



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Algoritmo de minimización de  
energía en redes IoT utilizando  
el espacio dual**

**TESIS**

Que para obtener el título de  
**Ingeniera en Telecomunicaciones**

**P R E S E N T A**

Gilde Valeria Rodríguez Jiménez

**DIRECTOR DE TESIS**

Dr. Luis Francisco García Jiménez



Ciudad Universitaria, Cd. Mx., 2020



# **Agradecimientos**

Agradezco el apoyo brindado para realizar este trabajo de tesis a Dr. Luis Francisco García Jiménez. Por otro lado agradezco el apoyo brindado por parte del proyecto DGAPA-PAPIIT IA105520.

Muchas gracias por apoyarme siempre: Guadalupe, Atala, Verónica y Victoria.  
Muchas gracias a cada una de las mujeres en mi vida.



# Índice general

<b>Resumen</b> . . . . .	<b>1</b>
<b>1. Introducción</b> . . . . .	<b>3</b>
1.1. Definición de problema . . . . .	4
1.2. Hipótesis . . . . .	4
1.3. Aproximación al problema . . . . .	4
1.4. Metas . . . . .	4
1.4.1. Meta general . . . . .	4
1.4.2. Metas particulares . . . . .	4
1.5. Metodología . . . . .	5
1.6. Contribución . . . . .	5
1.7. Descripción del contenido . . . . .	5
<b>2. Antecedentes</b> . . . . .	<b>7</b>
2.1. Estado del Arte . . . . .	7
<b>3. Espacio dual</b> . . . . .	<b>9</b>
3.1. Transformación al espacio dual . . . . .	9
3.2. Propiedades del espacio dual . . . . .	10
3.2.1. Intersección en el espacio dual. . . . .	10
3.2.2. Áreas en el espacio dual . . . . .	11
3.2.3. Cierre convexo . . . . .	12
3.2.4. Ángulo de intersección de dos líneas en el espacio dual. . . . .	13
3.3. Punto en movimiento sobre una línea . . . . .	14
<b>4. Desarrollo</b> . . . . .	<b>17</b>
4.1. Dos sensores y un objeto móvil . . . . .	17
4.2. Algoritmo . . . . .	21
4.3. Simulación . . . . .	23
4.3.1. Ejemplo 1 . . . . .	23
4.3.2. Ejemplo 2 . . . . .	26
<b>5. Conclusiones</b> . . . . .	<b>29</b>
5.1. Conclusiones generales . . . . .	29
5.2. Verificación de la hipótesis . . . . .	30
5.3. Perspectivas de investigación . . . . .	30



# Índice de figuras

3.1. Transformación de un conjunto de puntos del espacio primal a un conjunto de líneas al espacio dual. . . . .	10
3.2. Transformación de dos puntos del espacio primal al espacio dual ( <i>moño</i> ). . . . .	11
3.3. Puntos colineales. . . . .	11
3.4. Áreas infinitas y delimitadas. . . . .	12
3.5. Cierre convexo en el espacio primal y en el espacio dual. . . . .	13
3.6. Área delimitada con las intersecciones del cierre convexo. . . . .	13
3.7. Cinco puntos y una línea en tiempo $t$ . . . . .	14
3.8. Cinco puntos y una línea en tiempo $t + 1$ . . . . .	15
4.1. Dos sensores y un objeto móvil: pendiente positiva. . . . .	18
4.2. Dos sensores y un objeto móvil: pendiente negativa. . . . .	19
4.3. Dos sensores y un objeto móvil: cuatro líneas $l_i$ . . . . .	19
4.4. Cuatro sensores y un objeto móvil: moños consecutivos. . . . .	20
4.5. Cuatro sensores y un objeto móvil: cambio de pendiente. . . . .	21
4.6. Ejemplo 1: Simulación del algoritmo en tiempo $t$ . . . . .	23
4.7. Ejemplo 1: Simulación del algoritmo en tiempo $t + 1$ . . . . .	24
4.8. Ejemplo 1: Simulación del algoritmo en tiempo $t + 2$ . . . . .	25
4.9. Ejemplo 1: Simulación del algoritmo en tiempo $t + 3$ . . . . .	25
4.10. Ejemplo 2: Simulación del algoritmo en tiempo $t$ . . . . .	26
4.11. Ejemplo 2: Simulación del algoritmo en tiempo $t + 1$ . . . . .	27
4.12. Ejemplo 2: Simulación del algoritmo en tiempo $t + 2$ . . . . .	27

# Resumen

Internet de las cosas (IoT) es un paradigma crucial en este momento histórico y su crecimiento global será determinante en el futuro próximo. El objetivo principal de IoT es conectar objetos (*cosas*) físicos o virtuales con Internet para optimizar tareas de la vida cotidiana.

El seguimiento de objetos móviles (OTSN) juega una tarea importante en muchas aplicaciones de IoT como es la monitorización ambiental, detección de incendios, seguridad, diagnóstico de enfermedades en el cuerpo humano, aplicaciones militares y la protección de vida silvestre en reservas ecológicas, ríos o entornos rurales. Sin embargo, detectar el movimiento de un objeto y reportar su ubicación continuamente conlleva un consumo energético excesivo. Por lo que es necesario desarrollar modelos eficientes que puedan extender la vida útil de la red.

Actualmente existen diferentes técnicas que minimizan la energía en OTSN. Muchos de estos modelos consisten en seleccionar el menor número de sensores para realizar el seguimiento del objeto móvil. De esta forma se evita que un sensor permanezca activo cuando no se requiere y por ende no consuma energía. Estos modelos suelen estar basados en aprendizaje automatizado, en minería de datos, en modelos probabilísticos, en estrategias jerárquicas o en diagramas de Voronoi. Sin embargo, en esta tesis se propone un algoritmo para la detección de un objeto móvil utilizando un concepto tomado de la geometría computacional llamado espacio dual en el que un punto de coordenadas  $(a, b)$  (espacio primal) se mapea a una recta  $y = ax - b$  (espacio dual) y viceversa. Esta transformación en comparación a los modelos previamente mencionados no requiere de un periodo prolongado de entrenamiento o de cómputo exhaustivo para determinar que sensores deben activarse conforme el objeto móvil se desplaza. Esto permite minimizar la energía consumida por los sensores y prolongar la vida útil de la red.





# Capítulo 1

## Introducción

Internet de las Cosas (IoT) alude a la interconexión de objetos (*cosas*) físicos o virtuales con Internet para ejecutar tareas y servicios avanzados en provecho de la sociedad de la información. Al hablar de una red de cosas, se hace referencia a dispositivos o sensores que detectan señales del mundo físico y lo registran digitalmente. Muchas de estas cosas suelen ser sensores alimentados por pequeñas baterías donde el ahorro de energía es fundamental en el despliegue y funcionamiento de este tipo de redes [1] [2]. Más aún, debido a las limitaciones en tamaño y costo de cada sensor, la capacidad de procesamiento en IoT suele ser baja [21].

Por otro lado, el seguimiento de objetos móviles en una red de sensores (OTSN) tiene un papel importante en muchas aplicaciones. Por ejemplo, la monitorización ambiental, detección de incendios, observación en el cuerpo humano, aplicaciones militares y la protección de vida silvestre en reservas ecológicas, ríos o entornos rurales. Estos objetos móviles pueden variar desde partículas, hasta seres vivos. Sin embargo, detectar el movimiento de un objeto y reportar su ubicación continuamente conlleva un consumo energético excesivo. Por ello surge la necesidad de implementar redes de sensores que administren de forma eficiente la energía para extender la vida útil de la red.

Actualmente existen algoritmos de minimización de energía en OTSN. Por ejemplo, algunos autores proponen encontrar patrones de movimiento y de esta forma activar o desactivar los sensores, tal es el caso de [5, 6, 10, 11]. Sin embargo, estas propuestas requieren de periodos prolongados de entrenamiento o de un conjunto muy grande de entradas para realizar una predicción correcta. Otros algoritmos se basan en el diagrama de Voronoi [7, 8], los cuales seleccionan a los sensores más cercanos al objeto móvil a partir de las teselaciones de Voronoi. Sin embargo, obtener un diagrama de Voronoi de manera distribuida no es tarea fácil.

A diferencia de los modelos previamente mencionados que requieren de un periodo prolongado de entrenamiento o de cómputo exhaustivo para determinar que sensores deben activarse, esta tesis usa un concepto tomado de la geometría computacional llamado espacio dual en el que un punto de coordenadas  $(a, b)$  (espacio primal) se mapea a una recta  $y = ax - b$  (espacio dual) y viceversa. Esta simple transformada permite minimizar la energía y prolongar la vida útil de la red.

Específicamente, en esta tesis, la localización del objeto móvil se realiza con dos sensores a la vez. En el momento que los sensores detectan la presencia del objeto móvil, cada sensor mide la distancia Euclidiana al objeto móvil e intercambian su información. Posteriormente, cada sensor traza una circunferencia con centro en su posición y radio igual a la distancia al objeto móvil. La intersección de estas circunferencias permite trazar una recta que indica la dirección de movimiento del objeto móvil. Esta recta es mapeada a un punto en el espacio dual y las posiciones de los sensores son mapeados a un conjunto de rectas. Conforme el objeto móvil se desplaza, su representación en el espacio dual (un punto) se acerca o aleja de las rectas que representan los sensores en el espacio dual. Esto permite seleccionar las rectas (sensores) más cercanas al objeto móvil y apagar los sensores más lejanos.

Las simulaciones muestran que mapear una red de sensores a un espacio dual permite encontrar propiedades que no son visibles en el espacio primal. Más aún, el algoritmo propuesto selecciona el menor número de sensores minimizando el gasto energético y aumentando la

vida útil de la red.

## 1.1. Definición de problema

Muchas de las redes en IoT suelen estar compuestas por sensores alimentados por pequeñas baterías, donde el ahorro de energía es fundamental en el despliegue y funcionamiento de este tipo de redes [1] [2].

OTSN se considera una de las aplicaciones que más demandan recursos energéticos en IoT. Esto se debe a que la tarea principal de OTSN es detectar el movimiento de un objeto y reportar su ubicación continuamente. Por lo que es necesario desarrollar modelos energéticos eficientes que extiendan la vida útil de la red. Actualmente existen algoritmos de minimización de energía en OTSN. Sin embargo, muchos de ellos requieren de periodos prolongados de entrenamiento o de cómputo exhaustivo que va en contra del ahorro de energía. Es por ello que el estudio de algoritmos que sean capaces de minimizar el gasto energético en los sensores es de suma importancia.

