

2.1 Sistemas multi-agente

Un sistema multi-agente es en esencia un sistema estructurado por un conjunto de agentes autónomos, que son capaces de adaptar flexiblemente su comportamiento de acuerdo con las condiciones cambiantes en su operación y ambiente [8, p. 2]. Cada agente tiene información o capacidades incompletas para solucionar del problema, por tanto, un punto de vista limitado. En este tipo de sistemas no existe un control global o centralizado. El cómputo en los agentes es asíncrono, es decir, al momento en que perciben información del ambiente, de manera concurrente se procesa información previa para la toma de acciones locales o se comunica el resultado del procesamiento con otros agentes. La comunicación puede ser de manera indirecta o directa, en la comunicación indirecta se modifica el entorno para transmitir la información, por ejemplo, algún agente puede cambiar la posición de un objeto del entorno, o la propia, y esto puede tener un significado para otro agente; en la comunicación directa se utiliza una red de comunicación.

La coordinación entre los agentes puede ser de cooperativa o competitiva. La comunicación es imperativa para la coordinación de las actividades, para que cada uno de los subsistemas (o agentes) pueda alcanzar una meta en común o mejorar el desempeño global del sistema [9, p. 265]. Los objetivos de la cooperación en los sistemas multiagente son el consenso o la sincronización [10, pp. 24,25] con estas operaciones se puede llevar a los agentes a un estado en común. Por ejemplo, en el control de formación multi-agente, los agentes mantienen una relación geométrica, como un sólido rígido, al desplazarse en un medio; en la estimación multi-agente, las observaciones de los agentes se asocian o fusionan para obtener una estimación con mayores características; en la asignación de tareas multi-agente, se busca asignar tareas o recursos, realizar la planificación del sistema de manera distribuida. Las aplicaciones de los multi-agentes competitivos se relacionan con la teoría de juegos [11, p. 161], cada uno de los agentes busca su propio beneficio, y con ello obtener la mejor decisión o el agente más apto en un cierto ambiente.

Sistemas multi-robot

El uso de sistemas multi-robot generalmente tiene mayores ventajas sobre los sistemas de un solo robot. Se ha demostrado que múltiples robots pueden localizarse de manera más rápida y con mayor precisión cuando intercambian entre ellos información de sus posiciones, además al utilizar múltiples robots agrega redundancia, y con ello se hace más tolerante a fallas el sistema [12, p. 3]. Otra de las ventajas es el reducir costos al utilizar más robots baratos en lugar de un robot más caro, ya que se pueden utilizar sensores y actuadores más económicos.

Un sistema multi-robot es un conjunto de robots autónomos o semi-autónomos que interactúan entre sí, con un objetivo en común, intención o estrategia; el sistema tiene capacidades tanto distribuidas y/o compartidas de cómputo, percepción, actuación, fuentes de energía y comunicación [13, p. 5].

Los sistemas multi-robot tienen distintos tipos de propiedades [13, pp. 16-24] como son la comunicación, autonomía, escalabilidad, conocimiento colectivo, y cooperación. Las comunicaciones al igual que los sistemas multi-agente pueden ser directas o indirectas. La autonomía determina la capacidad de cada robot de actuar independientemente, el cual va desde agentes completamente autónomos hasta la telerobótica. La escalabilidad es la flexibilidad de adición de nuevos nodos o agentes a la red de comunicación. El conocimiento colectivo depende del nivel de abstracción de la representación del conocimiento, puede ir desde un conocimiento global hasta uno individual donde no se comparte nada, es partir de las interacciones donde surge conocimiento y comportamiento colectivo. La cooperación depende de arquitectura de comunicaciones, por ejemplo, la cooperación puede ser total si la red está totalmente conectada, o tener interacciones únicamente locales como en una red ad-hoc.

2.2 Fusión de datos

La fusión de datos o información en sistemas multi-agente se refiere a la asociación, correlación, estimación y combinación de datos o información; las fuentes de datos o información pueden tener distintos niveles de abstracción, estos niveles comprenden señales, patrones, información y decisiones [14, p. 3]. La fuente de información puede ser desde una base de datos hasta una red de sensores, este trabajo se centra en la fusión de datos mediante los sensores de los robots móviles. La fusión busca mejorar la representación, precisión y certeza contemplando la completitud de los datos. La representación es función del nivel de abstracción de los datos o

información y la técnica de alineación de la información; tanto la precisión como certeza dependerán del método utilizado en la fusión; la topología de las fuentes determina la completitud.

La representación en la fusión de modelos se categoriza de acuerdo con distintos aspectos, estos pueden ser respecto a la relación entre las fuentes, los niveles de abstracción de las entradas y salidas de los algoritmos [15, pp. 4-6]. La relación de las fuentes contempla tres casos: complementarias, cooperativas o competitivas. Cuando las fuentes son complementarias se puede obtener una imagen más amplia del modelo observado, por ejemplo, se puede tener el mismo tipo de sensor en distintas locaciones; en la fusión por fuentes cooperativas, estas pueden ser de distintos tipos de sensores, pero se estima la misma variable o estado, como es el caso de la combinación de sensor de infrarrojo y un sensor para visión para estimar de manera conjunta la posición de algún objetivo; por otro lado, en la fusión de fuentes competitivas se utiliza la fuente con la información más confiable, con el objetivo de reducir la incertidumbre y error inducido por las demás. Como se mencionó anteriormente, las fuentes de información o datos tienen distintos niveles de abstracción, desde el nivel más bajo que es la señal del sensor, el siguiente nivel son los patrones o características, y el último nivel son las decisiones [15, p. 3]. Los algoritmos de fusión pueden utilizar distintos niveles de abstracción en las entradas y las salidas, generalmente las entradas tienen un nivel de abstracción menor o igual al de su respectiva salida, por ejemplo, se puede utilizar como entradas las señales de sensores y fusionarlas para obtener patrones o decisiones. La alineación de la información es fundamental para la asociación en la fusión, las técnicas de alineación pueden ser tanto espaciales, temporales, semánticas, como estadísticas [15, p. 52].

Los métodos utilizados para la fusión de datos están basados en técnicas de estimación de parámetros o estado, estas técnicas tienen en general tres procedimientos básicos, los cuales son el modelo del sistema o ambiente, el criterio de optimización y el algoritmo [16] [17, pp. 5,6]. El ambiente o sistema cuyos parámetros o estados se desea estimar tiene un modelo que puede ser determinístico, probabilístico o heurístico [18, p. 23]. Para los modelos probabilísticos o determinísticos existen distintos criterios de optimización como son mínimos cuadrados, error medio cuadrático o máxima verosimilitud. Para alcanzar los criterios de optimización se pueden utilizar algoritmos derivativos o no derivativos, directos o indirectos. El algoritmo de fusión de los datos puede ser secuencial o de lote.

2.3 Sistemas distribuidos en tiempo real

Los sistemas distribuidos en tiempo real (DRTS por siglas en inglés) son sistemas que interactúan en un ambiente físico, entre ellos, e inclusive con un usuario u operador [19, 20]. Al ser un sistema de cómputo distribuido, los componentes del sistema deben tener capacidades de coordinación, tanto en el cómputo como en las comunicaciones. La necesidad del tiempo real se debe a que los componentes de cómputo se encuentran embebidos en un ambiente físico y el tiempo de respuesta depende del comportamiento o dinámica del ambiente. Estos sistemas se utilizan para monitorear o controlar dicho ambiente por lo general tiene un comportamiento continuo en el tiempo.

El diseño de un DRTS es una actividad multidisciplinaria ya que requiere conocer los requerimientos en hardware (procesador, telecomunicaciones, sensores, actuadores, etc.) y software (aplicaciones, middleware, sistemas operativos, protocolos de red), así como el modelo matemático del sistema a controlar o monitorear.

Al hablar DRTS es necesario definir los parámetros temporales y su variabilidad, así como, las acciones necesarias para reducir el impacto de estos en el rendimiento del sistema, es decir, las políticas de planificación y los componentes de software que permiten realizarlas. Dichos parámetros, acciones y componentes se pueden ver desde la perspectiva del cómputo y las comunicaciones.

Cómputo en tiempo real

En un sistema de cómputo en tiempo real, el correcto comportamiento del sistema no solo depende de la respuesta correcta del mismo, también depende del tiempo en que las respuestas fueron obtenidas [21, 20]. Como se ha mencionado, este tipo de sistemas son utilizados en aplicaciones de control industrial, control de sistemas de vuelo, robótica, telecomunicaciones y sistemas espaciales; en este tipo de aplicaciones la respuesta debe tener un límite temporal, ya que una respuesta tardía puede llevar a una falla crítica del sistema.

La unidad fundamental en el cómputo tiempo real es la tarea τ; en un sistema de cómputo pueden estar ejecutándose de manera concurrente n tareas, $\mathrm{T} = \{\tau_1, \tau_2 \dots \tau_n\}$. Una tarea es una secuencia de instrucciones. Las tareas pueden ser clasificadas de acuerdo con su periodicidad, las cuales pueden ser periódicas, esporádicas o aperiódicas. El modelo de tiempo de una tarea o conjunto de ellas utilizado en este escrito es el mismo que en [22, 23]. El modelo de tiempo de la

tareas ${}^{I}\tau_i(r, C, D, \delta_C)$ esta descrito por los parámetros temporales. Para la i-ésima tarea del I-ésimo nodo ${}^{I}\tau_i$ sus parámetros temporales son : el tiempo de liberación $r\left({}^{I}\tau_i\right)$, inicio $s\left({}^{I}\tau_i\right)$, finalización $f\left({}^{I}\tau_i\right)$, completitud $C\left({}^{I}\tau_i\right)$ y fin de plazo $D\left({}^{I}\tau_i\right)$. El tiempo de completitud es susceptible a la variabilidad $\delta_C\left({}^{I}\tau_i\right)$, es decir, no siempre se tiene los mismos tiempos, en cada tarea periódica. En el diseño del DRTS es necesario definir un límite en la variabilidad, para asegurar la respuesta adecuada del sistema. La variabilidad es en muchas ocasiones consecuencia de los tiempos de acceso a memoria.

Las políticas de planificación permiten a los sistemas operativos en tiempo real (RTOS por sus siglas en inglés) manejar los recursos del sistema, por ejemplo, asignar tiempo de procesador y memoria a alguna tarea o proceso. Estas permiten que el tiempo de respuesta sea menor que el fin de plazo de la tarea. En general, la planificación de tareas depende de su tiempo de completitud, tiempo de inicio de la tarea, periodicidad y apropiación.

Las políticas de planificación pueden tener dos categorías, la primera es fuera de línea, es decir, se define la asignación de tareas al procesador en tiempo de diseño, la segunda es en línea, en este caso, la asignación sucede en tiempo de ejecución. Cuando la política es en línea la planificación puede ser estática o dinámica, la planificación estática depende de parámetros invariantes, como es el periodo; mientras que la planificación dinámica depende de parámetros variantes en el tiempo, como puede ser la laxitud. También, tanto en la planificación dinámica como en la estática, se puede asignar a las tareas una prioridad, con ello se pueden apropiar del tiempo de procesador, es decir, la tarea que se encontraba en ejecución pasa a un estado de suspensión.

Existen diversos RTOS algunos son libres como Linux-RT, FreeRTOS, etc. o comerciales VxWorks, Windows CE, entre otros. Por ejemplo, en Linux se puede tener planificación en tiempo real en dos formas, la primera es utilizar un parche al kernel para permitir la política de apropiación, la segunda es utilizar un kernel dual, como lo es RTAI o Xenomai [24, p. 360].

Comunicaciones en tiempo real

Mientras que en el cómputo el procesador es el recurso compartido a asignar y la tarea lo que utiliza el recurso, en las comunicaciones, el recurso compartido es el canal de comunicación y quien lo utiliza es el mensaje entre el remitente I y destinatario J ${}^{I}_{J}\mu_i(r, L, \delta_L)$. Los parámetro temporal de interés en las comunicaciones es el tiempo de liberación del mensaje $r\left({}^{I}_{J}\mu_i\right)$ y el

retardo o latencia $L\left({}_{J}^{I}\mu_i\right)$, es decir, el tiempo que tarda un mensaje en transferirse desde el nodo fuente I hasta nodo J, en general la latencia dependerá de la capacidad o ancho de banda al momento de enviar el mensaje y del protocolo de comunicación [25, p. 472]. Al igual que el tiempo de completitud en las tareas, la latencia del mensaje es susceptible a la variabilidad $\delta_L\left({}_{J}^{I}\mu_i\right)$.

En los sistemas de comunicación en tiempo real se define como la calidad de servicio a las políticas utilizadas para disminuir los efectos de latencia y aumentar la capacidad del canal. Los protocolos de planificación de mensajes se definen en la capa de control de acceso al medio (MAC por sus siglas en inglés) [26]. Los protocolos de acceso al medio pueden ser clasificados en tres formas: acceso por multiplexado, por acceso aleatorio o por multiplexado estadístico. En las comunicaciones inalámbricas el acceso aleatorio al medio es el más utilizado.

El middleware es el componente de software que permite compartir información entre sistemas heterogéneos u homogéneos que se encuentran en una red de computadoras. El middleware permite la interacción entre nodos de la red mediante distintos métodos, estos pueden ser síncronos, como la llamada de proceso remoto; o asíncronos, como el paso de mensajes. Los métodos síncronos, en general, deterioran la respuesta en las comunicaciones de tiempo real, ya que dependen del tiempo de ejecución del proceso o tarea remota, sí el proceso o tarea remota falla en dar respuesta, también fallará la tarea que invoca. Los métodos asíncronos son muy utilizados en las comunicaciones en tiempo real. un ejemplo de ellos es la interacción publicador/subscriptor como es el caso de RT-CORBA, TAO y DDS [27].

2.4 Trabajos relacionados

El problema de la fusión de mapas es la alineación de los mapas independientes de los robots para crear un único mapa global del entorno, con mayores características que los mapas individuales. En [28] proponen un sistema de dos robots, donde la estimación tanto de la pose como de los robots como de los puntos de referencia se realiza mediante el filtro extendido de Kalman (EKF por sus siglas en inglés). La fusión de mapas comienza procesando la distancia y orientación relativa entre los robots, así cuando los dos robots se encuentran dentro del rango de detección uno del otro, sucede la fusión. La transformación entre los dos mapas se calcula utilizando los parámetros mencionados, sí los mapas creados por los dos robots se superponen, se pueden identificar los puntos de referencia duplicados en el mapa fusionado. Una vez que los dos mapas se fusionan, los dos robots pueden continuar explorando de forma cooperativa su

entorno mientras expresan todas las nuevas medidas con respecto al marco de referencia común, es decir, a partir del marco global. Para reducir la complejidad computacional de fusionar dos mapas se realiza una búsqueda en un árbol kd. Este trabajo se puede extender para más de dos robots. Para este caso se utilizaron sensores laser para determinar las distancias y orientaciones de los puntos de referencia, y cámaras de color para detectar a los demás robots.