## 1.2. Hipótesis

*Modelos basados en aprendizaje automatizado, minería de datos o diagramas de Voronoi requieren de un cómputo exhaustivo para seleccionar cuál sensor debe activarse o desactivarse. Por el contrario, modelos basados en la transformada al espacio dual permiten encontrar propiedades para predecir qué sensores deben activarse de manera muy simple y así reducir el consumo de energía de la red.*

## 1.3. Aproximación al problema

Una red de sensores puede representarse como puntos en el espacio primal y como líneas en el espacio dual. La localización del objeto móvil en este trabajo se realiza mediante dos sensores a la vez, donde cada uno mide la distancia Euclidiana al objeto móvil. Posteriormente, cada sensor traza una circunferencia con centro en su posición y radio igual a la distancia Euclidiana al objeto móvil. La intersección de estas circunferencias permite trazar una recta que indica la dirección de movimiento del objeto móvil. Esta recta es mapeada a un punto en el espacio dual y la posición de los sensores se mapea a un conjunto de rectas. Conforme el objeto móvil cambia de posición, su representación en el espacio dual (un punto) se acerca o aleja de las rectas que representan los sensores. Esto permite seleccionar las rectas (sensores) más cercanas al objeto móvil y desactivar los sensores más lejanos.

## 1.4. Metas

### 1.4.1. Meta general

Proponer un algoritmo que sea capaz de minimizar el tiempo en el que los sensores están activos utilizando un concepto tomado de la geometría computacional llamado espacio dual.

### 1.4.2. Metas particulares

- Visualizar y analizar propiedades de la transformada al espacio dual.
- Representar y simular una red de sensores como puntos en el espacio primal y analizar su transformación al espacio dual.
- Desarrollar un algoritmo de minimización de energía a partir de las propiedades del espacio dual.

## 1.5. Metodología

El desarrollo de este proyecto consta de dos etapas. En la primera etapa, a partir de la definición de la transformada dual, se encuentran propiedades del espacio dual que no son evidentes en el espacio primal. Algunas de estas son la unión entre dos puntos, intersección de líneas, áreas finitas e infinitas y cierre convexo. Además, se simula una red de sensores y su transformación al espacio dual.

En la segunda etapa, por medio del análisis de las propiedades encontradas en la primera etapa, se propone un algoritmo de minimización de energía en una red de sensores que realiza el seguimiento de un objeto móvil.

## 1.6. Contribución

- En este trabajo se analizan propiedades de la transformada al espacio dual.
- Se simula una red de sensores en el espacio primal y su transformada en el espacio dual con el fin de obtener propiedades no evidentes en el espacio primal.
- Proponemos un algoritmo de minimización de energía, el cual selecciona los sensores que deben activarse y desactivarse para realizar el seguimiento de un objeto móvil.

## 1.7. Descripción del contenido

Para alcanzar la meta general planteada previamente, este trabajo de tesis se presenta de la siguiente manera:

- En el capítulo 2, se presenta una revisión de los trabajos más relevantes respecto a los algoritmos de minimización de energía en OTSN.
- En el capítulo 3, se presenta la definición de la transformada al espacio dual, además de las propiedades de la transformada del espacio primal al espacio dual ejemplificadas en un conjunto de figuras con diferentes ejemplos.
- En el capítulo 4, primero se propone un algoritmo capaz de localizar a un objeto móvil a partir de solo dos sensores. Posteriormente, se presenta como este algoritmo minimiza el tiempo en el que los sensores se encuentran activos. Finalmente, se realizan simulaciones para ejemplificar el funcionamiento del algoritmo.
- En el capítulo 5, se presentan las conclusiones generales, la verificación de la hipótesis y las perspectivas de investigación.



# Capítulo 2

## Antecedentes

### 2.1. Estado del Arte

En esta sección se presentan los trabajos más relevantes en el área de seguimiento de objetos móviles en redes de sensores.

El seguimiento de objetos móviles en redes de sensores (OTSN) se considera como una de las aplicaciones de mayor consumo de energía en las redes de sensores inalámbricas (WSN), dentro del paradigma de IoT [10]. La tarea principal de OTSN es seguir un objeto en movimiento y reportar su ubicación en el área monitorizada. Este proceso mantiene a la red en un continuo uso de recursos energéticos [10]. Debido a esto, actualmente existen algoritmos de minimización de energía en OTSN. Los más usados son modelos predictivos, cuya función es seleccionar los sensores necesarios para realizar el seguimiento. Por ejemplo, en [3] los autores realizan el seguimiento de un objeto (*robot ubicuo*) mediante trilateración, es decir, se debe garantizar al menos tres sensores activos en cada momento. Sin embargo, esto no siempre se puede cumplir en una red de sensores, ya que garantizar una cobertura de tres sensores es un problema *NP*.

En [5], se presenta un algoritmo de ahorro de energía distribuido que predice el seguimiento de objetos. Los autores proponen una red de sensores en un entorno boscoso en el que los objetos móviles a seguir son animales. Como los animales se comportan de forma predeterminada, se puede obtener un patrón de movimiento por cada animal y así seleccionar los sensores que deben activarse y de esta manera ahorrar energía.

En [10] y [11], se proponen modelos de minimización de energía sobre redes de sensores basados en minería de datos, que a diferencia de [5], el seguimiento se realiza de manera centralizada; es decir, un nodo principal procesa la información.

En [6], los autores proponen un modelo basado en aprendizaje automatizado (ML) aplicado en OTSN para eficientar el consumo de energía. Este consiste en predecir el comportamiento de los objetos móviles y activar un número pequeño de sensores. Además cuenta con un mecanismo de recuperación en caso de perder al objeto móvil. Sin embargo, ML requiere de muchos datos de entrada para realizar una predicción correcta.

En [12], se presenta una estrategia de predicción jerárquica. La red se divide en clústers por medio de una división de Voronoi. Estos clústers se componen de un nodo cabecera (CH), y un conjunto de nodos (NN); los primeros tienen mayor capacidad que los segundos. Los CH deciden qué nodos NN activar mediante un algoritmo predictivo basado en una técnica de mínimos cuadrados recursivos.

En [7] se presentan dos modelos para OTSN donde las áreas de cobertura de cada sensor no son uniformes. Mediante el uso de diagramas de Voronoi los autores proponen un algoritmo que evita que múltiples sensores detecten al objeto móvil en la misma área de cobertura y de esta forma reducir el número de sensores activos en la red.

En [8] al igual que en [7] se utiliza el diagrama de Voronoi con el fin de activar los sensores vecinos al polígono en el que se encuentra el objeto móvil. Gracias a esto solo se activan los sensores más cercanos y se desactivan aquellos sensores que se encuentran lejanos al objeto móvil.

En [14], los autores proponen un algoritmo para seguir un fenómeno físico como una tormenta o una nube de gases mediante el uso de una red de sensores. Para ello, delimitan el fenómeno físico mediante líneas continuas que son mapeadas a puntos en el espacio dual. De la misma forma las coordenadas de los sensores son mapeados a líneas rectas. Mediante un barrido topológico, los autores pueden detectar que sensores deben activarse y apagarse. Sin embargo, un barrido topológico requiere de cómputo exhaustivo y centralizado, ya que se requiere conocer la red entera.

A diferencia de [14], esta tesis no sigue un fenómeno físico de gran dimensión. Por el contrario, esta tesis propone un algoritmo que detecta a un objeto móvil a través de solo dos sensores. Para ello, los sensores envían un mensaje de *despierta* al vecino que esté más cerca del objeto móvil y un mensaje de *duerme* al sensor que está más lejos del objeto móvil mediante propiedades del espacio dual. A diferencia de [14] nuestra propuesta sólo requiere el conocimiento del vecindario a dos saltos y su complejidad es lineal.

## Capítulo 3

# Espacio dual

En este capítulo se presenta la definición de la transformada al espacio dual, además de las propiedades de la transformada del espacio primal al espacio dual ejemplificadas en un conjunto de figuras.

### 3.1. Transformación al espacio dual

Un punto en un plano  $\mathbb{R}^2$  se puede caracterizar por dos parámetros, la coordenada en el eje  $x$  y la coordenada en el eje  $y$ . De la misma forma, una línea (no vertical) en  $\mathbb{R}^2$  tiene dos parámetros, la pendiente y la ordenada al origen. La transformada dual consiste en mapear un punto a una línea y viceversa [16].

Sea  $p := (p_x, p_y)$  un punto en el espacio primal. Su transformada  $*p$  en el espacio dual se denota como una línea:

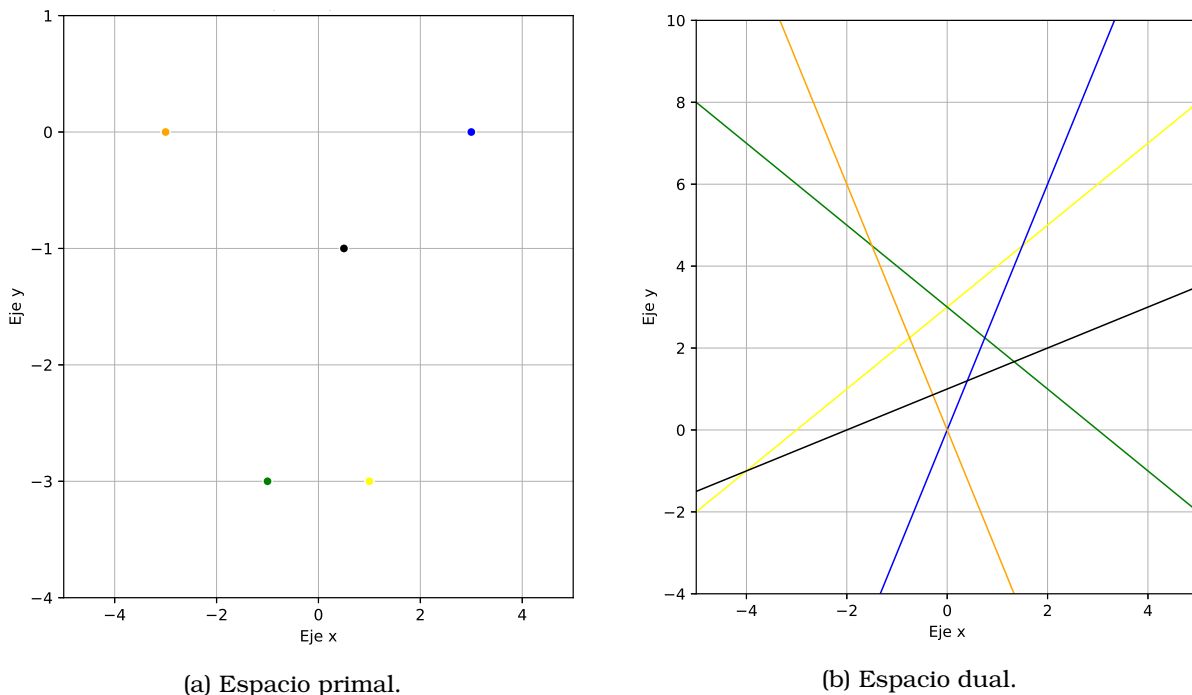
$$*p := (y = p_x x - p_y). \quad (3.1)$$

Sea una línea  $l : (y = mx - b)$  en el espacio primal. Su transformada  $*l$  en el espacio dual se denota como el punto:

$$*l := [m, -b]. \quad (3.2)$$

La figura 3.1(a) muestra un conjunto de puntos (naranja, azul, verde, amarillo y negro), mientras que la figura 3.1(b) muestra su transformada al espacio dual. Se puede observar en esta figura que cada una de las cinco líneas son intersecadas 4 veces, es decir, cada línea interseca con el resto de las líneas y a su vez dividen el espacio  $\mathbb{R}^2$  en polígonos convexos. Por ejemplo, una de estas áreas es el pentágono formado en el centro. También se puede observar en la figura que las pendientes de las líneas naranja y azul son las más inclinadas y que sus correspondientes puntos en el plano primal son los más alejados del punto  $x = 0$ . La línea negra es la menos inclinada y el punto negro es el más cercano a la coordenada  $x = 0$ . Asimismo, la pendiente positiva o negativa de una línea depende de si su coordenada  $x$  en el espacio primal es positiva o negativa. Además, se puede observar que los puntos verde y amarillo se encuentran ubicados por debajo de los puntos naranja y azul, y en el espacio dual, la intersección entre las líneas verde y amarillo está por encima de la intersección de las líneas azul y naranja.





(a) Espacio primal.

(b) Espacio dual.

Figura 3.1: Transformación de un conjunto de puntos del espacio primal a un conjunto de líneas al espacio dual.

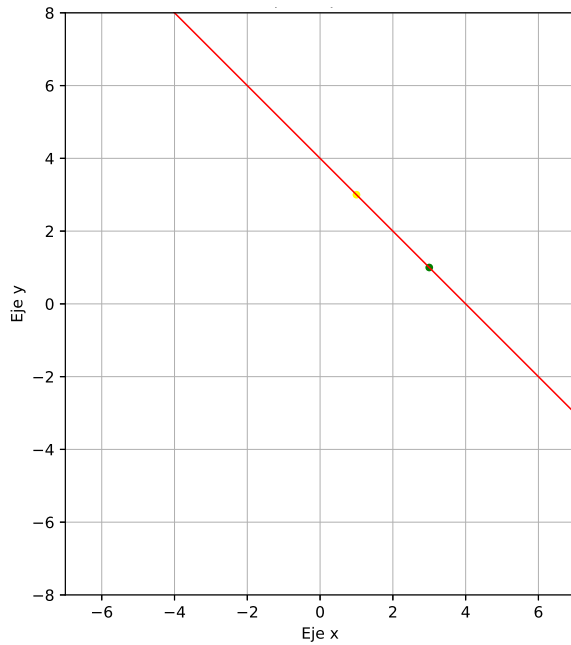
## 3.2. Propiedades del espacio dual

### 3.2.1. Intersección en el espacio dual.

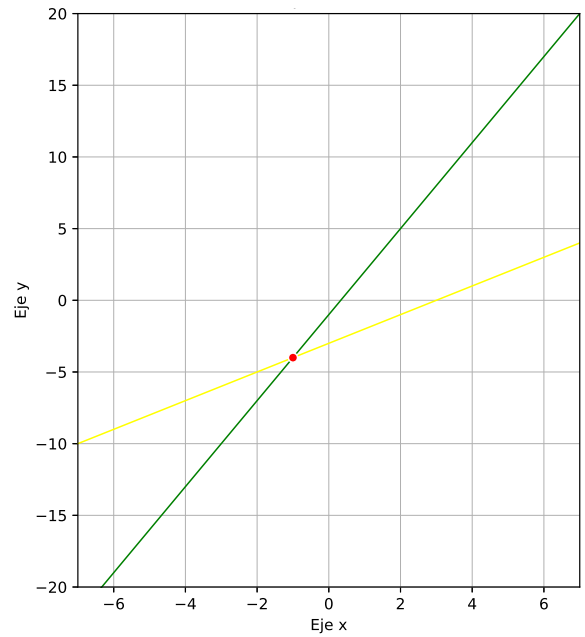
Dos puntos en el espacio primal, se mapean a dos líneas en el espacio dual. Si los puntos no comparten la misma coordenada en  $x$ , las líneas siempre intersecan en algún punto del espacio dual. La intersección de dos líneas en el espacio dual la llamamos *moño*. El centro del moño en el espacio dual es la línea que atraviesa los dos puntos en el espacio primal. Como se observa en la figura 3.2(a), la línea roja atraviesa los puntos amarillo y verde en el espacio primal. Mientras que en el espacio dual el punto rojo es el centro del moño que forman las líneas amarilla y verde.

Si un conjunto de puntos en el espacio primal es colineal; en el espacio dual las líneas intersecan en un mismo punto. Por ejemplo, en la figura 3.3(a), la línea roja atraviesa todos los puntos en el espacio primal, mientras que en el espacio dual (ver figura 3.3(b)), todas las líneas intersecan el punto rojo.

En la figura 3.3(a), se puede observar que los puntos extremos son el amarillo y el negro, y entre ellos se encuentran los puntos verde, azul y naranja, todos ellos colineales. Mientras que en el espacio dual (ver figura 3.3(b)), todos los moños están contenidos dentro del moño formado por las líneas amarillo y negro.

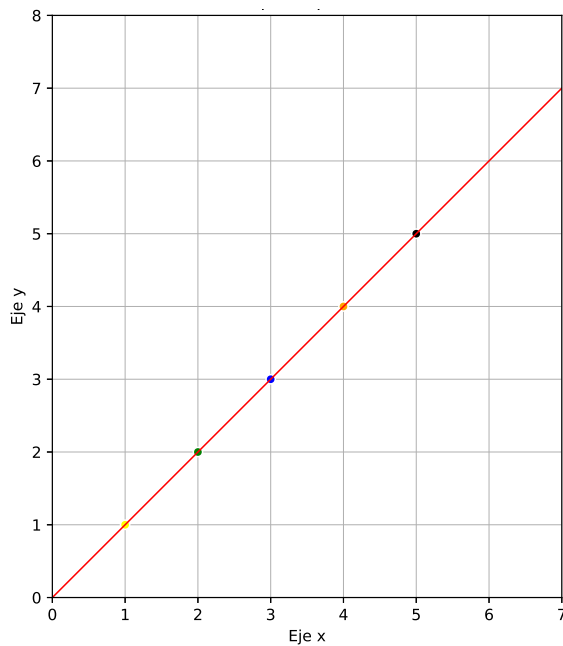


(a) Espacio primal.

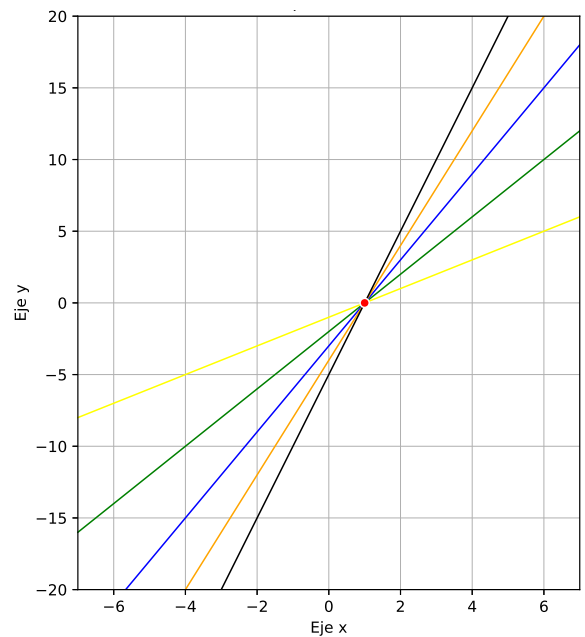


(b) Espacio dual.

Figura 3.2: Transformación de dos puntos del espacio primal al espacio dual (*moño*).



(a) Espacio primal.



(b) Espacio dual.

Figura 3.3: Puntos colineales.

### 3.2.2. Áreas en el espacio dual

Tanto en el espacio primal como en el dual pueden formarse áreas infinitas o finitas. Por ejemplo, la figura 3.4 muestra los dos tipos de áreas sobre un conjunto de 5 líneas. En la figura 3.4(a) se muestran áreas que no están delimitadas (coloreadas en gris). Aunque esta figura está acotada en ambos ejes, puede extenderse hasta el infinito. Mientras que en la figura 3.4(b) se muestran áreas delimitadas, se necesitan al menos tres líneas para formarse.

Es importante notar que es imposible que existan áreas delimitadas y áreas infinitas que estén compuestas por las mismas líneas. Por ejemplo, en la figura 3.4(a) se genera una área infinita a partir de las líneas verde, naranja, amarillo y negro. Y en la figura 3.4(b) no existe una área delimitada a partir de las mismas líneas.