En [29] se presenta una propuesta para SLAM distribuido, el cual presenta un algoritmo de estimación de estado descentralizado, el cual puede trabajar en redes de robots dinámicas, en el mismo se demuestra que el robot solo necesita conocer el estado de vecinos y no de la red en general. Por tanto, el algoritmo presenta un buen de rendimiento en diferentes configuraciones de conectividad de red, así como la capacidad de escalabilidad.

2.5 Resumen

En las secciones anteriores se presentó al sistema multi-robot como un sistema multi-agente cooperativo. Un sistema multi-robot es un ejemplo de implementación de un DRTS, ya que es un conjunto de sistemas autónomos que están en contacto con el ambiente físico, por tanto, se requiere en cada uno de ellos cómputo con capacidades en tiempo real, y son múltiples robots comunicados a través de una red de comunicación.

La aplicación por realizar es la fusión de mapas locales de cada robot, en uno global. Esto se logra utilizado la fusión de datos, donde los mapas locales se representan mediante puntos de referencia (ver A.B).

Capítulo 3. Descripción del algoritmo FDIE-C

El FDIE-C busca la fusión de mapas locales de múltiples robots móviles en uno global, cada robot tiene una versión parcial (local) del mapa explorado, conforme avanzan comparten esta información con sus pares, y crean un mapa global con mayor precisión y puntos de referencia, que los mapas locales.

El FDIE-C consta de cuatro etapas, en la primera y segunda etapa se realiza la actualización de movimiento y de la observación, estos algoritmos se describen en A.A y A.B respectivamente. En estos algoritmos se actualiza la estimación del estado debido al movimiento del robot, mientras explora el ambiente y la actualización de estado por observación de los puntos de referencia.

Breve descripción en Bayes del movimiento y la observación

$FDIE-C\ (\mathbf{Y}(k-1\|k-1), \hat{\boldsymbol{y}}(k-1\|k-1), \hat{\boldsymbol{x}}(k-1\|k-1), \boldsymbol{u}(k), \boldsymbol{z}(k))$		
1	$[\mathbf{Y}(k\|k-1), \hat{\boldsymbol{y}}(k\|k-1), \hat{\boldsymbol{x}}(k\|k-1)]$ $= movimiento\big(\mathbf{Y}(k-1\|k-1), \hat{\boldsymbol{y}}(k-1\|k-1), \hat{\boldsymbol{x}}(k\|k-1), \boldsymbol{u}(k)\big),$	Algoritmo A.1
2	$[\mathbf{Y}(k\|k), \hat{\boldsymbol{y}}(k\|k), \hat{\boldsymbol{x}}(k\|k)]$ $= observación\big(\mathbf{Y}(k\|k-1), \hat{\boldsymbol{y}}(k\|k-1), \hat{\boldsymbol{x}}(k\|k-1), \boldsymbol{z}(k)\big),$	Algoritmo A.2
3	$[\mathbf{Y}(k\|k), \hat{\boldsymbol{y}}(k\|k)] = FDIE\big(\mathbf{Y}(k\|k), \hat{\boldsymbol{y}}(k\|k)\big),$	Algoritmo 3.1
4	**if** $\left(\boldsymbol{m}^{\mathbf{0}}\left({}^{I}k\right) \neq \boldsymbol{m}^{\mathbf{0}}\left({}^{I}k-1\right)\right)$	
5	$[\mathbf{Y}(k\|k), \hat{\boldsymbol{y}}(k\|k)] = FC\big(\mathbf{Y}(k\|k), \hat{\boldsymbol{y}}(k\|k), \hat{\boldsymbol{x}}(k\|k)\big),$	Algoritmo 3.3
6	**end**	
return$\big(\mathbf{Y}(k\|k), \hat{\boldsymbol{y}}(k\|k), \hat{\boldsymbol{x}}(k\|k)\big)$		

Algoritmo 3.1

En la tercera etapa se usa el filtro disperso de información extendida (FDIE por sus siglas), el cual es derivado del filtro información extendida (EIF por sus siglas en inglés), con EIF la

estimación del estado del robot (posición y mapa del robot) se representa mediante vector y matriz de información, esta representación permite que las nuevas observaciones de los robots se puedan agregar de manera aditiva. Al dispersar la matriz de información en FIE se puede actualizar tanto la posición y mapa locales con una complejidad $O(M)$ [5] [6]. Este algoritmo se presenta a mayor detalle en la siguiente sección

La cuarta etapa del algoritmo es el filtro de consenso (-C), el cual los robots llegan a un consenso de los puntos de referencia, es decir la fusión de los mapas locales en uno global. Al tener las estimaciones del estado del robot en su forma de información, la fusión en el robot I se logra mediante la adición de las estimaciones de los puntos de referencia de los robot vecinos ($N(I)$) [30, p. 501], este algoritmo tiene una complejidad $O(N(I)M)$ en cada nodo de la red de robots. El filtro de consenso es utilizado en redes de sensores, permite que la fusión de las estimaciones del estado de la red de sensores se pueda realizar de manera escalable, es decir, se pueden añadir nodos a la red conforme se realizan las estimaciones locales. También, con este filtro se logra la convergencia en la fusión de las estimaciones, aun cuando se presentan cambios en la topología y retrasos en la red [31].

3.1 Filtro disperso de información extendida

El objetivo del FDIE es obtener una matriz de información dispersa, para lograr que la actualización de la estimación de estado se realice en tiempo lineal $O(M)$, respecto al número de puntos de referencia, en lugar de un tiempo cuadrático $O(M^2)$ comoen EKF. Para ello, se requiere eliminar relaciones dependencia condicional entre el estado del robot ($\boldsymbol{r}_I({}^I k), \boldsymbol{m}^+({}^I k)$) y aquellos puntos pasivos pasivos $\boldsymbol{m}^-({}^I k)$**,** es decir puntos que ya han sido observados antes de ${}^I k$, para realizarlo de forma secuencial, se elimina la dependencia de aquellos puntos que pasan de activos a ser pasivos $\boldsymbol{m^0}({}^I k)$, estos puntos en el futuro próximo pasan a formar parte de los puntos pasivos $\boldsymbol{m}^-({}^I k)$. Es decir:

$$\boldsymbol{m^0}({}^I k + K) \, \epsilon \, \boldsymbol{m}^-({}^I k), \qquad K > 0, \tag{3.1}$$

El procedimiento mostrado en los siguientes párrafos es similar al que se encuentra en modelos de redes probabilísticas [32]. Para iniciar se define la siguiente notación:

$$\boldsymbol{m}({}^I k) = \boldsymbol{m}^+({}^I k) \cup \boldsymbol{m^0}({}^I k) \cup \boldsymbol{m}^-({}^I k). \tag{3.2}$$

En (3.2) los puntos de referencia $\boldsymbol{m}({}^I k)$ son la unión de los puntos activos $\boldsymbol{m}^+({}^I k)$, activos a pasivos $\boldsymbol{m^0}({}^I k)$ y pasivos $\boldsymbol{m}^-({}^I k)$ en el instante ${}^I k$; utilizando el hecho que el filtro de

información extendido es un filtro de Bayes en un modelo oculto de Markov [5]. Se tiene lo siguiente.

$$p\left(\boldsymbol{r}_I(\ ^I k), \boldsymbol{m}(\ ^I k)\right) = p(\boldsymbol{r}_I(\ ^I k), \boldsymbol{m}^+(\ ^I k), \boldsymbol{m}^0(\ ^I k), \boldsymbol{m}^-(\ ^I k)). \tag{3.3}$$

El primer paso es aplicar el teorema de Bayes en (3.3).

$$p\left(\boldsymbol{r}_I(\ ^I k), \boldsymbol{m}(\ ^I k)\right) = p(\boldsymbol{r}_I(\ ^I k)|\boldsymbol{m}^+(\ ^I k), \boldsymbol{m}^0(\ ^I k), \boldsymbol{m}^-(\ ^I k))\, p(\boldsymbol{m}^+(\ ^I k), \boldsymbol{m}^0(\ ^I k), \boldsymbol{m}^-(\ ^I k)) \tag{3.4}$$

Para eliminar las dependencias condicionales con $\boldsymbol{m}^0(\ ^I k)$ en (3.4), se puede realizar una marginalización de $\boldsymbol{m}^0(\ ^I k)$, con ello la información de estos puntos de referencia se almacena en las demás variables.

$$p\left(\boldsymbol{r}_I(\ ^I k), \boldsymbol{m}(\ ^I k)\right) \approx \int p\left(r(\ ^I k), \boldsymbol{m}^+(\ ^I k), \boldsymbol{m}^0(\ ^I k), \boldsymbol{m}^-(\ ^I k)\right)\, d\boldsymbol{m}^0, \tag{3.5}$$

$$p\left(\boldsymbol{r}_I(\ ^I k), \boldsymbol{m}(\ ^I k)\right) \simeq p(\boldsymbol{r}_I(\ ^I k)|\boldsymbol{m}^+(\ ^I k), \boldsymbol{m}^-(\ ^I k)) p(\boldsymbol{m}^+(\ ^I k), \boldsymbol{m}^-(\ ^I k)).$$

Para simplificar aún más (3.5) podemos hacer el mismo procedimiento para los puntos pasivos, sin embargo, esto, ya se ha realizado en instantes menores a $^I k$, como lo describe (3.1). Por lo tanto, podemos asignarle a esta variable cualquier valor.

$$p\left(\boldsymbol{r}_I(\ ^I k), \boldsymbol{m}(\ ^I k)\right) \simeq p(\boldsymbol{r}_I(\ ^I k)|\boldsymbol{m}^+(\ ^I k), \boldsymbol{m}^-(\ ^I k) = \mathbf{0}) p\left(\boldsymbol{m}^+(\ ^I k), \boldsymbol{m}^-(\ ^I k)\right). \tag{3.6}$$

Para realizar un menor número de operaciones, como se demostrará más adelante, se utiliza de nuevo el teorema de Bayes, en el primer término del lado derecho (3.6).

$$p\left(\boldsymbol{r}_I(\ ^I k), \boldsymbol{m}(\ ^I k)\right) \simeq \frac{p(\boldsymbol{r}_I(\ ^I k), \boldsymbol{m}^+(\ ^I k)|\boldsymbol{m}^-(\ ^I k) = \mathbf{0})}{p(\boldsymbol{m}^+(\ ^I k)|\boldsymbol{m}^-(\ ^I k) = \mathbf{0})} p\left(\boldsymbol{m}^+(\ ^I k), \boldsymbol{m}^-(\ ^I k)\right). \tag{3.7}$$

Cada uno de los tres términos del lado derecho representa el estado del robot mediante una distribución de probabilidad Gaussiana, esta distribución se define con los vectores y matriz de información [33].

$$p(\boldsymbol{r}_I(\ ^I k), \boldsymbol{m}^+(\ ^I k)|\boldsymbol{m}^-(\ ^I k) = \mathbf{0}) \rightarrow \tag{3.8}$$

$$\hat{\boldsymbol{y}}_1(k|k) = \hat{\boldsymbol{y}}(k|k)[rm^+] - \mathbf{Y}(k|k)[rm^+][m^0]\{\mathbf{Y}(k|k)[m^0][m^0]\}^{-\mathbf{1}}\hat{\boldsymbol{y}}(k|k)\left[\boldsymbol{m}^0\right],$$

$$\mathbf{Y}_1(k|k) = \mathbf{Y}(k|k)[rm^+][rm^+] - \mathbf{Y}(k|k)[rm^+][m^0]\{\mathbf{Y}(k|k)[m^0][m^0]\}^{-\mathbf{1}}\mathbf{Y}(k|k)[m^0][rm^+],$$

$$p(\boldsymbol{m}^+(\ ^I k)|\boldsymbol{m}^-(\ ^I k) = \mathbf{0}) \rightarrow \tag{3.9}$$

$$\hat{\boldsymbol{y}}_2(k|k) = \hat{\boldsymbol{y}}(k|k)[m^+] - \mathbf{Y}(k|k)[m^+][rm^0]\{\mathbf{Y}(k|k)[rm^0][rm^0]\}^{-\mathbf{1}}\hat{\boldsymbol{y}}(k|k)\left[rm^{\mathbf{0}}\right],$$

$$\mathbf{Y}_2(k|k) = \mathbf{Y}(k|k)[m^+][m^+] - \mathbf{Y}(k|k)[m^+][rm^0]\{\mathbf{Y}(k|k)[rm^0][rm^0]\}^{-\mathbf{1}}\mathbf{Y}(k|k)[rm^0][m^+],$$

$$p\left(\boldsymbol{m}^+(\ ^I k), \boldsymbol{m}^-(\ ^I k)\right) \rightarrow \tag{3.10}$$

$$\hat{\boldsymbol{y}}_3(k|k) = \hat{\boldsymbol{y}}(k|k)[m^+m^-] - \mathbf{Y}(k|k)[m^+m^-][rm^0]\{\mathbf{Y}(k|k)[rm^0][rm^0]\}^{-\mathbf{1}}\hat{\boldsymbol{y}}(k|k)\left[rm^{\mathbf{0}}\right],$$

$$\mathbf{Y}_3(k|k) = \mathbf{Y}(k|k)[m^+m^-][m^+m^-] - \mathbf{Y}(k|k)[m^+m^-][\boldsymbol{r}m^0]\{\mathbf{Y}(k|k)[rm^0][rm^0]\}^{-\mathbf{1}}\mathbf{Y}(k|k)[rm^0][m^+m^-].$$

Dado que, La razón de descomponer el primer término del lado derecho de (3.6) como se muestra en (3.7), se debe a que la inversión de matriz es un proceso de cómputo con una complejidad cúbica, y como en general la cardinalidad de los puntos pasivos es mayor a la cantidad puntos activos, puntos de activos a pasivos y de pose de los robots $card(\boldsymbol{m}^+) < card(\boldsymbol{m}^-)$, $card(\boldsymbol{m}^{\mathbf{0}}) < card(\boldsymbol{m}^-)$, $card(\boldsymbol{r}_I) < card(\boldsymbol{m}^-)$, es preferente realizar la inversión de matrices con dimensiones menores como en $\{\mathbf{Y}(k|k)[rm^0][rm^0]\}^{-\mathbf{1}}$ y $\{\mathbf{Y}(k|k)[m^0][m^0]\}^{-\mathbf{1}}$, además $\{\mathbf{Y}(k|k)[rm^0][rm^0]\}^{-\mathbf{1}}$ se repite en dos operaciones, por tanto, ya no es necesario calcularlo nuevamente.

$[\mathbf{Y}(k\|k), \hat{\boldsymbol{y}}(k\|k)] = dispersion\left(\mathbf{Y}(k\|k), \hat{\boldsymbol{y}}(k\|k)\right)$		
1	$\hat{\boldsymbol{y}}_1(k\|k) = \hat{\boldsymbol{y}}(k\|k)[rm^+] - \mathbf{Y}(k\|k)[rm^+][m^0]\{\mathbf{Y}(k\|k)[m^0][m^0]\}^{-\mathbf{1}}\hat{\boldsymbol{y}}(k\|k)\left[m^{\mathbf{0}}\right]$, $\mathbf{Y}_1(k\|k) = \mathbf{Y}(k\|k)[rm^+][rm^+] - \mathbf{Y}(k\|k)[rm^+][m^0]\{\mathbf{Y}(k\|k)[m^0][m^0]\}^{-\mathbf{1}}\mathbf{Y}(k\|k)[m^0][rm^+]$.	(3.8)
2	$\hat{\boldsymbol{y}}_2(k\|k) = \hat{\boldsymbol{y}}(k\|k)[m^+] - \mathbf{Y}(k\|k)[m^+][rm^0]\{\mathbf{Y}(k\|k)[rm^0][rm^0]\}^{-\mathbf{1}}\hat{\boldsymbol{y}}(k\|k)\left[rm^{\mathbf{0}}\right]$, $\mathbf{Y}_2(k\|k) = \mathbf{Y}(k\|k)[m^+][m^+] - \mathbf{Y}(k\|k)[m^+][\boldsymbol{r}m^0]\{\mathbf{Y}(k\|k)[rm^0][rm^0]\}^{-\mathbf{1}}\mathbf{Y}(k\|k)[rm^0][m^+]$,	(3.9)
3	$\hat{\boldsymbol{y}}_3(k\|k) = \hat{\boldsymbol{y}}(k\|k)[m^+m^-] - \mathbf{Y}(k\|k)[m^+m^-][rm^0]\{\mathbf{Y}(k\|k)[rm^0][rm^0]\}^{-\mathbf{1}}\hat{\boldsymbol{y}}(k\|k)\left[rm^{\mathbf{0}}\right]$, $\mathbf{Y}_3(k\|k) = \mathbf{Y}(k\|k)[m^+m^-][m^+m^-] - \mathbf{Y}(k\|k)[m^+m^-][\boldsymbol{r}m^0]\{\mathbf{Y}(k\|k)[rm^0][rm^0]\}^{-\mathbf{1}}\mathbf{Y}(k\|k)[rm^0][m^+m^-]$,	(3.10)
4	$\hat{\boldsymbol{y}}(k\|k) = \hat{\boldsymbol{y}}_1(k\|k) - \hat{\boldsymbol{y}}_2(k\|k) + [\hat{\boldsymbol{y}}_3(k\|k)]$, $\mathbf{Y}(k\|k) = \mathbf{Y}_1(k\|k) - \mathbf{Y}_2(k\|k) + \mathbf{Y}_3(k\|k)$,	
$\text{return}\left(\mathbf{Y}(k\|k), \hat{\boldsymbol{y}}(k\|k)\right)$.		

Algoritmo 3.2

3.2 Filtro de consenso

Los algoritmos de consenso son métodos distribuidos para que un conjunto de agentes acuerde en un específico estado consensado ξ. En el filtro de conseso cada agente I intercambia

información con los agentes de su vecindario $N(I)$, conforme pasa el tiempo, los agentes llegan a un acuerdo o consenso. En control cooperativo el consenso es usado para la coordinación y formación en sistemas multi agente [34].

Para poder representar el algoritmo de consenso se utiliza teoría de grafos [35, pp. 14-33]. En cualquier instante de tiempo k, la topología de las comunicaciones entre los agentes se puede describir por un grafo $G(t) = G(V, E(k))$, donde los nodos $V = \{1, \ \dots, \ N\}$ representan cada uno de los agentes y las aristas $E(k)$ son los enlaces de comunicación entre dos nodos. El protocolo de consenso promedio se representa mediante un sistema dinámico de primer orden

$$ {}^{I}\boldsymbol{\xi}\big({}^{I}k\big) = {}^{I}\boldsymbol{\xi}\big({}^{I}k-1\big) - \kappa \sum_{J \,\epsilon\, N(J)} \Big({}^{I}\boldsymbol{\xi}\big({}^{I}k-1\big) - {}^{J}\boldsymbol{\xi}\big({}^{I}k-1\big)\Big). \tag{3.11} $$

Este protocolo busca el consenso o acuerdo entre los agentes, tal que:

$$ \lim_{{}^{I}k \,\to\, \infty} \Big\| {}^{I}\boldsymbol{\xi}\big({}^{I}k\big) - {}^{I}\boldsymbol{\xi}\big({}^{I}k-1\big)\Big\| = 0, \quad \forall\, I, J \,\epsilon\, V. \tag{3.12} $$

Con las características anteriores el sistema descrito por (3.11) tiene la siguiente representación.

$$ {}^{I}\boldsymbol{\xi}\big({}^{I}k\big) = (\boldsymbol{I} - \kappa \boldsymbol{L}(G))\ {}^{I}\boldsymbol{\xi}(k-1). \tag{3.13} $$

En (3.13) $\boldsymbol{L}(G)$ es el laplaciano de la red de interacción (comunicación directa) entre los agentes. Con lo anterior es posible definir el comportamiento del sistema dinámico global. Para alcanzar la condición (3.12), el laplaciano de G, debe tener las siguientes caracteristicas:

Todos los eigenvalores de $(\boldsymbol{I} - \kappa \boldsymbol{L}(G))$ son no negativos $\lambda_s \leq 1$, es decir, la red esta conectada.

Solo existe un eigenvalor $\lambda_1 = 1$ con eigenvector derecho $\boldsymbol{v}_1 = \boldsymbol{1} = [1,1, \dots 1]\, \epsilon\, \mathbb{R}^m$, e izquierdo $\boldsymbol{u}_1 = \boldsymbol{1}/m$

Para que las condiciones anteriores sucedan el valor de κ debe encontrarse en el siguiente rango $\kappa\, \epsilon\, [0, 1/\Delta]$, donde Δ se define como el grado máximo de un nodo en la red;

Entonces la ecuación se puede representar como

$$ {}^{I}\boldsymbol{\xi}(k) = \big({\lambda_1}^{k}\boldsymbol{u}_1\boldsymbol{v}_1^T + {\lambda_2}^{k}\boldsymbol{u}_2\boldsymbol{v}_2^T \dots + {\lambda_n}^{k}\boldsymbol{u}_n\boldsymbol{v}_n^T\big)\ {}^{I}\boldsymbol{\xi}(0) $$

Donde ${}^{I}\boldsymbol{\xi}(0)$ son las condiciones iniciales de los agentes, $\boldsymbol{u}_s$ y $\boldsymbol{v}_s$ son los eigenvectores izquierdo y derecho respectivamente de cada eigenvalor λ_s. Dado que $\lambda_i < \lambda_j$ para $\forall\ 1 \leq i < j$, entonces la ecuación (3.13) en tiempo infinito:

$$ \lim_{k\to\infty}\ {}^{I}\boldsymbol{\xi}(k) = \lim_{t\to\infty}\big({\lambda_1}^{k}\boldsymbol{u}_1\boldsymbol{v}_1^T + {\lambda_2}^{k}\boldsymbol{u}_2\boldsymbol{v}_2^T \dots + {\lambda_n}^{k}\boldsymbol{u}_n\boldsymbol{v}_n^T\big)\ {}^{I}\boldsymbol{\xi}(0). \tag{3.14} $$

$$ \lim_{\mathrm{t}\to\infty}\ {}^{I}\boldsymbol{\xi}(k) = \boldsymbol{u}_1\boldsymbol{v}_1^T\ {}^{I}\boldsymbol{\xi}(0) = \frac{\boldsymbol{1}}{m}(\boldsymbol{1}^T\ {}^{I}\boldsymbol{\xi}(0)) = \frac{1}{m}\sum_{I\,\epsilon\,V}\ {}^{I}\boldsymbol{\xi}(0). $$

El término del lado derecho es el valor promedio de las condiciones iniciales de cada uno de los agentes. De este modo se obtiene el consenso descrito por (3.11)**.** La rapidez del consenso

dependerá en general del segundo eigenvalor del Laplaciano de G [35]. La robustez a retardos en el tiempo y cambio en la topología en la red dependerán del parámetro de κ, en [36] se propone $\kappa = \frac{1}{N(J)+1}$, con este valor la ecuación (3.11), se puede expresar de forma más sencilla como.

$$ {}^{I}\boldsymbol{\xi}\left({}^{I}k\right) = \frac{1}{N(J)+1}\left({}^{I}\boldsymbol{\xi}\left({}^{I}k-1\right) + \sum_{J \in N(J)} {}^{J}\boldsymbol{\xi}\left({}^{I}k-1\right)\right). \tag{3.15} $$

En el caso de la fusión de los mapas el estado de consenso ${}^{I}\boldsymbol{\xi}$, son el vector de información y matriz de información de cada nodo $\mathbf{Y}(k|k), \hat{\boldsymbol{y}}(k|k)$.

	$[\mathbf{Y}(k\|k), \hat{\boldsymbol{y}}(k\|k)] = fusión\left(\mathbf{Y}(k\|k), \hat{\boldsymbol{y}}(k\|k), \hat{\boldsymbol{x}}(k\|k)\right),$	
1	$${}^{I}\hat{\boldsymbol{y}}(k|k) = \frac{1}{N(J)+1}\left({}^{I}\hat{\boldsymbol{y}}(k|k) + \sum_{J \in N(I)} {}^{J}\hat{\boldsymbol{y}}(k|k)\right).$$	(3.15)
2	$${}^{I}\mathbf{Y}(k|k) = \frac{1}{N(J)+1}\left({}^{I}\mathbf{Y}(k|k) + \sum_{J \in N(I)} {}^{J}\mathbf{Y}(k|k)\right),$$	(3.15)
	$\text{return}\left(\mathbf{Y}(k\|k), \hat{\boldsymbol{y}}(k\|k)\right).$	

Algoritmo 3.3

Capítulo 4. Implementación y resultados

Para la implementación se simularon cuatro robots, similar a un espacio de oficina como se muestra en la Figura 4.1. El código se desarrolló en Octave/Matlab con el objetivo de reutilizar este código en robots físicos con el uso de frameworks o middleware enfocados a robots móviles, como son ROS, Octave o Microsoft Robotic Studio.

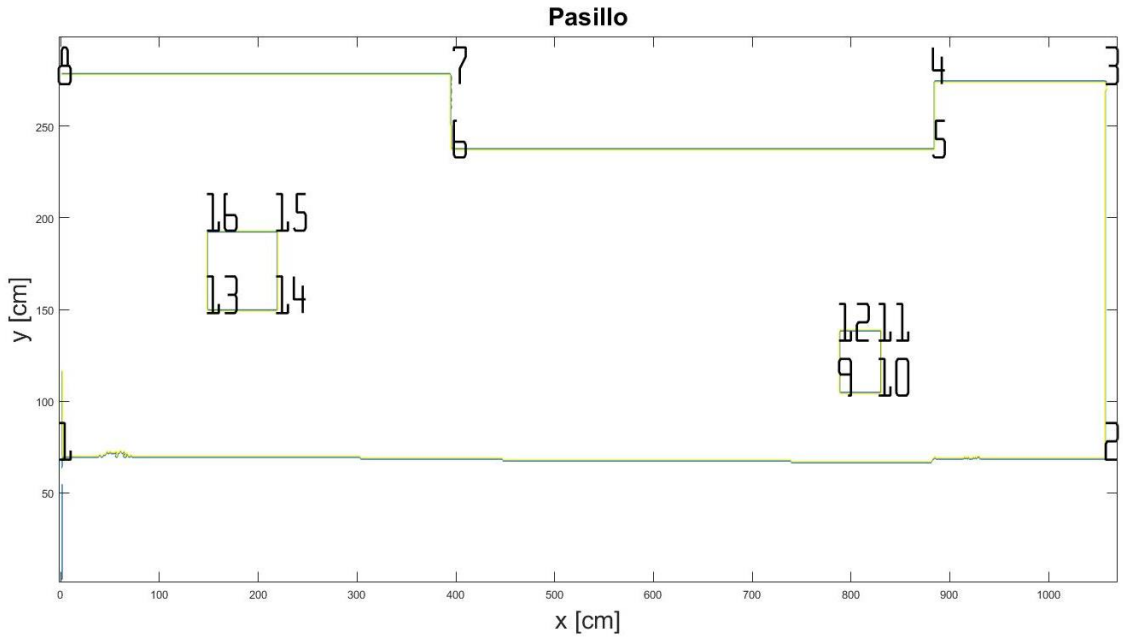

Figura 4.1 Espacio de Simulación

El espacio de simulación está basado en el pasillo cercano al Laboratorio de Control del edificio del IIMAS, se modela un tramo de 1050 [cm] de largo por 200 [cm] de ancho, se colocaron dos obstáculos como se observa a la izquierda y derecha de la Figura 4.1**.** El sistema de coordenadas el mismo para todos los robots y el origen es la esquina inferior izquierda.

En cada uno de los robots se simulan acciones, observaciones, estimaciones de estado, fusión de mapas. Cada robot se modela como un robot diferencial con 35 [cm] de diámetro, pueden alcanzar una velocidad máxima de 100 [cm/s]. El sensor modelado es un láser que puede medir un rango entre 50 [cm] y 500 [cm] con una resolución de 0.5 [cm], tiene una apertura de 120 [°] con una resolución de 1 [°]. El estado del robot I está definido por la pose del mismo, posición $(X_I\ [cm], Y_I [cm])$ y orientación $(\theta_I [rad])$, y 16 posiciones de los puntos del mapa

$(X_P\ [cm], Y_P[cm])$ dando un total de 19 variables de estado. Los robots inician la simulación conociendo su pose inicial y final. La matriz de información se inicializa con valores cercanos a cero en el mapa y valores superiores a cero, respecto a la pose, debido a que se conoce con certeza desde el inicio; de manera similar el vector de información se inicializa en cero en sección del mapa.

Dado que cada robot tiene su propio estado, su pose y un mapa local, por separado se almacenan el mapa fusionado, que se actualiza con la información propia y de los robots vecinos. En condiciones ideales con una red de comunicación siempre conectada se espera que todos los mapas sean idénticos, es decir, la fusión por consenso distribuido se comporta como una fusión por consenso centralizado.

Respecto a la simulación de las comunicaciones, la topología de la red varía en cada instante de manera aleatoria, eliminado en cada instante del tiempo distintos enlaces de la red. Los eventos de comunicación entre los robots para compartir información en el consenso suceden cuando se cambian el conjunto de puntos de activo a pasivo, es decir, cuando sucede un cambio significativo en el vector información y la matriz de información.