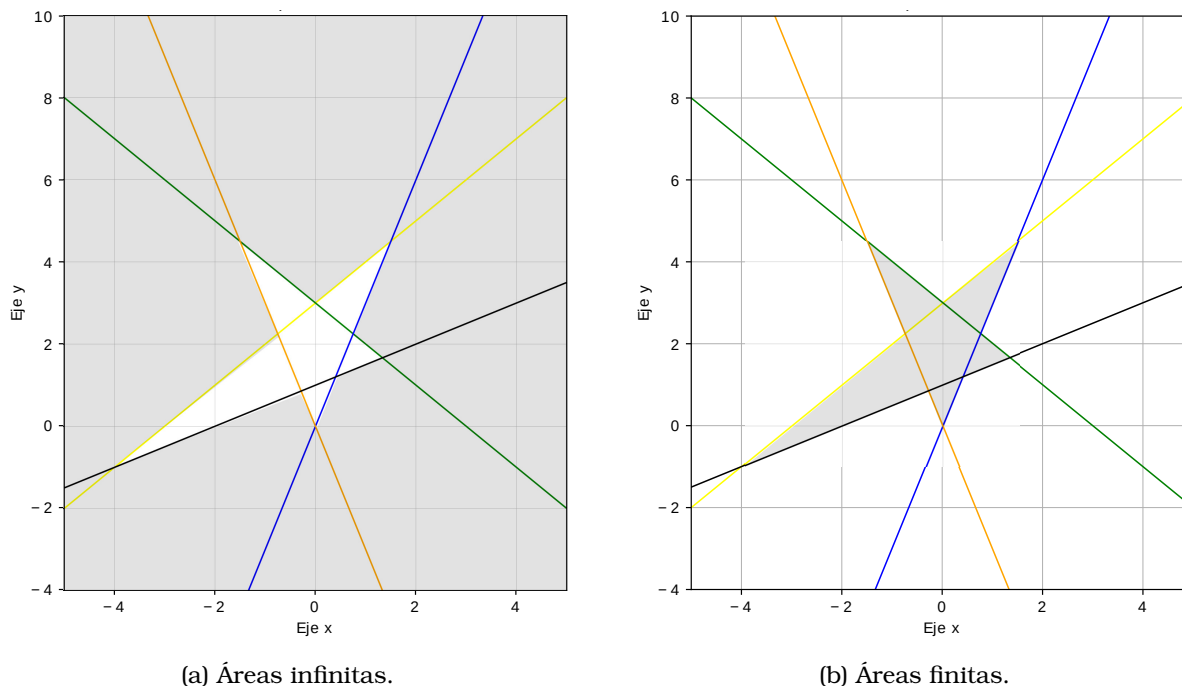
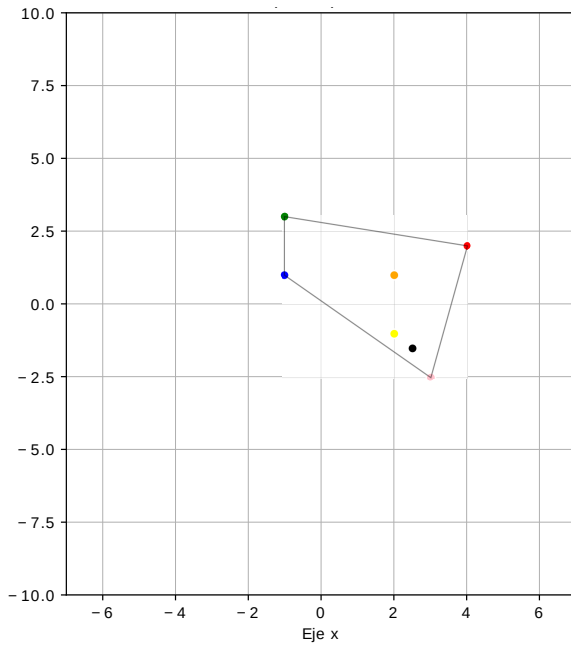


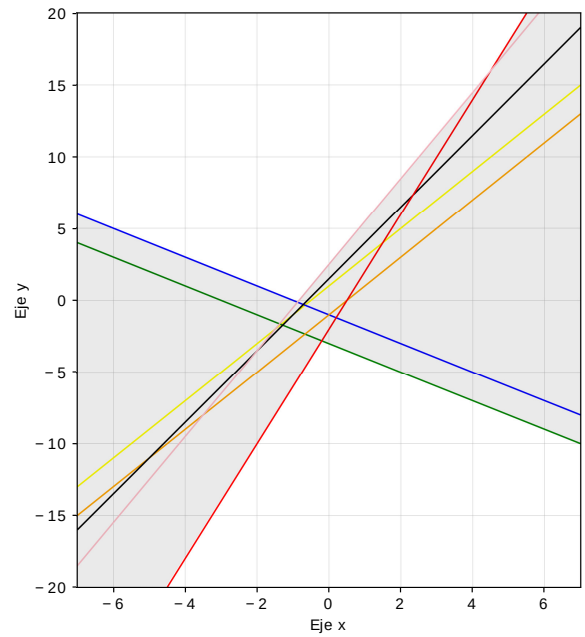
Figura 3.4: Áreas infinitas y delimitadas.

### 3.2.3. Cierre convexo

El *cierre convexo* de un conjunto de puntos se define como el conjunto convexo más pequeño que contiene a todos los puntos, o como la intersección de todos los conjuntos convexos que contienen a todos los puntos [16]. Generalmente, el algoritmo para obtener el cierre convexo consiste en obtener una envolvente superior y una envolvente inferior. De la misma forma, el problema de calcular el cierre convexo en el plano dual, se reduce a calcular la envolvente superior e inferior de un conjunto de líneas. Por ejemplo, en la figura 3.5(a) los puntos que pertenecen al cierre convexo en el espacio primal son 4, estos cuatro puntos contienen a todos los demás puntos. La envolvente superior corresponde a los puntos rojo y verde, mientras que el envolvente inferior a los puntos rojo, rosa y azul. De la misma manera, en el espacio dual la envolvente superior está conformada por las líneas azul, rosa y roja (ver figura 3.5(b)), mientras que la envolvente inferior está compuesta por las líneas roja y verde. Estas líneas contienen a todas las demás, es decir, las líneas amarilla, naranja y negra siempre estarán dentro del cierre convexo. Es importante notar que en el espacio dual se invierten las envolventes superior e inferior (ver la figura 3.5(b)) con respecto al primal. Es conocido que la complejidad del algoritmo es  $O(n \log n)$  en ambos espacios [20]. Es importante notar, que las líneas que pertenecen al cierre convexo en el espacio dual, forman una área infinita a diferencia del espacio primal. Aún así, esta área contiene al resto de las líneas. Sin embargo, se puede crear una área finita a través de todas las intersecciones posibles entre dos líneas. Por ejemplo, en la figura 3.6(a), los puntos azul, verde, rojo y rosa forman el cierre convexo, mientras los puntos negros en la figura 3.6(b) representan todas las intersecciones posibles de las líneas que forman el cierre convexo. En este ejemplo, las líneas azul y verde son paralelas, ya que su coordenada en el eje  $x$  es igual a  $-1$ , por lo que no intersecan entre ellas.

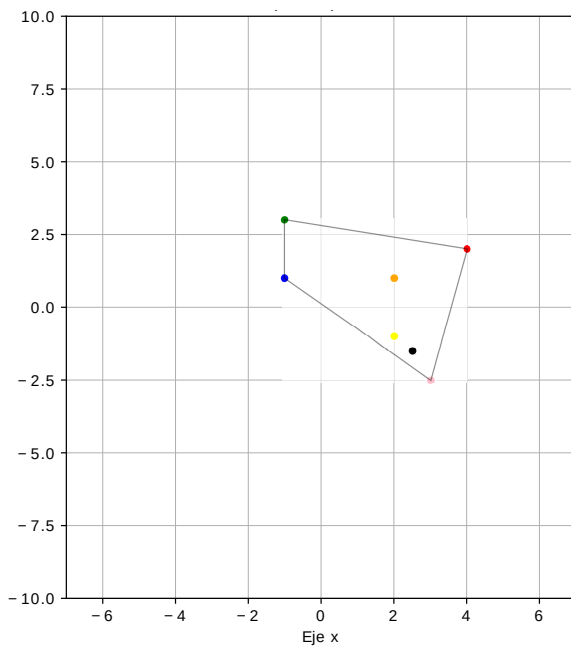


(a) Cierre convexo en el espacio primal.

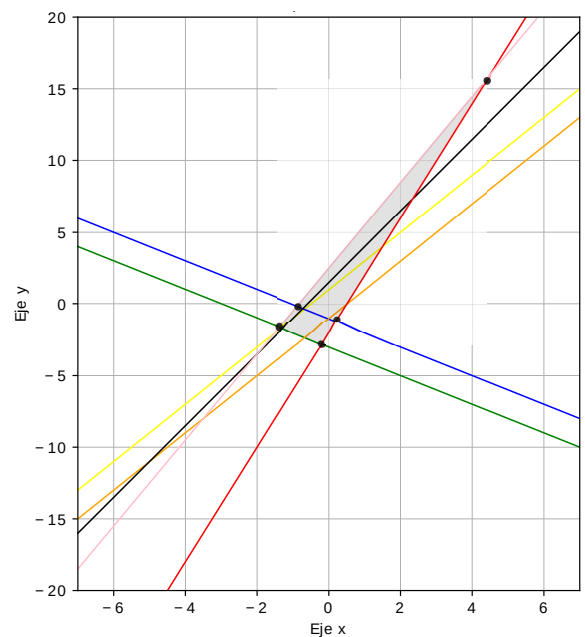


(b) Cierre convexo en el espacio dual.

Figura 3.5: Cierre convexo en el espacio primal y en el espacio dual.



(a) Cierre convexo en el espacio primal.



(b) Área delimitada en el espacio dual.

Figura 3.6: Área delimitada con las intersecciones del cierre convexo.

### 3.2.4. Ángulo de intersección de dos líneas en el espacio dual.

El ángulo de intersección entre dos líneas en el espacio dual describe la similitud entre sus pendientes, mientras que en el plano primal, describe que tan cercanos o lejanos son esos puntos. Este ángulo ( $\theta$ ) se puede obtener mediante:

$$\theta = \arctan \frac{(m1 - m2)}{1 + (m2 * m1)}, \quad (3.3)$$

donde  $m1$  y  $m2$  son las pendientes de las líneas  $l_1$  y  $l_2$ , respectivamente. Cuando dos líneas se intersecan, se forman cuatro ángulos, de los cuales los opuestos son iguales. Ya que los ángulos no opuestos son suplementarios, se puede conocer el ángulo suplementario de  $\theta$  al restarle 180 grados. Finalmente, se puede comparar la similitud de las pendientes mediante:

$$\Delta\theta = |\theta - (180 - \theta)|. \quad (3.4)$$

Entre más cercano  $\Delta\theta$  a 180, mayor similitud tendrán las dos pendientes. Esto implica que los puntos en el plano primal están muy cercanos o viceversa. Esta propiedad solo está definida para cuando las coordenadas en  $x$  de todos los puntos en el espacio primal son todas negativas o todas positivas. Por ejemplo, en la figura 3.6(a) se observa que los puntos dentro del cierre convexo son amarillo, negro y naranja. Y el punto negro es el punto más cercano al cierre convexo. En el espacio dual (ver figura 3.6(b)), se observa que la línea rosa tiene una pendiente muy similar a la línea negra.

### 3.3. Punto en movimiento sobre una línea

En las figuras 3.7(a) y 3.8(a), se muestra un punto negro que se mueve a lo largo de una línea roja, mientras que los puntos verde, azul, naranja y amarillo permanecen estáticos. Por otro lado, en las figuras 3.7(b) y 3.8(b), se muestra su transformada al espacio dual, conservando los mismos colores.