En los siguientes apartados se definen las tareas a realizar por cada robot, así como su prioridad, dependencia, periodicidad y complejidad del algoritmo. Posteriormente presentan los resultados de la simulación con ciertas condiciones específicas.

4.1 Planificación de tareas en tiempo real

Con el algoritmo FDIE-C se cubre el apartado de la fusión de mapas. Para definir cómo se resuelve dicho algoritmo en tiempo es necesario definir las tareas, estas se pueden resumir en dos actividades principales

- Navegación autónoma
- Fusión de modelos

La navegación autónoma es local, es decir cada nodo (robot), tiene la capacidad de exploración, evasión de obstáculos, y percepción del mundo. Mientras que la fusión de modelos es distribuida, para obtenerla es necesario que cada robot estime su estado y lo comparta con los demás nodos.

Actividad	Tarea	Prioridad	Dependencia	Periodicidad	Complejidad
Navegación Autónoma	Planeación de movimiento para evasión de obstáculos	1	Extracción de puntos de referencia	Periódica	$O(1)$
Navegación Autónoma	Extracción de puntos de referencia	2		Periódica	$O(N_{obs} \ln N_{obs})$
Fusión de modelos	Estimación local	3	Extracción de puntos de referencia	Periódica	$O(M)$
Fusión de modelos	Consenso	4	Estimación local	No periódica	

4.2 Resultados

Se definieron condiciones específicas de la simulación, con el objetivo de observar que la fusión de mapas mediante consenso es robusta a pesar de estas condiciones. El robot en rojo (ver Figura 4.2 Simulación en instante t = 0[s]

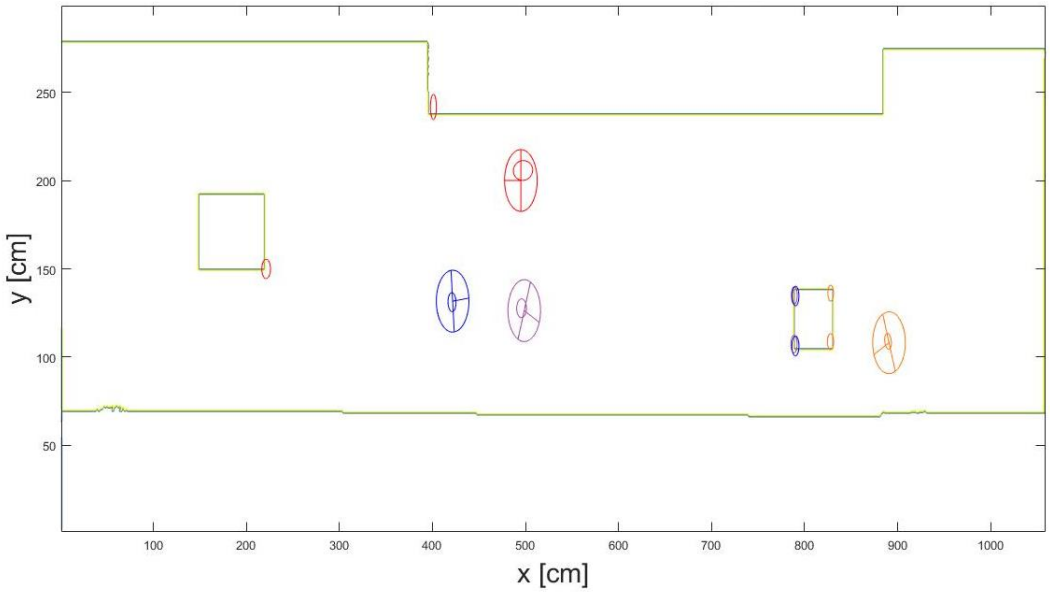

Figura 4.2) tiene una velocidad superior a los demás, con ello se espera que la información de su estado disminuya, además en un lapso de la simulación (6.80[s]-9.40[s]), no observa ningún punto del mapa, por tanto, la actualización de la medición no aumenta su información. El robot naranja ver Figura 4.2 Simulación en instante t = 0[s]

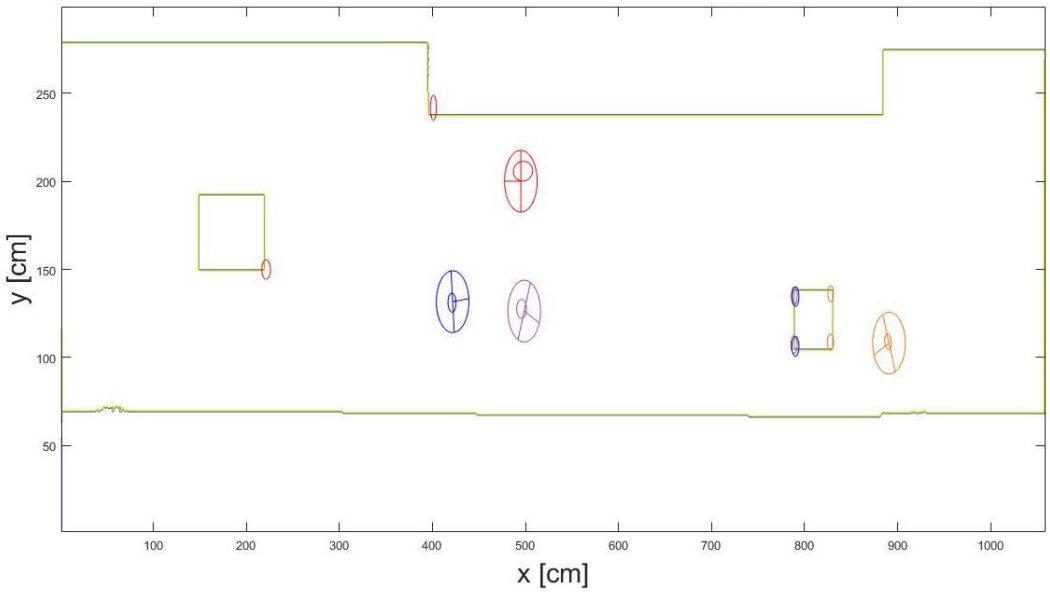

Figura 4.2) tiene una trayectoria, que le permite ver en casi toda la duración de la simulación observar los mismos puntos del mapa. Los robots azul y morado tienen trayectorias que casi no presentan periodos donde los robots no observen algún punto de referencia, además que pueden contener mapas locales similares. Cada uno de los puntos se inicializa con un valor de covarianza de 100 [cm^2], este valor indica que se tiene al principio de la simulación una incertidumbre de 10 [cm], sí lo comparamos con el diámetro del robot (35 [cm]), este valor se elijío experimentalmente, ya que en la teoría este valor debe ser inifinito, sin embargo al invertir este valor numéricamente no tiene significado, si el valor valor de covarianza de 100 [cm^2] si se invierte, en la matriz de información da un valor inicial de 0.01 [cm^{-2}], con este valor los cálculos al inicial la simulación no se ven afectados.

Con las condiciones mencionadas anteriormente se desea observar el aumento del mapa del robot amarillo, a pesar de que el robot observa únicamente una región con pocos puntos del mapa, en la fusión de mapas obtenga información de otros puntos de mapa. Se espera que la matriz de información del robot rojo tenga valores más cercanos a cero, debido a incertidumbre generada por su rápido desplazamiento, además de no actualizar su matriz de información al no observar algún punto del mapa en un lapso considerable. Se espera que el consenso ayude a incrementar la información de su mapa fusionado.

En la Figura 4.2 se muestran las poses iniciales de los robots, rojo, morado, azul y naranja, así como los primeros puntos observados en ese instante. En la Figura 4.3 se presentan las poses finales de los mismos.

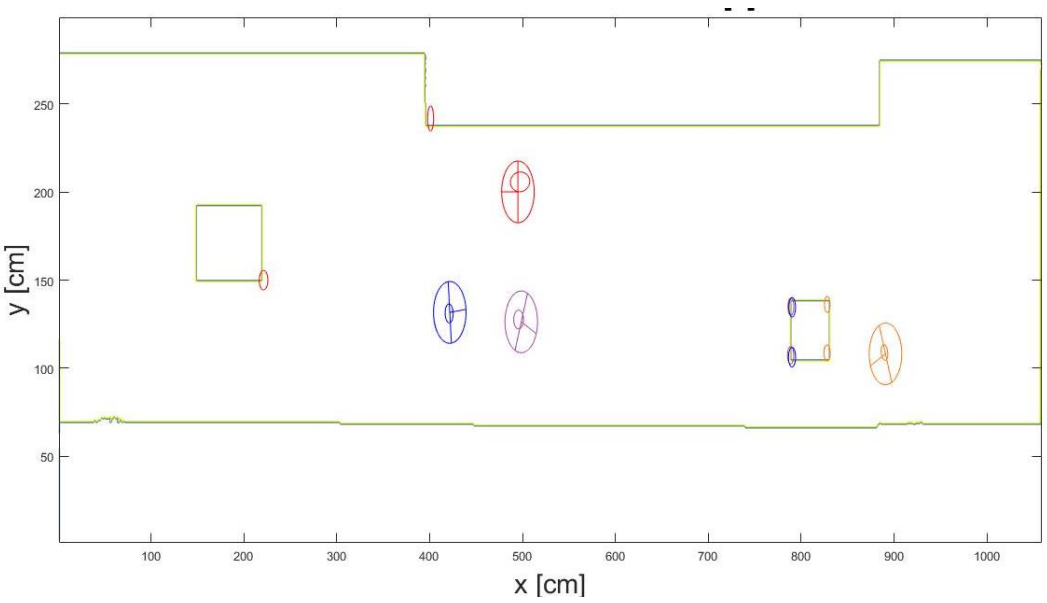

Figura 4.2 Simulación en instante t = 0[s]

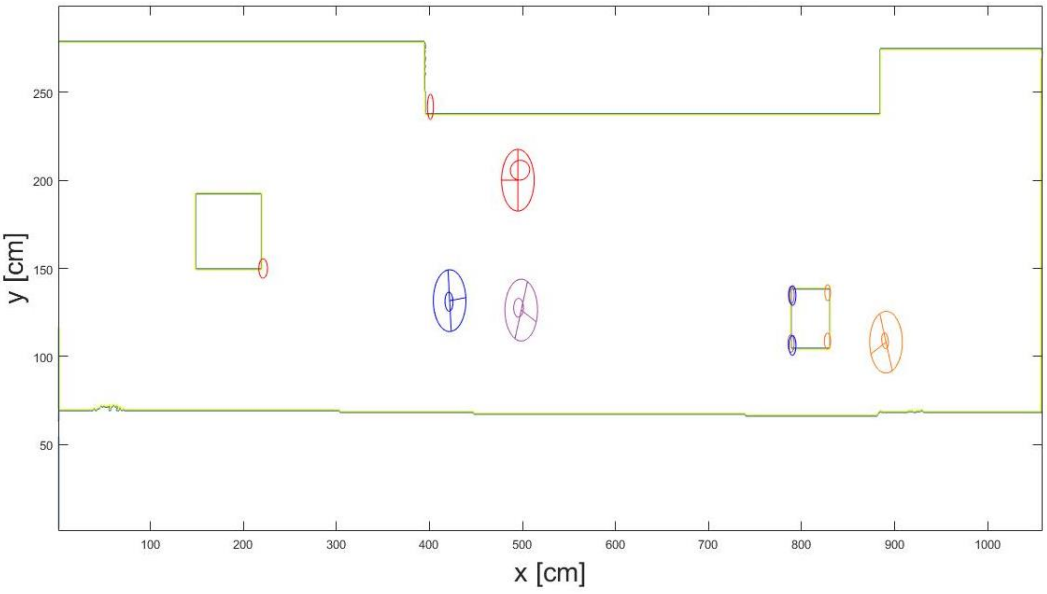

Figura 4.3 Simulación en instante t = 20[s]

En la Figura 4.4 se observa tanto los mapas locales de cada robot, como los mapas fusionados por el consenso, se puede apreciar que en cada uno de ellos tiene un offset en la posición, sin embargo, en general conservan la forma del mapa, como se observa en el obstáculo derecho (puntos 9, 10, 11 y 12). Para entender el problema se utilizan las gráficas de la evolución de la fusión de los mapas, es decir, la evolución del consenso (Figura 4.5 y Figura 4.6).

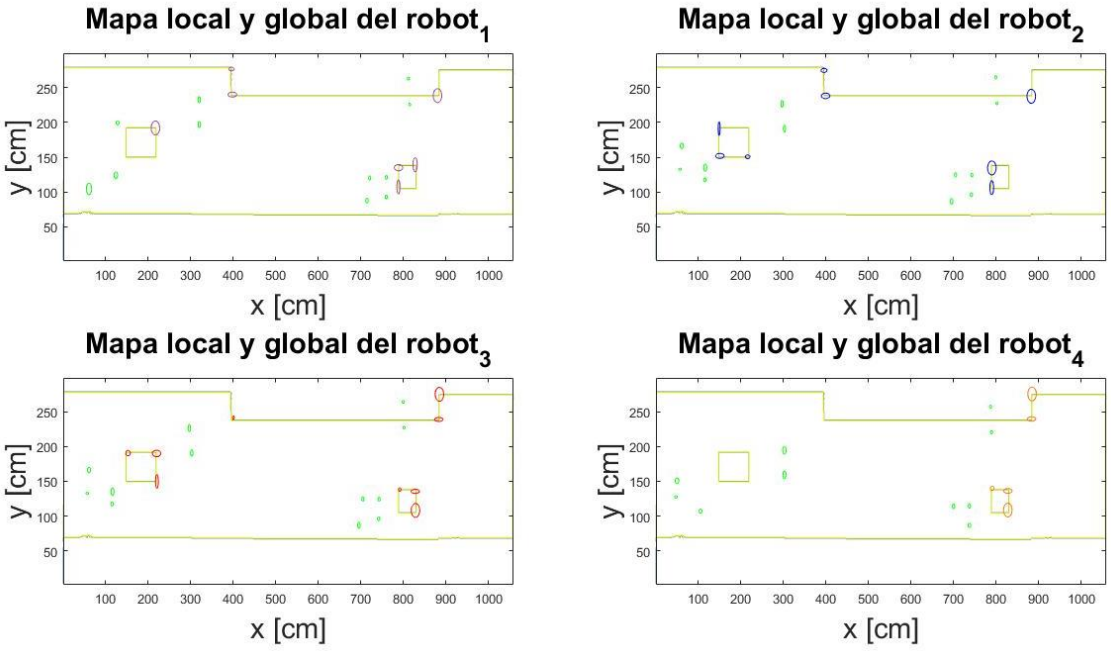

Figura 4.4 Mapas locales y mapas globales de cada uno de los robots de la red

Para la presentación de la evolución del consenso de algunos puntos específicos del mapa, se utiliza la estimación del vector de estados y la matriz de covarianza, en lugar del vector de información y la matriz de información, debido a que estos últimos no tienen representan una variable física del mundo, en este caso los puntos de referencia. Los puntos elegidos son los numerados 7, 11 y 12. El punto 7 corresponde a una esquina en la parte superior central del pasillo, se elige este punto ya que los robots rojo y naranja jamás los observan, pero a través de la fusión de la información de los mapas, lo adquieren. Los puntos 11 y 12 son parte del obstáculo derecho el punto 12 es observado por todos los robots, mientras que el punto 11 es solo observado por el robot naranja.