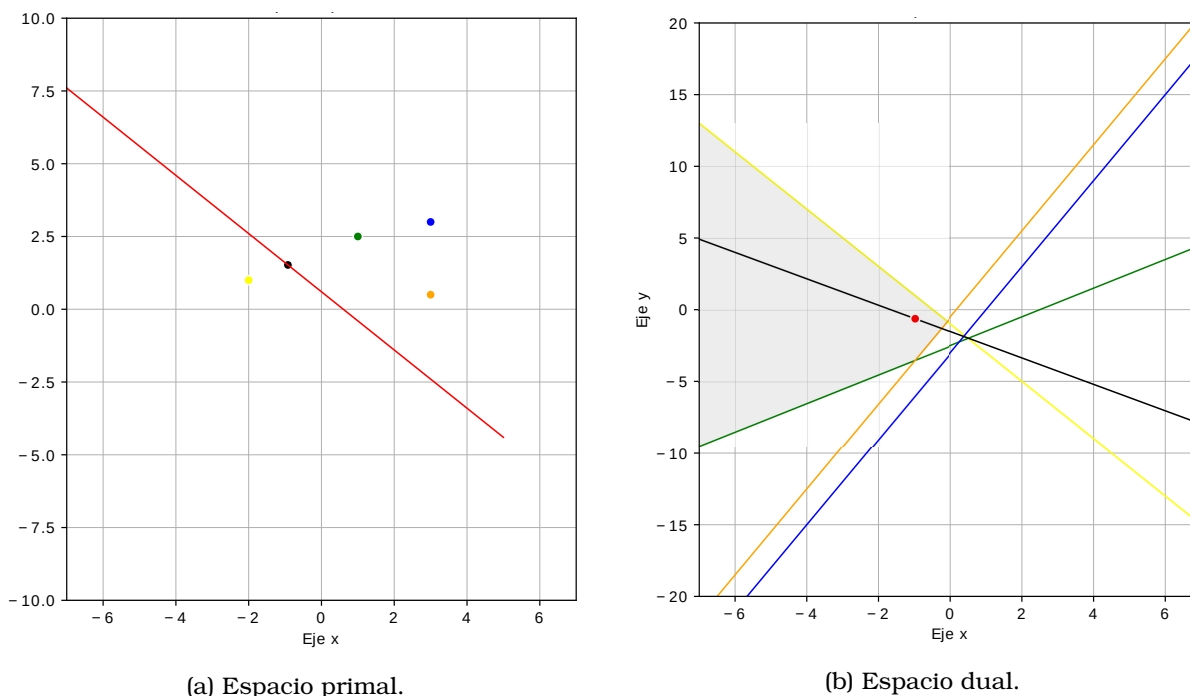
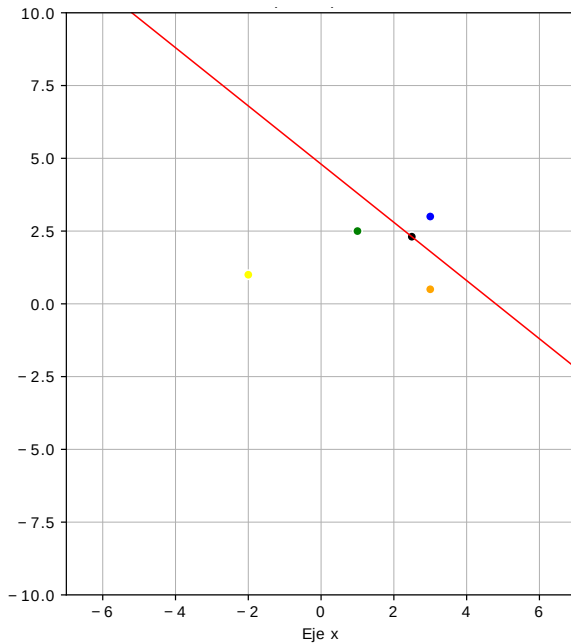


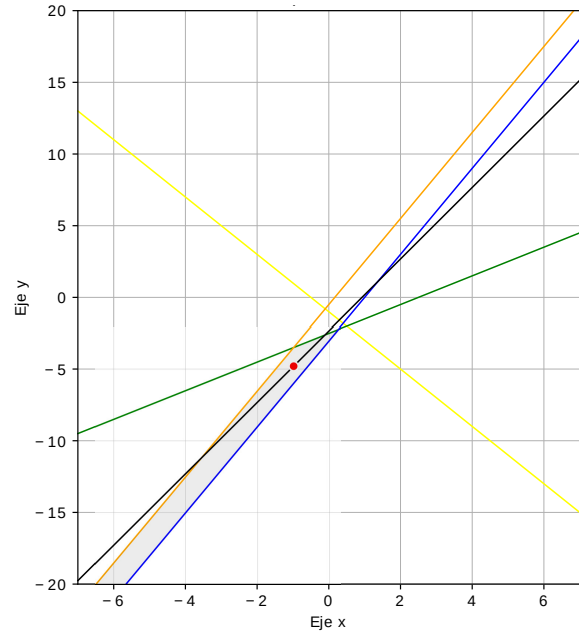
Figura 3.7: Cinco puntos y una línea en tiempo  $t$ .

En la figura 3.7(b) se puede observar una área sombreada, la cual encierra al punto rojo que describe la línea roja a través de la cual se mueve el punto negro. Los bordes del área sombreada son la línea verde, naranja y amarillo. En la figura 3.7(a) se puede observar que la línea roja se encuentra entre los puntos amarillo, verde y naranja. También se puede observar que la línea con pendiente más cercana a la línea negra es la línea amarilla, debido a que su ángulo de intersección es el más chico.

En la figura 3.8(b) al igual que en la figura anterior, el punto rojo está contenido en el área sombreada, sin embargo en la figura 3.8(a) tanto la línea roja como el punto negro se desplazaron con respecto a la figura 3.7(a). Ahora los bordes del área sombreada son las líneas naranja, verde y azul. En la figura 3.8(a) se puede observar que la línea roja se encuentra entre los puntos verde, azul y naranja. El punto más cercano a la línea roja es el punto azul, mientras que en el plano dual, la mayor similitud entre pendientes se encuentra entre la recta negra y la azul.



(a) Espacio primal.



(b) Espacio dual.

Figura 3.8: Cinco puntos y una línea en tiempo  $t + 1$ .

En general, a través de los ejemplos presentados, se puede observar que un punto en el plano dual encerrado en una determinada área puede describir la dirección de un punto que se mueve sobre una línea con respecto a un conjunto de puntos estáticos, es decir, dividir el espacio en polígonos permite ubicar con mayor facilidad la posición de un punto en el espacio dual. Además, se puede observar que entre menor sea el ángulo de intersección entre líneas, implica que en el espacio primal un punto se encuentra cercano a otro punto.



# Capítulo 4

## Desarrollo

En este capítulo, primero se propone un algoritmo capaz de localizar a un objeto móvil a partir de solo dos sensores. Posteriormente, se presenta como este algoritmo minimiza el tiempo en el que los sensores se encuentran activos. Finalmente, se realizan simulaciones para ejemplificar el funcionamiento del algoritmo.

### 4.1. Dos sensores y un objeto móvil

El método de localización más utilizado en las redes de sensores es conocido como *trilateración*. Este método consiste en la intersección de tres circunferencias cuyos centros (no colineales) son las posiciones de los sensores involucrados y cuyos radios son iguales a la distancia Euclidiana entre cada sensor y el objeto a localizar. La intersección de las tres circunferencias describe la ubicación del objeto. Sin embargo, garantizar una cobertura de tres sensores durante el seguimiento de un objeto móvil, no siempre es posible. Por ello, en esta tesis proponemos un algoritmo que es capaz de seguir a un nodo móvil con tan solo dos sensores basado en la transformada al espacio dual.

Dos sensores  $C_1$  y  $C_2$  con centros  $(a_1, b_1)$  y  $(a_2, b_2)$ , respectivamente, trazan dos circunferencias con radio igual a la distancia ( $d_1$  y  $d_2$ , respectivamente) Euclidiana entre sus centros y el objeto en movimiento. Las dos circunferencias intersecan en dos puntos. Estos puntos son las soluciones matemáticas de igualar las ecuaciones de cada circunferencia. Pero, solo uno de ellos es la posición real del objeto móvil. Sin más información que esta, los sensores no pueden distinguir cual de estas dos soluciones es la posición real del objeto. Es decir, los sensores tienen un 50% de ambigüedad en la localización del objeto móvil. Sin embargo, se puede observar que las dos soluciones son colineales a una línea recta  $l_i$ , la cual siempre es perpendicular a la línea que une a los dos sensores. Es importante observar, que la línea  $l_i$  representa la dirección de movimiento del objeto móvil. Esta línea  $l_i$  se puede obtener a partir de igualar la ecuación de cada circunferencia, ya que se conocen los centros y las distancias (radios) al objeto móvil de la siguiente forma:

$$C_1 = (x - a_1)^2 + (y - b_1)^2 - d_1^2 = 0, \quad (4.1)$$

$$C_1 = x^2 + y^2 - (2 * a_1)x - (2 * b_1)y + a_1^2 + b_1^2 - d_1^2, \quad (4.2)$$

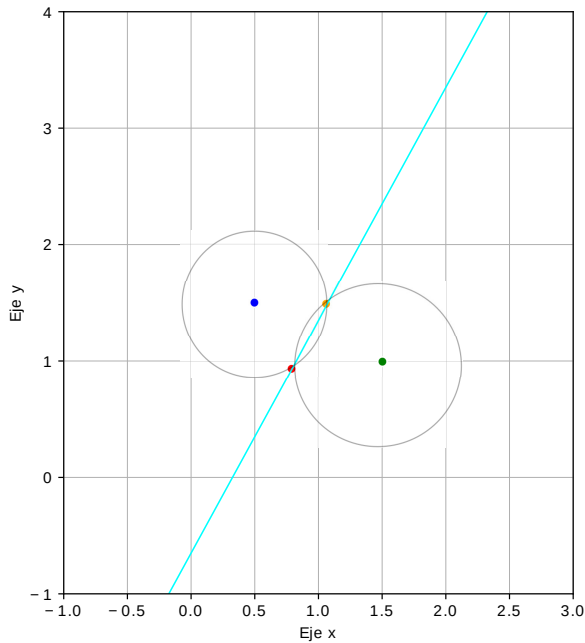
$$C_2 = x^2 + y^2 - (2 * a_2)x - (2 * b_2)y + a_2^2 + b_2^2 - d_2^2, \quad (4.3)$$

$$l_i = C_1 - C_2, \quad (4.4)$$

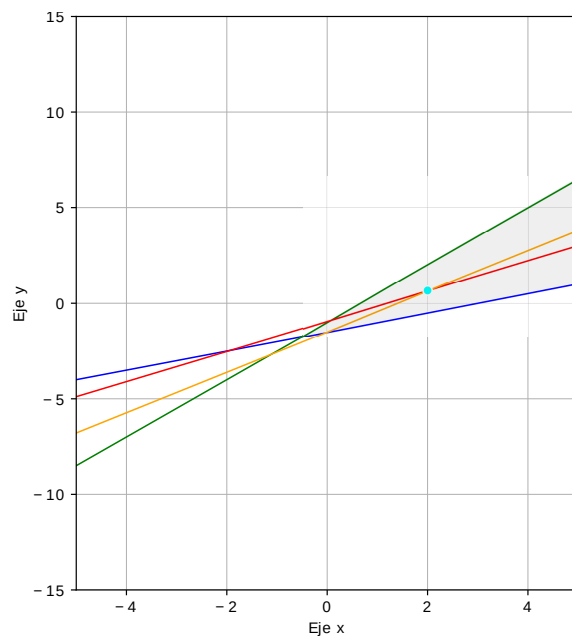
$$l_i = 2(a_2 - a_1)x + (b_2 - b_1)y + a_1^2 + b_1^2 - d_1^2 - a_2^2 - b_2^2 + d_2^2. \quad (4.5)$$



Por ejemplo, en la figura 4.1(a) los sensores azul y verde miden la distancia  $d_1$  y  $d_2$  al objeto móvil. Posteriormente, trazan una circunferencia alrededor de sí mismos. Ambas circunferencias intersecan en los puntos rojo y naranja (soluciones matemáticas a la intersección). Se puede ver en esta figura que la línea cian ( $l_i$ ) es colineal a las dos posibles posiciones del objeto móvil (los puntos rojo y naranja). En la figura 4.1(b) se observa que el punto cian ( $*l_i$ ) se encuentra en el área sombreada que corresponde al área superior del moño que forman los sensores azul y verde. Esto se debe a que la pendiente de  $l_i$  en el plano primal es positiva.