En la Figura 4.5 se observa en cada color de los robots la evolución del consenso valor real del punto de referencia dibujado en verde. Del lado izquierdo se muestran los valores del consenso en la coordenada X, en el lado derecho se muestran los valores en la coordenada Y. Se puede observar que en general ninguno de los valores alcanza el valor correcto, sin embargo, casi todos los valores alcanzan asintóticamente un valor similar, cada uno tiene un error de estado estable [37, pp. 161-162].

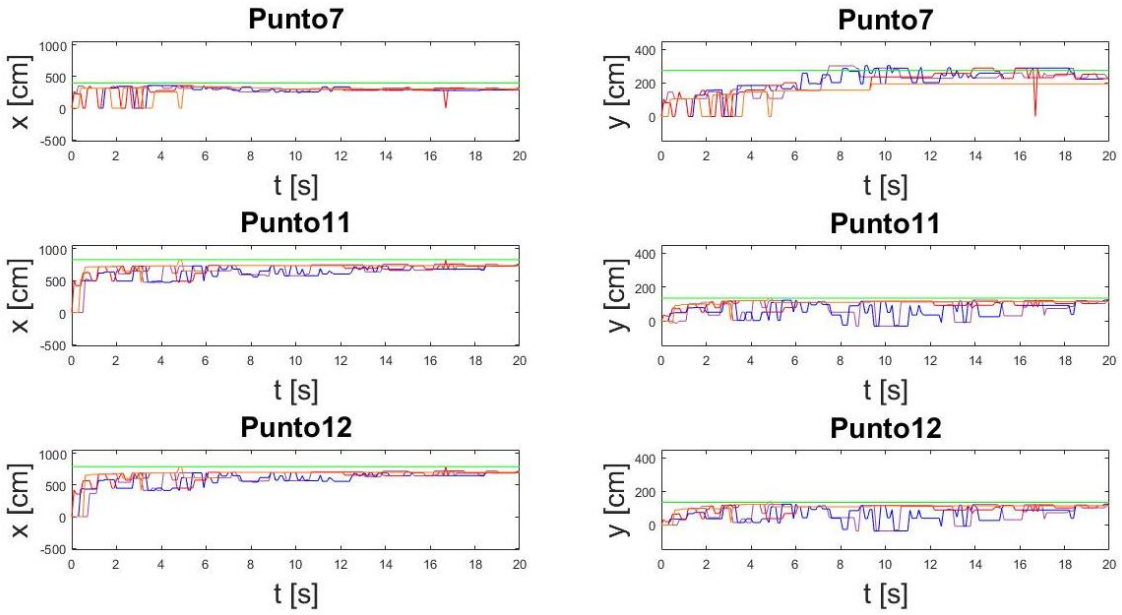

Figura 4.5 Evolución de consenso en la estimación de los puntos 7, 11 y 12

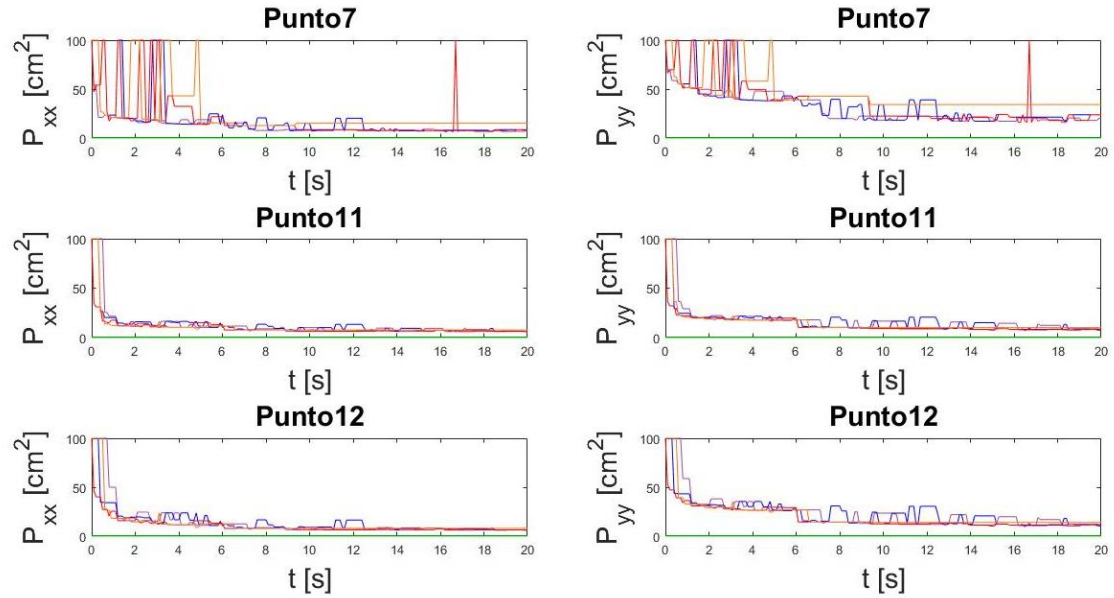

Figura 4.6 Evolución de consenso en la covarianza de los puntos 7, 11 y 12

Un comportamiento similar es observado en la Figura 4.6 en este caso, como ya se mencionó anteriormente, se busca que la covarianza de los puntos se acerque lo más posible a cero, debido a que conforme se explora el mapa se obtiene más información. También se puede apreciar que la covarianza varía mucho al inicio de la simulación, esto se debe, a que los puntos no siempre son observados de manera ininterrumpida por el robot, como es el caso del punto 7, para los robots azul y morado, esto puede ser consecuencia de un desplazamiento rápido del robot

y/o errores en la extracción del punto de referencia. Los puntos 11 y 12 son observados más detenidamente por el robot naranja debido a su baja velocidad y por tanto el valor de la covarianza en ambos puntos desciende más rápidamente.

Capítulo 5. Conclusiones

En esta tesis se presenta la coordinación en tiempo real de un conjunto de robots móviles, conectados a través de una red de comunicaciones. Este capítulo resume las metas más significativas alcanzadas en el trabajo, además de algunas sugerencias de trabajo futuro.

5.1 Logros

La motivación de este trabajo se debe a un incremento de la presencia de robots móviles autónomos en distintos aspectos de la vida humana, es decir, que son sistemas de cómputo con capacidades de observación, actuación y comunicación inmersos en el mundo físico. Al tener un conjunto de robots es necesario considerar la coordinación entre ellos, para que la ejecución de tareas asignadas se realice de manera segura, tanto para el sistema robótico, como para los usuarios en contacto con el mismo. Por la misma razón, es necesario que estos sistemas cumplan con las restricciones en su tiempo de respuesta, es decir, que su respuesta tiene que ser determinística, tanto en los objetivos de las tareas, como en tiempo.

Uno de logros importantes a destacar es la conjunción de algoritmos de baja complejidad ya que esto es útil para cumplir con el tiempo real. En la planeación de movimientos se utilizan potenciales artificiales, que se pueden computar en tiempo constante. La extracción de los puntos de interés mediante el algoritmo de división y fusión (Split and merge en inglés [38]) que se puede lograr en tiempo $O(N_{obs} \ln N_{obs})$.

El filtro para de estimación de estado se basa en modelos probabilísticos basados en grafos, este tipo de algoritmos son muy utilizados para reducir el número de variables de estado, mediante la eliminación por dependencia condicional, con ello ayuda a reducir la cantidad de espacio de memoria, así como el tiempo de respuesta del filtro.

El filtro de consenso demostró ser estable a las variaciones de la topología de la red y retrasos en la misma, estos últimos son consecuencia de la no periodicidad al compartir información entre los robots. Los valores fusionados por el consenso son estables, sin embargo, no alcanzan el valor real, tienen un error de estado estable, se ha demostrado en [4], que es posible agregar un integrador al filtro de consenso para corregir este error, además el filtro se

vuelve robusto a las variaciones rápidas de los valores de la estimación y la varianza, como se observan en las Figura 4.5 y Figura 4.6

5.2 Trabajo a futuro

En este trabajo se enfocó el problema de la coordinación para la generación de mapa colectivo, es decir, fusionar la información individual en una global con mayores propiedades que las locales. Es decir, el enfoque fue crear una red robótica de información. Otro enfoque es de la coordinación de una red de robots para acciones, es decir un control colectivo, por ejemplo, mantener una formación de una cuadrilla o paralelizar tareas de manufactura en una fábrica, tanto para aumentar la producción o las características de algún producto.

Cuando los robots trabajan en ambiente en un ambiente sin el conocimiento del número de puntos de referencia, es decir, que el número de variables de estado cambiará conforme se explora el mundo. Para mantener estas variables almacenadas en los equipos de cómputo, es necesario plantear el problema asociado al manejo de memoria dinámica, esto afecta al determinismo temporal en la tarea de actualización del estado. En el caso del control cooperativo las variables de estado del sistema se incrementan debido a que se requiere controlar simultáneamente las poses, velocidades y fuerzas de cada robot. Por tanto, es importante considerar el manejo de memoria dinámica en tiempo real.

Desde el punto de vista de las comunicaciones en la coordinación de robots en tiempo real, un tema interesante a abordar es el de redes sin infraestructura, es decir, una red dinámica de robots [29]. Esto trae consigo distintos problemas asociados, como es la mayor cantidad de tráfico en la red, para conocer la topología de esta y, por tanto, es necesario considerar que cada nodo debe asignar tiempo de cómputo para cumplir con esta tarea, lo cual puede degradar el tiempo real del mismo. Por otro lado, los algoritmos de coordinación deben ser robustos al cambio en las variables de estado del sistema global.

A. Anexos

En los anexos siguientes se detalla el marco teórico para la localización y mapeo de cada robot utilizando el modelo cinemático de los robots, la planificación y control de estos, mediante potenciales artificiales.

A.A. Actualización de movimiento

Para definir el modelo cinemático del robot Qbot2 se utiliza la Figura A.1, en la cual se representa en sistema de referencia global $\{W\}$ y un sistema de referencia acoplado al i-ésimo robot $\{I\}$. Como se observa en la figura

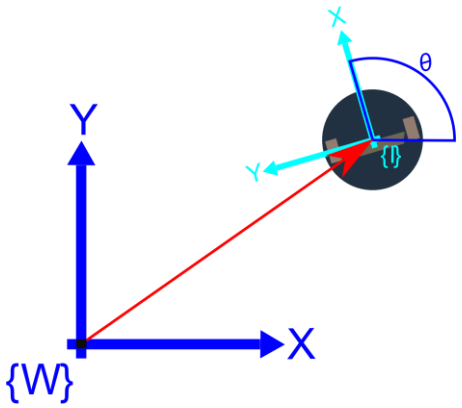

Figura A.1 Sistemas de referencia global $\{W\}$ y local del i-ésimo robot $\{I\}$

El estado de i-ésimo robot se compone de la pose del robot $\boldsymbol{r}_I(\ ^I k)$ y los puntos de referencia del que el robot va observando $\boldsymbol{m}(\ ^I k)$ hasta el momento $^I k$, la pose comprende la posición en X_I, Y_I y la orientación θ_I del robot. El mapa $\boldsymbol{m}_P(\ ^I k)$ son el conjunto de puntos de referencia P (X_P, Y_p) que el robot ha observado desde $^I k = 0$ hasta el instante $^I k$.

$$\boldsymbol{r}_I(\ ^I k) = \begin{bmatrix} X_I(\ ^I k) \\ Y_I(\ ^I k) \\ \theta_I(\ ^I k) \end{bmatrix}, \qquad \boldsymbol{m}(\ ^I k) = \begin{bmatrix} \vdots \\ X_P(\ ^I k) \\ Y_P(\ ^I k) \\ \vdots \end{bmatrix}$$

Así el estado del robot i-ésimo se define en (A.1)

$$\boldsymbol{x}_I(\ ^I k) = \begin{bmatrix} \boldsymbol{r}_I(\ ^I k) \\ \boldsymbol{m}(\ ^I k) \end{bmatrix} \tag{A.1}$$

El modelo por odometría, Figura A.2, es utilizado para expresar la función de transición de la pose del i-ésimo robot.

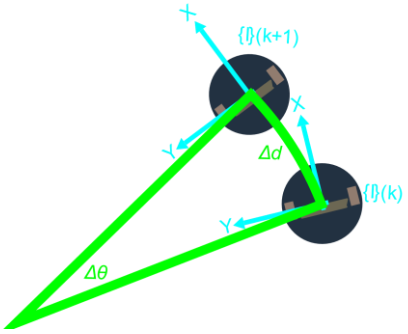

Figura A.2 Odometría para localización

$$\boldsymbol{r}_I(\ ^I k) = \boldsymbol{f}\left(\boldsymbol{r}_I(\ ^I k-1),\ ^I\boldsymbol{u}(\ ^I k-1)\right) \tag{A.2}$$

$$\boldsymbol{r}_I(\ ^I k) = \begin{bmatrix} X_I(\ ^I k-1) + \ ^I\Delta d(\ ^I k)\cos\left(\theta_I(\ ^I k-1)\right) \\ Y_I(\ ^I k-1) + \ ^I\Delta d(\ ^I k)\sin\left(\theta_I(\ ^I k-1)\right) \\ \theta_I(\ ^I k-1) + \ ^I\Delta\theta_I(\ ^I k) \end{bmatrix}$$

$$\boldsymbol{r}_I(\ ^I k) = \boldsymbol{r}_I(\ ^I k-1) + \boldsymbol{g}\left(\boldsymbol{r}_I(\ ^I k-1)\right)\ ^I\boldsymbol{u}(\ ^I k)$$

$$\boldsymbol{g}\left(\boldsymbol{r}_I(\ ^I k-1)\right) = \begin{bmatrix} \cos\left(\theta_I(\ ^I k-1)\right) & 0 \\ \sin\left(\theta_I(\ ^I k-1)\right) & 0 \\ 0 & 1 \end{bmatrix}$$

Dado quelos valores dela entrada $^I\boldsymbol{u}(k)$, $^I\Delta d(k)$ y $^I\Delta\theta(k)$ son suceptibles a errores $^I\delta d(k)$, $^I\delta\theta(k)$, por tanto, el modelo (A.2) discreto es el siguiente:

$$\boldsymbol{r}_I(\ ^I k) = \boldsymbol{f}\left(\boldsymbol{r}_I(\ ^I k-1),\ ^I\boldsymbol{u}(\ ^I k) + \ ^I\delta\boldsymbol{u}(\ ^I k)\right) \tag{A.3}$$

$$\boldsymbol{r}_I(\ ^I k) = \boldsymbol{r}_I(\ ^I k-1) + \boldsymbol{g}\left(\boldsymbol{r}_I(\ ^I k-1)\right)\{\ ^I\boldsymbol{u}(\ ^I k) + \ ^I\delta\boldsymbol{u}(\ ^I k)\}$$