(a) Espacio primal.

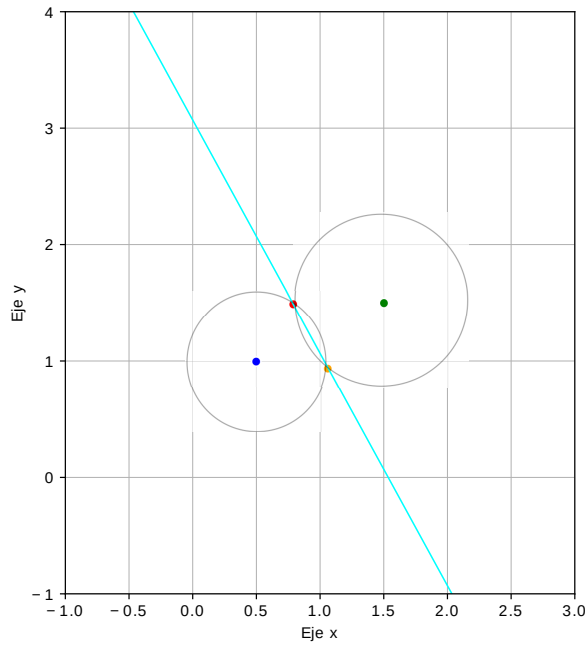


(b) Espacio dual.

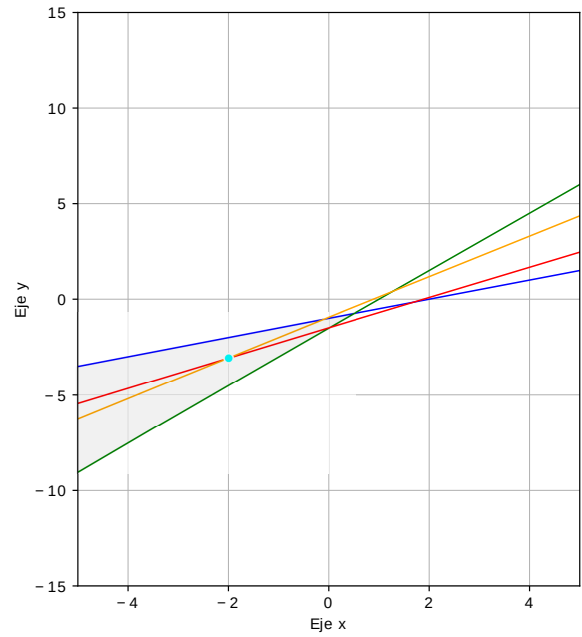
Figura 4.1: Dos sensores y un objeto móvil: pendiente positiva.

En la figura 4.2(a) se invierte el esquema de la figura 4.1(a). La línea cian ( $l_i$ ) tiene pendiente negativa. El punto cian ( $*l_i$ ) en el espacio dual se encuentra en el área inferior del moño que forman los sensores azul y verde.

En general, si la línea  $l_i$  tiene pendiente positiva, su transformada en el espacio dual ( $*l_i$ ) se encontrará en la parte superior del moño, por el contrario, si la pendiente de  $l_i$  es negativa, su transformada  $*l_i$  estará en la parte inferior del moño. Es importante mencionar que el punto  $*l_i$  siempre estará dentro del área formada (moño) por los sensores que lo están monitorizando en el plano dual.

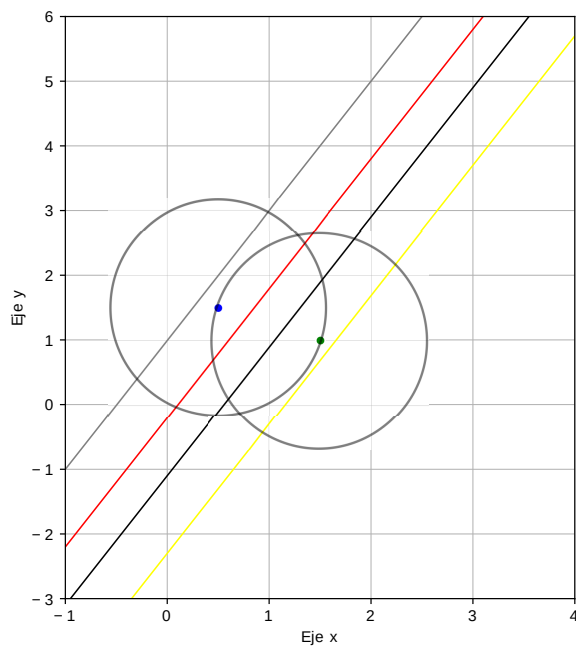


(a) Espacio primal.

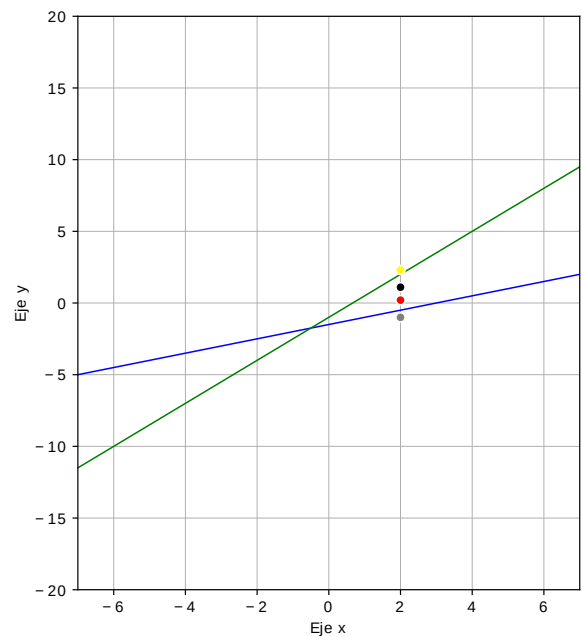


(b) Espacio dual.

Figura 4.2: Dos sensores y un objeto móvil: pendiente negativa.



(a) Espacio primal.



(b) Espacio dual.

Figura 4.3: Dos sensores y un objeto móvil: cuatro líneas  $l_i$ .

En la figura 4.3(a) se muestran cuatro líneas  $l_i$  y dos sensores, azul y verde. Las líneas roja y negra se encuentran dentro del área de cobertura de los sensores, mientras que las líneas gris y amarilla se encuentran fuera del área de cobertura de los sensores. La línea roja está cerca del sensor azul. En la figura 4.3(b), el punto rojo también se encuentra próximo a la línea (sensor) azul. El punto gris está cerca del sensor azul, pero se encuentra fuera del área superior del moño que forman los dos sensores. Esto se debe a que en el espacio primal (ver figura 4.3(a)), la línea gris está fuera del área de cobertura del sensor. Lo mismo ocurre en el

espacio dual con el sensor verde. El punto negro se encuentra dentro del moño que forman los dos sensores. Y el punto amarillo se encuentra fuera del moño.

En la figura 4.3(b) se observa que los puntos se encuentran siempre sobre la misma coordenada en  $x$ . Solo varía la coordenada en  $y$ . Esto se debe a que la pendiente de la línea  $l_i$  no es variable, ya que esta línea siempre es perpendicular a la línea que une a los sensores, es decir, la coordenada en  $y$  determina la cercanía con algún sensor.

Para ejemplificar el algoritmo, la figura 4.4(a) muestra a un objeto que está en movimiento y dos sensores (amarillo y verde) que detectan su presencia, estos sensores estiman la distancia a el objeto e intercambian su información para trazar la línea  $l_i$  (línea naranja). Si el objeto móvil sigue el movimiento de la flecha roja, éste en algún momento saldrá del área de cobertura de los sensores amarillo y verde. Es decir, en el espacio primal (ver figura 4.4(a)), la línea naranja se acerca al sensor verde. Mientras que en el espacio dual (ver figura 4.4(b)), el punto naranja se acerca a la línea verde y posteriormente cambiará al moño formado por los sensores azul y verde, si el nodo móvil continúa en la dirección de la flecha roja, entrará al moño formado por los sensores azul y rosa.

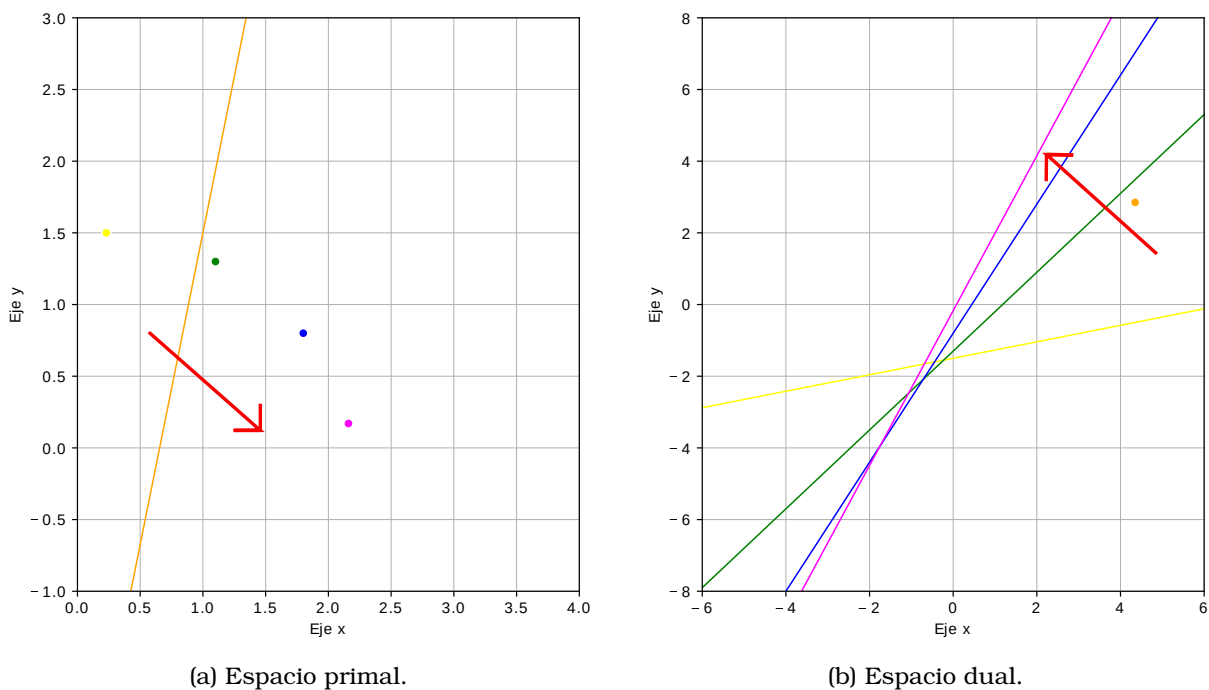


Figura 4.4: Cuatro sensores y un objeto móvil: moños consecutivos.

En el ejemplo de la figura 4.5 la red consta de cuatro sensores, tres aristas y la línea  $l_i$  en el espacio primal. Los sensores verde y amarillo detectan la presencia del objeto móvil. El punto  $*l_i$  se encuentra en la parte inferior del moño formado por los sensores amarillo y verde, sin embargo al mismo tiempo se encuentra en la parte superior del moño formado por los sensores azul-rosa.

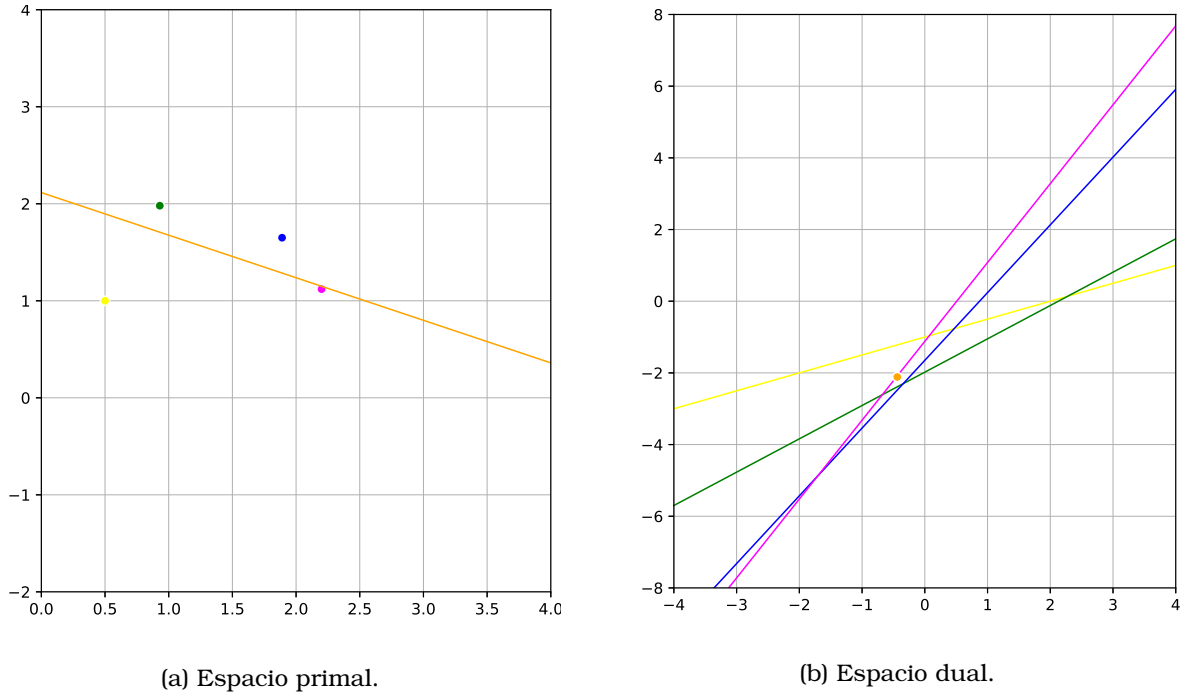


Figura 4.5: Cuatro sensores y un objeto móvil: cambio de pendiente.

## 4.2. Algoritmo

A continuación se propone un algoritmo que predice cuáles sensores deben activarse o desactivarse conforme el objeto a seguir cambia de posición. Este algoritmo se presenta en dos etapas:

- Conocimiento de la red a dos saltos.
- Activación y desactivación de sensores.

Para la primera etapa del algoritmo cada sensor debe conocer su vecindario a dos saltos mediante alguna técnica de bajo consumo de energía como: *RFC 6775* o *RFC 4861* [23]. A partir de la red a dos saltos se debe crear un grafo plano. Por ejemplo, mediante el uso de *Gabriel Graph* [18]. Con ello, solo los sensores que pertenecen al cierre convexo se mantienen activos para detectar por primera vez al objeto móvil (OM), los demás se mantienen desactivados.

La segunda etapa inicia cuando dos sensores  $s_1$  y  $s_2$  tienen en su área de cobertura  $AC$  (el radio máximo de comunicación) al objeto móvil.  $s_1$  y  $s_2$  miden la distancia  $d_{s1}$  y  $d_{s2}$  al objeto móvil. Se intercambian su información: ubicación, ID y la distancia al objeto. A partir de esta información uno de ellos toma el control y obtiene la línea  $l_i$  con la ecuación 4.5. Mapea su información a dos saltos, junto con la línea  $l_i$  al espacio dual, mediante las ecuaciones 3.1 y 3.2. El punto  $*l_i(x_i, y_i)$  se encuentra dentro del moño formado por los sensores  $s_1$  y  $s_2$ ; si la coordenada  $x_i$  es negativa,  $*l_i$  está en el área inferior del moño, de lo contrario está en el área superior del moño. Conforme el objeto móvil cambia de posición, la coordenada  $y_i$  de  $*l_i$  varía (sube o baja), mientras que la coordenada  $x_i$  de  $*l_i$  se mantiene constante (ver Sección 4). Por lo que  $y_i$  puede intersecar con el moño que lo contiene. Sean  $y_{m1}$  y  $y_{m2}$  las intersecciones de  $*l_i$  con el moño que lo contiene.

Existen tres posibles casos:

- Si  $y_i \approx y_{m1}$ , entonces  $s_1$  activa a sus vecinos y desactiva a  $s_2$ .
- Si  $y_i \approx y_{m2}$ , entonces  $s_2$  activa a sus vecinos y desactiva a  $s_1$ .
- Si  $d_{s1} \approx AC$  o  $d_{s2} \approx AC$ , y no se cumplen las condiciones anteriores. Se busca en que otros moños está  $*l_i$ . Para ello, se comparan las ordenadas de cada moño ( $y_{mj}$  y  $y_{mk}$ )

con respecto a la ordenada  $y_i$ , para una  $x_i$  constante, de tal forma que cumpla que  $y_{mj} < y_i < y_{mk}$  o  $y_{mk} < y_i < y_{mj}$ . Si esto es cierto, entonces se activa el moño  $s_j-s_k$  y se desactiva el moño  $s_1-s_2$ .

El algoritmo tiene una complejidad  $O(n)$ , ya que en el peor de los casos se debe comparar la ordenada  $y_i$  con los  $n$  moños posibles a dos saltos del sensor que tomó el control. A continuación se presenta el algoritmo 1 en pseudocódigo.

---

**Algoritmo 1** Minimización de energía
 

---

**Input:**  $s_1(a_1, b_1)$ ,  $s_2(a_2, b_2)$ ,  $d_{s_1}$ ,  $d_{s_2}$

**Output:** *nextBunActive*: Los siguientes moños que deben activarse

```

1:  $l_i = 2(a_2 - a_1)x + (b_2 - b_1)y + a_1^2 + b_1^2 - d_{s_1}^2 - a_2^2 - b_2^2 + d_{s_2}^2$ 
2:  $*(GG(G) = (V, E))$ 
3:  $*l_i(x_i, y_i)$ 
4:  $y_{m1} = a_1 * x_i - b_1$ 
5:  $y_{m2} = a_2 * x_i - b_2$ 
6: if  $y_i \approx y_{m1}$  then
7:    $s_2 = 0$ ;  $s_1 = 1$ 
8:   nextBunActive : vecinos de  $s_1$ 
9: if  $y_i \approx y_{m2}$  then
10:   $s_1 = 0$ ;  $s_2 = 1$ 
11:  nextBunActive : vecinos de  $s_2$ 
12: if  $AC \approx d_{s_1}$  or  $AC \approx d_{s_2}$  then
13:   for  $n$  in  $*E$  do
14:      $y_{mu} = x_u * x_i - b_u$ 
15:      $y_{mv} = x_v * x_i - b_v$ 
16:     if  $y_{mu} < y_i < y_{mv}$  or  $y_{mv} < y_i < y_{mu}$  then
17:       nextBunActive :  $s_u, s_v$ 
18:        $s_1 = s_2 = 0$ 
19: return nextBunActive

```

---

La línea 1 calcula  $l_i$  a partir de la ecuación 4.5 utilizando las coordenadas de los sensores y las distancias al objeto móvil. Las líneas 2 y 3 mapean el grafo a dos saltos y  $l_i$  al espacio dual. En las líneas 6 – 8 se activan los vecinos de  $s_1$ , se mantiene activo  $s_1$  y se desactiva  $s_2$ , dado que  $y_i$  se aproximó a  $y_{m1}$ , es decir, el punto  $*l_i$  se aproximó a  $s_1$ . En las líneas 9 – 11 se realiza el proceso inverso a las líneas 6 – 8, ya que el  $*l_i$  se aproxima al sensor  $s_2$ . En las líneas 12 – 18, se realiza la tercera condición, en la que se busca si el punto  $*l_i$  se encuentra dentro de algún moño adicional al moño formado por  $s_1$  y  $s_2$ . Si es así, se activan los sensores correspondientes a ese moño y se desactivan los sensores  $s_1$  y  $s_2$ .

### 4.3. Simulación

A continuación se presentan dos ejemplos del funcionamiento del algoritmo en una serie de figuras, cada una de estas figuras representa la misma red de sensores en diferentes tiempos.

Las simulaciones se realizaron utilizando el lenguaje de alto nivel Python y la biblioteca Matplotlib, la cual permite realizar figuras de alta calidad en una variedad de formatos y entornos [17].