En (A.3) $^I\boldsymbol{u}(\ ^I k) = [\ ^I\Delta d(\ ^I k),\ ^I\Delta\theta(\ ^I k)]^T$ y $^I\delta\boldsymbol{u}(\ ^I k) = [\ ^I\delta d(\ ^I k),\ ^I\delta\theta(\ ^I k)]^T$. También la determinación del mapa es susceptible a errores $\delta\boldsymbol{m}(\ ^I k)$, así, el vector de estados tendrá la siguiente forma

$$\boldsymbol{x}_I(\ {}^I k) = \begin{bmatrix} \boldsymbol{r}_I(\ {}^I k-1) + \boldsymbol{g}\left(\boldsymbol{r}_I(\ {}^I k-1)\right)\{\ {}^I\boldsymbol{u}(\ {}^I k) + \ {}^I\delta\boldsymbol{u}(\ {}^I k)\} \\ \boldsymbol{m}(\ {}^I k) + \quad \delta\boldsymbol{m}(\ {}^I k) \end{bmatrix} \tag{A.4}$$

De (A.4) se obtiene la estimación del vector de estado, para ello se utiliza las suposiciones del filtro de Kalman.

$$\hat{\boldsymbol{x}}_I(\ {}^I k|\ {}^I k-1) = \begin{bmatrix} \hat{\boldsymbol{r}}_I(\ {}^I k|\ {}^I k-1) \\ \widehat{\boldsymbol{m}}(\ {}^I k|\ {}^I k-1) \end{bmatrix} \tag{A.5}$$

$$\hat{\boldsymbol{x}}_I(\ {}^I k|\ {}^I k-1) = \begin{bmatrix} \hat{\boldsymbol{r}}_I(\ {}^I k-1|\ {}^I k-1) + \boldsymbol{g}\left(\hat{\boldsymbol{r}}_I(\ {}^I k-1|\ {}^I k-1)\right)\ {}^I\boldsymbol{u}(\ {}^I k) \\ \widehat{\boldsymbol{m}}(\ {}^I k-1|\ {}^I k-1) \end{bmatrix}$$

La matriz de covarianza, así como su predicción se define como

$$\boldsymbol{P}_{\mathrm{I}}(\ {}^I k|\ {}^I k) = E\left[\left(x_{\boldsymbol{I}}(\ {}^I k) - \hat{x}_I(\ {}^I k|\ {}^I k)\right)^2\right] \tag{A.6}$$

$$\boldsymbol{P}_I(\ {}^I k|\ {}^I k-1)$$

$$= \boldsymbol{x}(\ {}^I k)_{,\boldsymbol{x}(\ {}^I k-1)}\, \boldsymbol{P}_I(\ {}^I k-1|\ {}^I k-1)\, \boldsymbol{x}(\ {}^I k)^T{}_{,\boldsymbol{x}(\ {}^I k-1)}$$

$$+ \boldsymbol{x}(\ {}^I k)_{,\boldsymbol{u}(\ {}^I k-1)}\, \delta\boldsymbol{U}\, \boldsymbol{x}(\ {}^I k)^T{}_{,\boldsymbol{u}(\ {}^I k-1)}$$

Las matrices jacobianas en (A.6) se definen a manera de bloques en (A.7).

$$\boldsymbol{x}(\ {}^I k)_{,\boldsymbol{x}(\ {}^I k-1)} = \begin{bmatrix} \boldsymbol{r}(\ {}^I k)_{,\boldsymbol{r}(\ {}^I k-1)} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{m}(\ {}^I k)_{,\boldsymbol{m}(\ {}^I k-1)} \end{bmatrix} \tag{A.7}$$

$$\boldsymbol{x}(\ {}^I k)_{,\boldsymbol{u}(\ {}^I k)} = \begin{bmatrix} \boldsymbol{r}(\ {}^I k)_{,\boldsymbol{u}(\ {}^I k)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\boldsymbol{r}(\ {}^I k)_{,\boldsymbol{r}(\ {}^I k-1)} = \begin{bmatrix} 1 & 0 & -\ {}^I\Delta d(\ {}^I k)\sin\left(\theta_I(\ {}^I k-1)\right) \\ 0 & 1 & \ {}^I\Delta d(\ {}^I k)\cos\left(\theta_I(\ {}^I k-1)\right) \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{m}(\ {}^I k)_{,\boldsymbol{m}(\ {}^I k-1)} = \boldsymbol{I}$$

$$\boldsymbol{r}(\ {}^I k)_{,\boldsymbol{u}(\ {}^I k)} = \boldsymbol{g}\left(\hat{\boldsymbol{r}}_I(\ {}^I k-1|\ {}^I k-1)\right)$$

Por último se calcula matriz de información (A.8), la cual es el inverso de la matriz de covarianza, se obtiene a partir de las matrices jacobianas (A.7), junto la simetría de la matriz y la dispersión de la misma, permite realizar la actualización de los valores en tiempo lineal [39, p. 396].

$$\boldsymbol{Y}_I(\ {}^I k|\ {}^I k) = \{\boldsymbol{\Sigma}_I(\ {}^I k|\ {}^I k)\}^{-1}$$

$$\boldsymbol{Y}_I(\ {}^I k-1|\ {}^I k-1)=\begin{bmatrix}\boldsymbol{\alpha} & \boldsymbol{\beta}\\ \boldsymbol{\beta}^T & \boldsymbol{\gamma}\end{bmatrix}$$

$$\boldsymbol{\alpha}\ =\boldsymbol{Y}_I(\ {}^I k-1|\ {}^I k-1)[r][r]$$

$$\boldsymbol{\beta}=\boldsymbol{Y}_I(\ {}^I k-1|\ {}^I k-1)[r][m]$$

$$\boldsymbol{\gamma}=\boldsymbol{Y}_I(\ {}^I k-1|\ {}^I k-1)[m][m]$$

$$\boldsymbol{A}=\left\{\boldsymbol{r}(\ {}^I k)_{,\boldsymbol{r}(\ {}^I k-1)}\right\}^{-T}\boldsymbol{\alpha}\left\{\boldsymbol{r}(\ {}^I k)_{,\boldsymbol{r}(\ {}^I k-1)}\right\}^{-1}$$

$$\boldsymbol{B}=\left\{\left\{\left\{\boldsymbol{r}(\ {}^I k)_{,\boldsymbol{u}(\ {}^I k)}\right\}\delta\boldsymbol{U}\left\{\boldsymbol{r}(\ {}^I k)_{,\boldsymbol{u}(\ {}^I k)}\right\}^T\right\}^{-1}+\boldsymbol{A}\right\}^{-1}$$

$$\boldsymbol{C}=\left\{\boldsymbol{r}(\ {}^I k)_{,\boldsymbol{r}(\ {}^I k-1)}\right\}^{-T}\boldsymbol{\beta}$$

$$\boldsymbol{D}=\boldsymbol{\gamma}$$

$$\boldsymbol{Y}_I(\ {}^I k|\ {}^I k-1)=\begin{bmatrix}\boldsymbol{\alpha} & \boldsymbol{\beta}\\ \boldsymbol{\beta}^T & \boldsymbol{\gamma}\end{bmatrix}+\begin{bmatrix}\boldsymbol{A}-\boldsymbol{\alpha} & \boldsymbol{C}-\boldsymbol{\beta}\\ \boldsymbol{C}^T-\boldsymbol{\beta}^T & \boldsymbol{0}\end{bmatrix}-\begin{bmatrix}\boldsymbol{ABA} & \boldsymbol{ABC}\\ \boldsymbol{C}^T\boldsymbol{BA} & \boldsymbol{C}^T\boldsymbol{BC}\end{bmatrix}$$

$$\boldsymbol{Y}_I(\ {}^I k|\ {}^I k-1)=\boldsymbol{Y}_I(\ {}^I k-1|\ {}^I k-1)+\begin{bmatrix}\boldsymbol{A}-\boldsymbol{\alpha} & \boldsymbol{C}-\boldsymbol{\beta}\\ \boldsymbol{C}^T-\boldsymbol{\beta}^T & \boldsymbol{0}\end{bmatrix}-\begin{bmatrix}\boldsymbol{ABA} & \boldsymbol{ABC}\\ \boldsymbol{C}^T\boldsymbol{BA} & \boldsymbol{C}^T\boldsymbol{BC}\end{bmatrix} \tag{A.8}$$

El desarrollo del vector de información se presenta en las siguientes líneas

$$\widehat{\boldsymbol{y}}_I(\ {}^I k|\ {}^I k)=\boldsymbol{Y}_I(\ {}^I k|\ {}^I k)\widehat{\boldsymbol{x}}_I(\ {}^I k|\ {}^I k)$$

$$\widehat{\boldsymbol{y}}_I(\ {}^I k|\ {}^I k-1)=\boldsymbol{Y}_I(\ {}^I k|\ {}^I k-1)\widehat{\boldsymbol{x}}_I(\ {}^I k|\ {}^I k-1)$$

$$\widehat{\boldsymbol{y}}_I(\ {}^I k|\ {}^I k-1)$$

$$=\boldsymbol{Y}_I(\ {}^I k|\ {}^I k-1)\begin{bmatrix}\hat{\boldsymbol{r}}_I(\ {}^I k-1|\ {}^I k-1)+\boldsymbol{g}\left(\hat{\boldsymbol{r}}_I(\ {}^I k-1|\ {}^I k-1)\right)\ {}^I\boldsymbol{u}(\ {}^I k)\\ \widehat{\boldsymbol{m}}(\ {}^I k-1|\ {}^I k-1)\end{bmatrix}$$

$$\widehat{\boldsymbol{y}}_I(\ {}^I k|\ {}^I k-1)=\boldsymbol{Y}_I(\ {}^I k|\ {}^I k-1)\left\{\widehat{\boldsymbol{x}}_I(\ {}^I k-1|\ {}^I k-1)+\begin{bmatrix}\boldsymbol{g}\left(\hat{\boldsymbol{r}}_I(\ {}^I k-1|\ {}^I k-1)\right)\ {}^I\boldsymbol{u}(\ {}^I k)\\ \boldsymbol{0}\end{bmatrix}\right\}$$

Utilizando el resultado de (A.8) en la ecuación anterior se tiene

$$\widehat{\boldsymbol{y}}_I(\ {}^I k|\ {}^I k-1)=\boldsymbol{Y}_I(\ {}^I k-1|\ {}^I k-1)\widehat{\boldsymbol{x}}_I(\ {}^I k-1|\ {}^I k-1)$$

$$+\left\{\begin{bmatrix}\boldsymbol{A}-\boldsymbol{\alpha} & \boldsymbol{C}-\boldsymbol{\beta}\\ \boldsymbol{C}^T-\boldsymbol{\beta}^T & \boldsymbol{0}\end{bmatrix}-\begin{bmatrix}\boldsymbol{ABA} & \boldsymbol{ABC}\\ \boldsymbol{C}^T\boldsymbol{BA} & \boldsymbol{C}^T\boldsymbol{BC}\end{bmatrix}\right\}\widehat{\boldsymbol{x}}_I(\ {}^I k-1|\ {}^I k-1)$$

$$+\boldsymbol{Y}_I(\ {}^I k|\ {}^I k-1)\begin{bmatrix}\boldsymbol{g}\left(\hat{\boldsymbol{r}}_I(\ {}^I k-1|\ {}^I k-1)\right)\ {}^I\boldsymbol{u}(\ {}^I k)\\ \boldsymbol{0}\end{bmatrix}$$

$$\hat{\boldsymbol{y}}_I({}^I k|{}^I k-1)=\hat{\boldsymbol{y}}_I({}^I k-1|{}^I k-1) \tag{A.9}$$
$$+\left\{\begin{bmatrix}\boldsymbol{A}-\boldsymbol{\alpha} & \boldsymbol{C}-\boldsymbol{\beta}\\ \boldsymbol{C}^T-\boldsymbol{\beta}^T & \boldsymbol{0}\end{bmatrix}-\begin{bmatrix}\boldsymbol{ABA} & \boldsymbol{ABC}\\ \boldsymbol{C}^T\boldsymbol{BA} & \boldsymbol{C}^T\boldsymbol{BC}\end{bmatrix}\right\}\hat{\boldsymbol{x}}_I({}^I k-1|{}^I k-1)$$
$$+\begin{bmatrix}\boldsymbol{Ag}\left(\hat{\boldsymbol{r}}_I({}^I k-1|{}^I k-1)\right){}^I\boldsymbol{u}({}^I k)\\ \boldsymbol{C}^T\boldsymbol{g}\left(\hat{\boldsymbol{r}}_I({}^I k-1|{}^I k-1)\right){}^I\boldsymbol{u}({}^I k)\end{bmatrix}$$
$$+\begin{bmatrix}\boldsymbol{ABAg}\left(\hat{\boldsymbol{r}}_I({}^I k-1|{}^I k-1)\right){}^I\boldsymbol{u}({}^I k)\\ \boldsymbol{C}^T\boldsymbol{BAg}\left(\hat{\boldsymbol{r}}_I({}^I k-1|{}^I k-1)\right){}^I\boldsymbol{u}({}^I k)\end{bmatrix},$$

integrando las ecuaciones anteriores se obtiene el Algoritmo A.1, para la actualización de movimiento.

$movimiento\ (\mathbf{Y}(k-1\|k-1),\hat{\boldsymbol{y}}(k-1\|k-1),\hat{\boldsymbol{x}}(k-1\|k-1),\boldsymbol{u}(k))$		
1	$\boldsymbol{r}(k)_{,\boldsymbol{r}(k)}=\begin{bmatrix}1 & 0 & -{}^I\Delta d(k)\sin(\theta_I(k-1))\\ 0 & 1 & {}^I\Delta d(k)\cos(\theta_I(k-1))\\ 0 & 0 & 1\end{bmatrix}$ $\boldsymbol{r}(k)_{,\boldsymbol{u}(k)}=\boldsymbol{g}(\hat{\boldsymbol{r}}_I(k-1\|k-1))$ $\boldsymbol{\alpha}=\mathbf{Y}(k-1\|k-1)[\boldsymbol{r}][\boldsymbol{r}]$ $\boldsymbol{\beta}=\mathbf{Y}(k-1\|k-1)[\boldsymbol{r}][\boldsymbol{m}]\rightarrow\mathbf{Y}(k-1\|k-1)[\boldsymbol{r}][\boldsymbol{m}^+]$ $\boldsymbol{\gamma}=\mathbf{Y}(k-1\|k-1)[\boldsymbol{m}][\boldsymbol{m}]$ $\boldsymbol{A}=\{\boldsymbol{r}(k)_{,\boldsymbol{r}(k)}\}^{-T}\boldsymbol{\alpha}\{\boldsymbol{r}(k)_{,\boldsymbol{r}(k)}\}^{-1}$ $\boldsymbol{B}=\left\{\{\boldsymbol{r}(k)_{,\boldsymbol{u}(k)}\}\delta\boldsymbol{U}\{\boldsymbol{r}(k)_{,\boldsymbol{u}(k)}\}^T\right\}^{-1}+\boldsymbol{A}\Big\}^{-1}$ $\boldsymbol{C}=\{\boldsymbol{r}(k)_{,\boldsymbol{r}(k)}\}^{-T}\boldsymbol{\beta}$ $\boldsymbol{D}=\boldsymbol{\gamma}$	
2	$\mathbf{Y}(k\|k-1)=\mathbf{Y}(k-1\|k-1)+\begin{bmatrix}\boldsymbol{A}-\boldsymbol{\alpha} & \boldsymbol{C}-\boldsymbol{\beta}\\ \boldsymbol{C}^T-\boldsymbol{\beta}^T & \boldsymbol{0}\end{bmatrix}-\begin{bmatrix}\boldsymbol{ABA} & \boldsymbol{ABC}\\ \boldsymbol{C}^T\boldsymbol{BA} & \boldsymbol{C}^T\boldsymbol{BC}\end{bmatrix}$	(A.8)
3	$\hat{\boldsymbol{y}}(k-1\|k-1)=\hat{\boldsymbol{y}}(k-1\|k-1)+\left\{\begin{bmatrix}\boldsymbol{A}-\boldsymbol{\alpha} & \boldsymbol{C}-\boldsymbol{\beta}\\ \boldsymbol{C}^T-\boldsymbol{\beta}^T & \boldsymbol{0}\end{bmatrix}-\begin{bmatrix}\boldsymbol{ABA} & \boldsymbol{ABC}\\ \boldsymbol{C}^T\boldsymbol{BA} & \boldsymbol{C}^T\boldsymbol{BC}\end{bmatrix}\right\}\hat{\boldsymbol{x}}(k-1\|k-1)+\begin{bmatrix}\boldsymbol{Ag}(\hat{\boldsymbol{r}}_I(k-1\|k-1))\boldsymbol{u}(k)\\ \boldsymbol{C}^T\boldsymbol{g}(\hat{\boldsymbol{r}}_I(k-1\|k-1))\boldsymbol{u}(k)\end{bmatrix}+\begin{bmatrix}\boldsymbol{ABAg}(\hat{\boldsymbol{r}}_I(k-1\|k-1)){}^I\boldsymbol{u}({}^I k)\\ \boldsymbol{C}^T\boldsymbol{BAg}(\hat{\boldsymbol{r}}_I(k-1\|k-1)){}^I\boldsymbol{u}({}^I k)\end{bmatrix}$	(A.9)
4	$\hat{\boldsymbol{x}}(k\|k-1)=\begin{bmatrix}\hat{\boldsymbol{r}}_I(k-1\|k-1)+\boldsymbol{g}(\hat{\boldsymbol{r}}_I(k-1\|k-1)){}^I\boldsymbol{u}({}^I k)\\ \hat{\boldsymbol{m}}(k-1\|k-1)\end{bmatrix}$	(A.5)
$\mathrm{return}([\boldsymbol{Y}(k\|k-1),\hat{\boldsymbol{y}}(k\|k-1),\hat{\boldsymbol{x}}(k\|k-1)])$		

Algoritmo A.1

A.B. Actualización de observación

El ambiente que recorren los robots son pasillos de oficina, los puntos de referencia a detectar son la esquinas tanto los pasillos como de los interiores de las oficinas. Los obstáculos, mostrados en el centro del pasillo por dos rectángulos grises, también tienen esquinas en ángulos rectos como se muestra en Figura A.3.

Figura A.3 Corredor

Es necesario que los robots tengan la capacidad de detectar a otros robots, cuando un robot detecta a otros robots, este solicita que los demás envíen su posición para corroborar la detección y de este modo no agregarlos al mapa.

$$ {}^{I}\boldsymbol{g}_m\left(\boldsymbol{x}_I({}^{I}k),\ {}^{I}\boldsymbol{x}_m({}^{I}k)\right) = \begin{bmatrix} X_I({}^{I}k) + {}^{\mathrm{I}}r_m\cos\left(\theta_I(k) + {}^{\mathrm{I}}\alpha_m\right) \\ Y_I({}^{I}k) + {}^{\mathrm{I}}r_m\sin\left(\theta_I(k) + {}^{\mathrm{I}}\alpha_m\right) \end{bmatrix} $$

El modelo de la observación para los puntos de referencia es (A.10)

$$ {}^{I}\boldsymbol{z}_P({}^{I}k) = \boldsymbol{h}\left(\boldsymbol{r}_I({}^{I}k), \boldsymbol{m}_P\,({}^{I}k)\right) \tag{A.10} $$

$$ {}^{I}\boldsymbol{z}_P({}^{I}k) = \begin{bmatrix} {}^{\mathrm{I}}\rho_P({}^{I}k) \\ {}^{\mathrm{I}}\alpha_P({}^{I}k) \end{bmatrix} $$

$$ \Delta X_P({}^{I}k) = X_P({}^{I}k) - X_I({}^{I}k), \qquad \Delta Y_P({}^{I}k) = Y_P({}^{I}k) - Y_I({}^{I}k) $$

$$ {}^{\mathrm{I}}\rho_P({}^{I}k) = \left(\left(\Delta X_P({}^{I}k)\right)^2 + \left(\Delta Y_P({}^{I}k)\right)^2\right)^{\mathbf{1/2}} $$

$$ {}^{\mathrm{I}}\alpha_P({}^{I}k) = \operatorname{atan}\left(\frac{\Delta X_P({}^{I}k)}{\Delta Y_P({}^{I}k)}\right) - \theta_I $$

Del mismo modo que en el modelo de movimiento, el modelo de observación es susceptible a errores (A.10)

$$ {}^{I}\boldsymbol{z}_P({}^{I}k) = \boldsymbol{h}\left(\boldsymbol{r}_I({}^{I}k), \boldsymbol{m}_P\,({}^{I}k)\right) + {}^{I}\delta\boldsymbol{z}_P({}^{I}k) \tag{A.11} $$

En (A.10) ${}^{I}\delta\boldsymbol{z}_P({}^{I}k) = \left[{}^{I}\delta r_P({}^{I}k),\ {}^{I}\delta\alpha_P({}^{I}k)\right]^T$. La matriz de covarianza de error de observación de punto de referencia P del robot $\{I\}$ se muestra en

$$ {}^{I}\delta \boldsymbol{Z}({}^{I}k) = E\left[\left({}^{I}\delta \boldsymbol{z}_m({}^{I}k)\right)^2\right] \tag{A.12} $$

La matriz jacobiana de observación (A.13), se utiliza para la actualización del vector de información y matriz de información

$$ \boldsymbol{{}^{I}z}_P({}^{I}k)_{,\boldsymbol{x}({}^{I}k-1)} \tag{A.13} $$
$$ = \begin{bmatrix} -\frac{\Delta X_P({}^{I}k)}{{}^{\mathrm{I}}\rho_P({}^{I}k)} & -\frac{\Delta Y_P({}^{I}k)}{{}^{\mathrm{I}}\rho_P({}^{I}k)} & 0 & \cdots & \frac{\Delta X_P({}^{I}k)}{{}^{\mathrm{I}}\rho_P({}^{I}k)} & \frac{\Delta Y_P({}^{I}k)}{{}^{\mathrm{I}}\rho_P({}^{I}k)} & \cdots \\ \frac{\Delta Y_P({}^{I}k)}{\left({}^{\mathrm{I}}\rho_P({}^{I}k)\right)^2} & -\frac{\Delta X_P({}^{I}k)}{\left({}^{\mathrm{I}}\rho_P({}^{I}k)\right)^2} & -1 & \cdots & -\frac{\Delta Y_P({}^{I}k)}{\left({}^{\mathrm{I}}\rho_P({}^{I}k)\right)^2} & \frac{\Delta X_P({}^{I}k)}{\left({}^{\mathrm{I}}\rho_P({}^{I}k)\right)^2} & \cdots \end{bmatrix} $$

$$ \begin{aligned} \hat{\boldsymbol{y}}({}^{I}k|\,{}^{I}k) &= \hat{\boldsymbol{y}}({}^{I}k|\,{}^{I}k-1) \\ &\quad + \sum_P \left\{ \boldsymbol{{}^{I}z}_P({}^{I}k)_{,\boldsymbol{x}({}^{I}k-1)} \right\}^T \left\{ {}^{I}\delta \boldsymbol{Z}({}^{I}k) \right\}^{-1} \left\{ \boldsymbol{{}^{I}z}_P({}^{I}k) \right. \\ &\quad \left. - \boldsymbol{h}\left(\hat{\boldsymbol{r}}_I({}^{I}k|\,{}^{I}k-1), \hat{\boldsymbol{m}}_P({}^{I}k|\,{}^{I}k-1)\right) \right\} \end{aligned} \tag{A.14} $$

$$ \begin{aligned} \boldsymbol{Y}({}^{I}k|\,{}^{I}k) &= \boldsymbol{Y}({}^{I}k|\,{}^{I}k-1) \\ &\quad + \sum_P \left\{ \boldsymbol{{}^{I}z}_P({}^{I}k)_{,\boldsymbol{x}({}^{I}k-1)} \right\}^T \left\{ {}^{I}\delta \boldsymbol{Z}({}^{I}k) \right\}^{-1} \left\{ \boldsymbol{{}^{I}z}_P({}^{I}k)_{,\boldsymbol{x}({}^{I}k-1)} \right\} \end{aligned} $$

De este modo integrando las ecuaciones anteriores obtenemos el Algoritmo A.2, este es utilizado para la actualización por observación.

$[\mathbf{Y}(k|k), \hat{\boldsymbol{y}}(k|k), \hat{\boldsymbol{x}}(k|k)] = observación(\mathbf{Y}(k|k-1), \hat{\boldsymbol{y}}(k|k-1), \hat{\boldsymbol{x}}(k|k-1), \boldsymbol{z}(k));$

1	$\boldsymbol{A} = \begin{bmatrix} \frac{\Delta X_P(k)}{{}^{\mathrm{I}}\rho_P(k)} & \frac{\Delta Y_P(k)}{{}^{\mathrm{I}}\rho_P(k)} \\ -\frac{\Delta Y_P(k)}{\left({}^{\mathrm{I}}\rho_P(k)\right)^2} & \frac{\Delta X_P(k)}{\left({}^{\mathrm{I}}\rho_P(k)\right)^2} \end{bmatrix}; \quad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ & \boldsymbol{-1} \end{bmatrix}$	
2	**for each** $\boldsymbol{m_p} \boldsymbol{\epsilon}\, \boldsymbol{m}^+$	
3	$\boldsymbol{i}[\boldsymbol{r}] = \boldsymbol{i}[\boldsymbol{r}] + \{\boldsymbol{B^T}\}\left\{ {}^{I}\delta \boldsymbol{Z}(k) \right\}^{-1} \left\{ {}^{I}\boldsymbol{z}_P(k) - \boldsymbol{h}\left(\hat{\boldsymbol{r}}_I(k|k-1), \hat{\boldsymbol{m}}_P(k|k-1)\right) \right\}$	
4	$\boldsymbol{i}[\boldsymbol{m}] = \boldsymbol{i}[\boldsymbol{m}] + \{\boldsymbol{A}\}\left\{ {}^{I}\delta \boldsymbol{Z}(k) \right\}^{-1} \left\{ {}^{I}\boldsymbol{z}_P(k) - \boldsymbol{h}\left(\hat{\boldsymbol{r}}_I(k|k-1), \hat{\boldsymbol{m}}_P(k|k-1)\right) \right\}$	
5	$\boldsymbol{I}[\boldsymbol{r}][\boldsymbol{r}] = \boldsymbol{I}[\boldsymbol{r}][\boldsymbol{r}] + \{\boldsymbol{B^T}\}\left\{ {}^{I}\delta \boldsymbol{Z}(k) \right\}^{-1} \{\boldsymbol{B}\}$	
6	$\boldsymbol{I}[\boldsymbol{r}][\boldsymbol{m}] = \boldsymbol{I}[\boldsymbol{r}][\boldsymbol{m}] + \{\boldsymbol{B^T}\}\left\{ {}^{I}\delta \boldsymbol{Z}(k) \right\}^{-1} \{\boldsymbol{A}\}$	
7	$\boldsymbol{I}[\boldsymbol{m}][\boldsymbol{m}] = \boldsymbol{I}[\boldsymbol{m}][\boldsymbol{m}] + \{\boldsymbol{A}\}\left\{ {}^{I}\delta \boldsymbol{Z}(k) \right\}^{-1} \{\boldsymbol{A}\}$	
8	**end**	
9	$\hat{\boldsymbol{y}}(k|k) = \hat{\boldsymbol{y}}(k|k-1) + \begin{bmatrix} \boldsymbol{i}[\boldsymbol{r}] \\ \boldsymbol{i}[\boldsymbol{m}] \end{bmatrix}$	

10	$\boldsymbol{Y}(k\|k) = \boldsymbol{Y}(k\|k-1) + \begin{bmatrix} \boldsymbol{I}[\boldsymbol{r}][r] & \boldsymbol{I}[\boldsymbol{r}][\boldsymbol{m}] \\ \boldsymbol{I}^{\boldsymbol{T}}[\boldsymbol{r}][m] & \boldsymbol{I}[\boldsymbol{m}][\boldsymbol{m}] \end{bmatrix}$	
return$\left(\mathbf{Y}(k\|k), \hat{\boldsymbol{y}}(k\|k), \hat{\boldsymbol{x}}(k\|k)\right)$		

Algoritmo A.2

A.C. Planificación de movimiento: potenciales artificiales

Una de las virtudes de utilizar potenciales artificiales, es realizar la planeación de rutas tanto de robots móviles como manipuladores, en tiempo real [40]. Sin embargo, esta técnica es susceptible errores; por ejemplo, cuando se presentan mínimos locales, es decir, el robot no llega a destino, debido a que se queda atorado en el mínimo local. Para solucionar este problema se pueden emplear potenciales armónicos, ya que solo contienen un solo mínimo global [41].

En la planificación de trayectorias por potenciales artificiales, se puede asociar el gradiente del potencial a la velocidad del robot, es como si el robot fuese afectado por una fuerza de cuerpo virtual [42, pp. 550, 551, 2]. Esta técnica de planificación se representa en (A.15).

$$\boldsymbol{u}_I = T\,\boldsymbol{\nabla}\,{}^{I}\phi\left(X_I({}^{I}k), Y_I({}^{I}k)\right) = \Delta t \begin{bmatrix} {}^{I}v \\ {}^{I}\omega \end{bmatrix} \tag{A.15}$$

La dimensión del gradiente $\boldsymbol{\nabla}\,{}^{I}\phi$ es dos, ya que solo depende de la posición del robot y no de la orientación, sin embargo, se puede relacionar el desplazamiento creado por el gradiente $\left\|\boldsymbol{\nabla}\,{}^{I}\phi\right\|$ y el cambio de orientación, mediante la curvatura ${}^{I}\kappa$ [30, p. 46].

$${}^{I}\kappa\;{}^{I}v = {}^{I}\omega \tag{A.16}$$

También es posible calcular ${}^{I}\kappa$, utilizando derivadas parciales de los potenciales artificiales [43, pp. 35-43]. El potencial ${}^{I}\phi$ se puede descomponer en evasión de obstáculos y cumplimiento de objetivos.

$${}^{I}\phi = {}^{I}\phi_{source} + {}^{I}\phi_{sink} + {}^{I}\phi_{robot} + \Phi \tag{A.17}$$

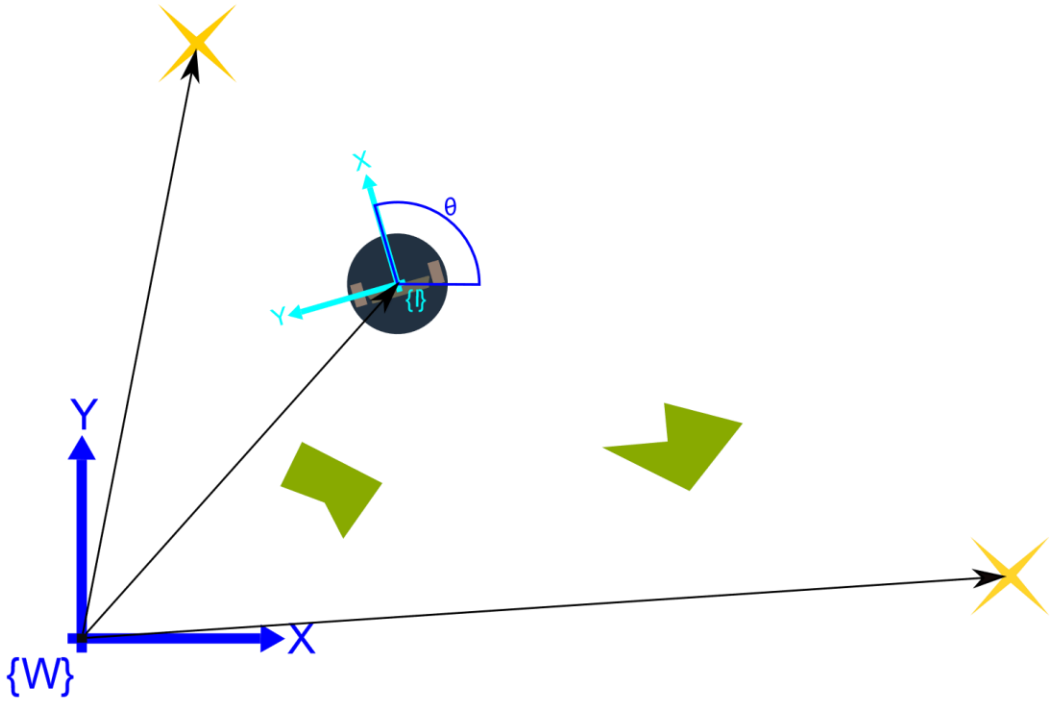

Figura A.4 Planificación y control

Los potenciales para el cumplimiento de los objetivos se tiene un potencial de origen ${}^{I}\phi_{source}$ y otro de destino ${}^{I}\phi_{sink}$, ambos se pueden representar como flujo de fuente y sumidero respectivamente (ver Figura A.5). La fuente es un máximo global, mientras que el sumidero es un mínimo global para el robot.

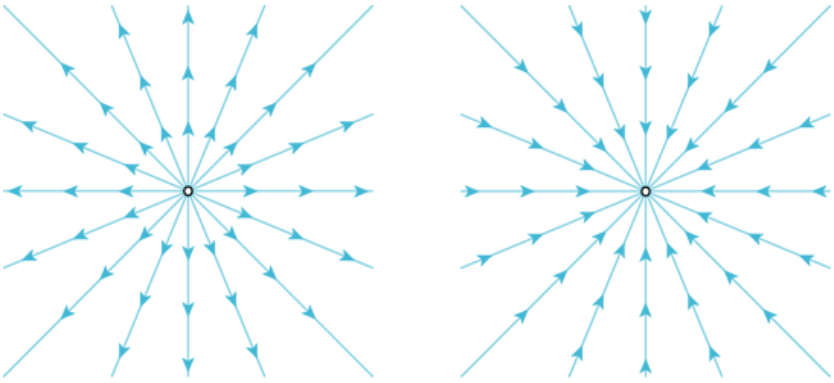

Figura A.5 Flujo de fuente y sumidero (Basado en [44])

Los potenciales para evasión de obstáculos se dividen en dos, el primero es para evitar colisionar con otros robots ${}^{I}\phi_{robot}$, el cual se modela como un flujo doblete (Figura A.6).

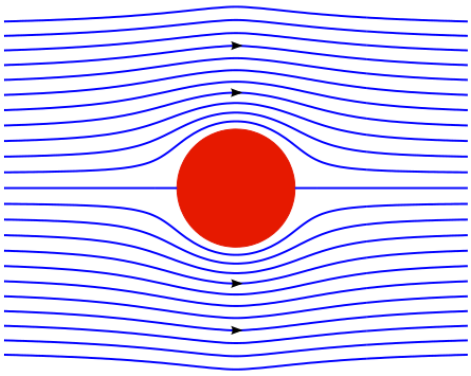

Figura A.6 Flujo alrededor para evitar la colisión con otro robot (Basado en [44])

El segundo potencial para evitar colisiones es consecuencia de los puntos del mapa ϕ, estos los puntos son las esquinas del pasillo y de los obstáculos. Es posible modelar los efectos de las esquinas con el flujo irrotacional de pared (A.18), en esta ecuación n es el valor de la potencia del flujo de pared [44]. En la Figura A.7 se observan como cambia de dirección el flujo debida a la presencia de estos elementos.

$$\phi = \sum Ar^n \cos n\alpha \tag{A.18}$$

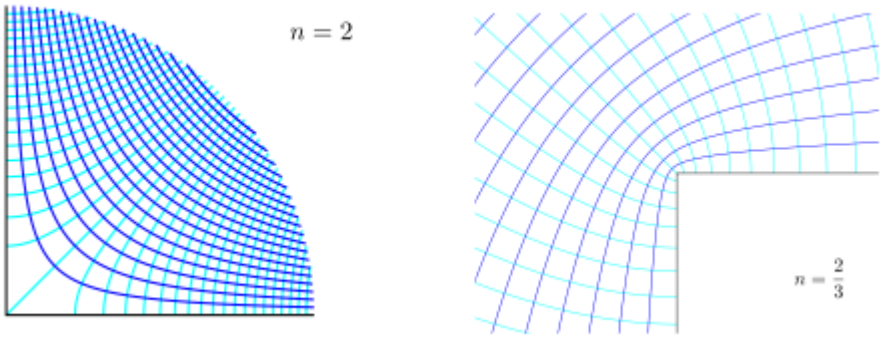

Figura A.7 Flujo debido a una esquina en el pasillo o de un obstáculo (Basado en [44])

Referencias

[1] S. Thurn, «Robot Mapping: A Survey,» de *Exploring Artificial Intelligence in the New Millenium*, Pittsburgh, Morgan Kaufmann, 2002, p. 31.

[2] F. Conte, A. Cristofaro, A. Renzaglia y A. Martinelli, «Cooperative Localization and SLAM Based on the Extended Information Filter,» de *Multi-Robot Systems, Trends and Development*, Londres, InTech, 2011, p. 21.

[3] Martin BuehlerKarl IagnemmaSanjiv Singh, The 2005 DARPA Grand Challenge, Alemania: Springer, 2007.

[4] R. Aragues, C. Sagues y Y. Mezouar, Parallel and Distributed Map Merging and Localization, New York: Springer, 2015.

[5] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani y H. Durrant-Whyte, «Simultaneous Localization and Mapping With Sparse Extended Information Filters.,» *International Journal of Robotics Research,* vol. 23, nº 7/8, pp. 693-716, 2004.

[6] R. Eustice, M. Walter y J. Leonard, «Sparse extended information filters: insights into sparsification,» de *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* , Edmonton, Alta., Canada , 2005 .

[7] S. Thrun y Y. Liu, «Multi-Robot SLAM With Sparse Extended Information Filers,» de *The Eleventh International Symposium*, vol. The Eleventh International Symposium, Heildelberg, Springer, 2005, pp. 254-266.

[8] D. Weyns, Architecture-Based Design of Multi-Agent Systems, Berlín: Springer-Verlag, 2010.

[9] J. Lunze, Control theory of digitally networked dynamic systems, New York: Springer, 2014.

[10] Y. C. Wie Ren, Distributed Coordination of Multi-agent Networks: Emergent Problems, Models, and Issues, New York: Springer, 2010.

[11] S. J. Russell y P. Norvig, Artificial intelligence : a modern approach, Upper Saddle River, New Jersey: Prentice Hall, 2010.

[12] J. Liu y J. Wu, Multi-agent robotic systems, Boca Raton, Florida: CRC Press, 2001.

[13] S. Kernbach, Handbook of collective robotics fundamentals and challenges, Boca Raton, Florida: CRC Press, 2013.

[14] M. S. Mahmoud y Y. Xia, Networked filtering and fusion in wireless sensor networks, Boca Raton, Florida: CRC Press, 2015.

[15] H. Mitchell, Data fusion: concepts and ideas, Berlín: Springer, 2012.

[16] S. S. Iyengar y R. R. Brooks, « Estimation and Kalman Filters,» de *Distributed sensor networks* , Boca Raton, Florida, Chapman & Hall, 2005, p. 26.

[17] J. V. Candy, Model-based signal processing, Hoboken, New Jersey: IEEE Press : Wiley-Interscience, 2006.

[18] J. R. Raol, Multi-Sensor Data Fusion with MATLAB, Boca Raton, Florida: CRC Press, 2009.

[19] L. Lavagno y R. Zurawski, «Embedded Systems: Toward Networking of Embedded Systems,» de *Embedded Systems Handbook* , Boca Raton, Florida, CRC Press, 2005, p. 17.

[20] H. Kopetz, Real-Time Systems Design priciples for Distributed Embedded Aplications, berlín: Springer, 2011.

[21] G. C. Buttazzo, «Real-Time Schedulling and Resource Management,» de *Handbook of Real-Time and Embedded Systems*, Boca Raton, Florida, Chapman & Hall, 2008, p. 14.

[22] G. Buttazzo, Hard Real-Time Computing Systems, New York: Springer, 2011.

[23] H. Hansson, M. Nolin y T. Nolte, «Real-Time in Embedded Systems,» de *Embedded Systems Handbook*, Boca Ratom, Florida, CRC Press, 2005, p. 31.

[24] K. Yaghmour, J. Masters, G. Ben-Yossef y P. Gerum, Building Embedded Linux Systems, Beijing : O'Reilly, 2008.

[25] D. E. Comer, Computer Networks and Internets, Upper Saddle River, New Jersey: Pearson Education, 2009.

[26] V. R. Syrotiuk, «Medium Access Control (MAC) protocols for Wireless Networks,» de *Handbook of algorithms for wireless networking and mobile computing* , Boca Raton, Florida, Champan & Hall, 2006, p. 41.

[27] V. Subramonian y C. Gill, «Middleware Design and Implementation for Networked Embedded Systems,» de *Embedded systems handbook* , Boca Raton, Florida, CRC Press, 2005, p. 14.

[28] X. S. Zhou y S. I. Roumeliotis, «Multi-robot SLAM with Unknown Initial Correspondence: The Robot Rendezvous Case,» de *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* , Beijing, China, 2006.

[29] T. D. B. H. H. T. L. Keith Y. K. Leung, «Decentralized Localization of Sparsely-Communicating Robot Networks: A Centralized-Equivalent Approach,» *IEEE Transactions on Robotics,* vol. 26, nº 1, pp. 62-77, 2010.

[30] S. G. Tzafestas, Introduction to Mobile Robot Control, London, UK: Elsevier , 2014.

[31] R.Olfati-Saber y R.M.Murray, «Consensus problems in networks of agents with switching topolgy and time delays,» *IEEE TRANSACTIONS ON AUTOMATIC CONTROL,* vol. 49, nº 9, pp. 1520-1533, 2004.

[32] M. Paskin, «Thin Junction Tree Filters for Simultaneous Localization and Mapping,» de *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, San Francisco, CA, 2003.

[33] C. Bishop, Pattern Recognition and Machine Learning, Heidelberg: Springer, 2006.

[34] G. G.Rigatos, Modelling and Control for Intelligent Industrial Systems Adaptive Algorithms in Robotics and Industrial Engineering, Berlin: Springer-Verlag, 2011.

[35] M. Mesbahi y M. Egerstedt, Graph Theoretic Methods in Multiagent Networks, New Jersey: Princeton University Press, 2010.

[36] R. Olfati-Saber, J. A. Fax y R. M. Murray, «Consensus and Cooperation in Networked Multi-Agent Systems,» *Proceedings of the IEEE,* vol. 95, nº 1, pp. 215 - 233 , 2007 .

[37] K. Ogata, Modern Control Engineering, Boston : Pearson, 2010 .

[38] V. Nguyen, A. Martinelli, N. Tomatis y R. Siegwart, «A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics,» de *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* , Edmonton, Alta., Canada , 2005.

[39] S. Thrun, W. Burgard y D. Fox, Probabilistics robotics, Cambridge, Massachusetts: The MIT Press, 2006.

[40] O. Khatib, «Real-Time Obstacle Avoidance for Manipulators and Mobile Robots,» *The International Journal of Robotics Reaserch,* vol. 5, nº 1, p. 9, 1986.

[41] R. M. M. Stephen Waydo, «Vehicle Motion Planning Using Stream Functions,» de *International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.

[42] B. Siciliano, L. Sciavicco, L. Villani y G. Oriolo, Robotics Modelling, Planning and Control, Londres, RU: Springer-Verlag, 2009.

[43] A. Pressley, Elementary Differential Geometry, Londres, RU: Springer-Verlag, 2012.

[44] R. K. Kundu y I. M. Cohen, Fluid Mechanics, San Diego, EUA: Academic Press, 2002.