Se establecieron 5 sensores cuyas coordenadas se presentan a continuación y con un radios de cobertura de 0,74.

Las coordenadas de los sensores son:

- Sensor negro: (0.31,0.92)
- Sensor verde: (1.0,1.0)
- Sensor amarillo: (1.5,0.5)
- Sensor azul: (1.35,1.64)
- Sensor rosa: (2.1,1.9)

Las figuras con inciso (a) representan el espacio primal; los sensores están representados como puntos negro, verde, azul, amarillo y rosa. Y la línea roja  $l_i$  representa la dirección de movimiento del objeto móvil. También se presentan los enlaces entre sensores, los cuales se representan con líneas grises. La transformada al espacio dual se muestra en las figuras con inciso (b).

#### 4.3.1. Ejemplo 1

En la figura 4.6(a) cada sensor conoce sus vecinos a dos saltos y se tiene un grafo plano.

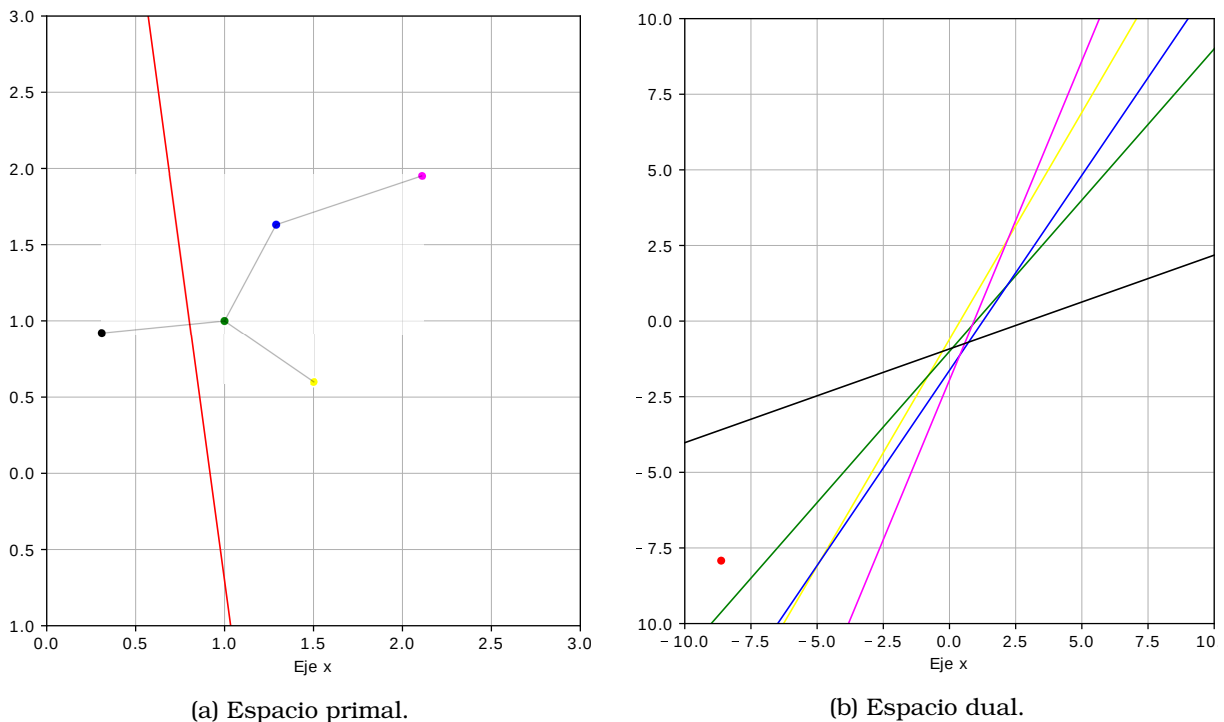


Figura 4.6: Ejemplo 1: Simulación del algoritmo en tiempo  $t$ .

Los sensores negro  $s_1$  y verde  $s_2$  detectan la presencia del objeto móvil y obtienen la distancia a él, posteriormente intercambian esta información para calcular la dirección del objeto

móvil (línea roja  $l_i$ ). En la figura 4.6(b) el punto  $*l_i$  se ubica en el moño que forman los sensores negro y verde. Se puede observar que  $*l_i$  se ubica más próximo al sensor verde. También se puede observar que este punto aparece en la parte inferior del moño dado que la pendiente de  $l_i$  es negativa.

En la figura 4.7(b) el objeto móvil se aproxima aún más a la línea verde  $s_2$ . Por lo tanto, se utiliza la segunda condición del algoritmo 1, donde se desactiva el sensor negro  $s_1$ , se mantiene activo el sensor verde y se activan los sensores azul y amarillo, que son los sensores a un salto del sensor verde. Esto es debido a que en este momento no se sabe si el objeto se moverá hacia el nodo azul o hacia el nodo amarillo.

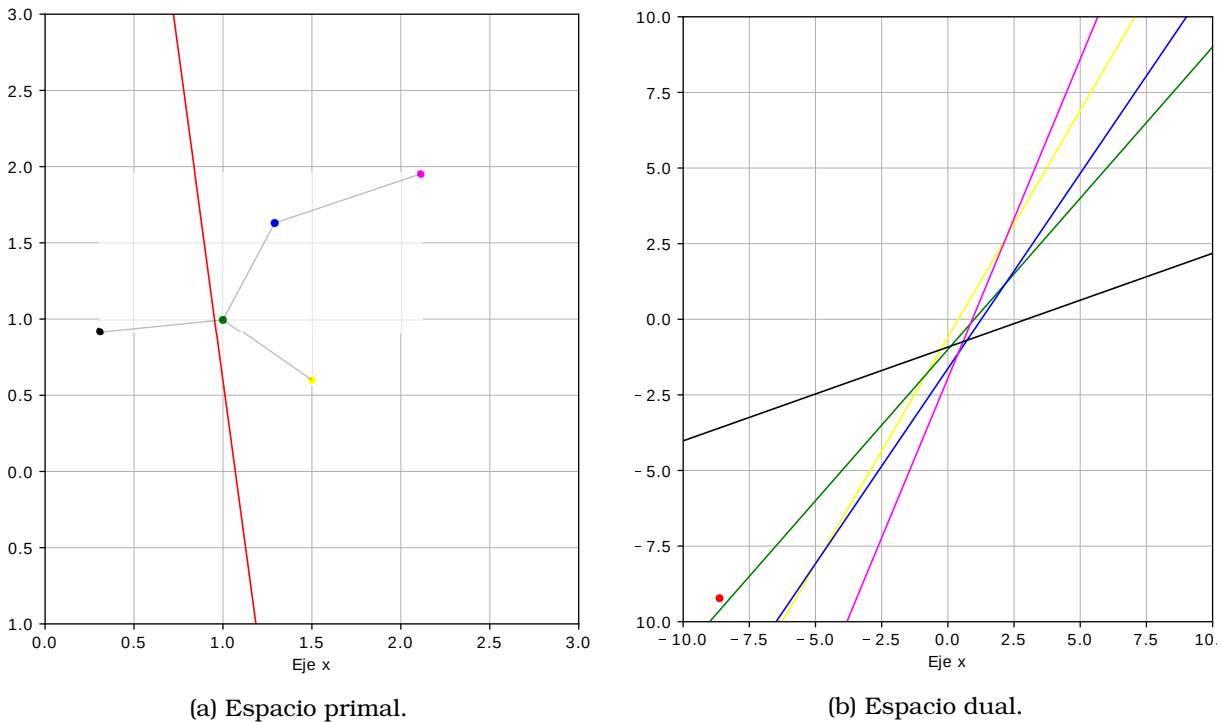
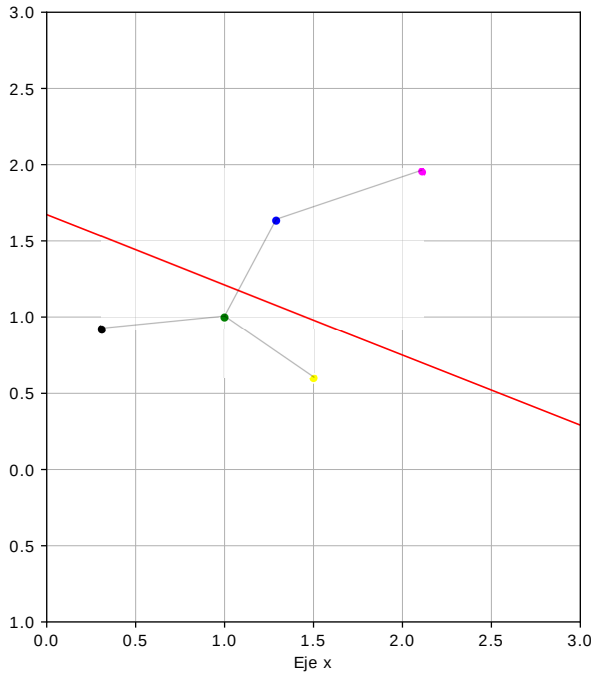
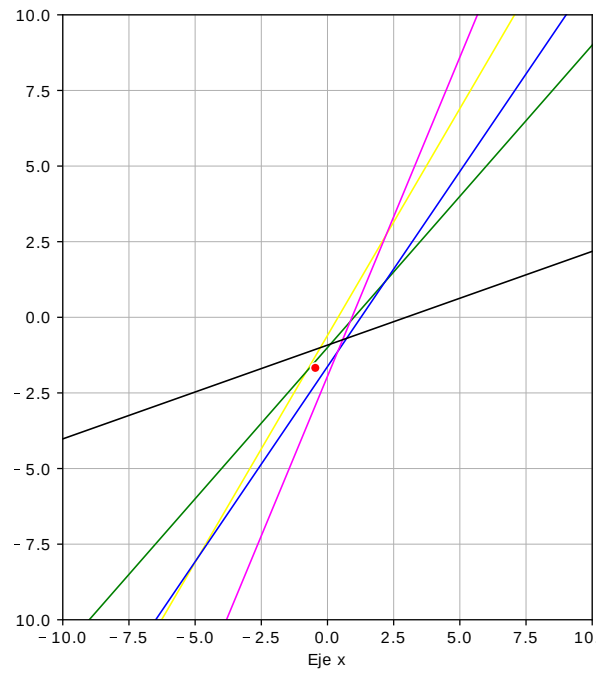


Figura 4.7: Ejemplo 1: Simulación del algoritmo en tiempo  $t + 1$ .

En la figura 4.8(a) se puede observar que los sensores azul y verde tienen en su área de cobertura al objeto móvil (ver línea roja  $l_i$ ). El sensor amarillo se puede desactivar ya que el punto  $*l_i$  está fuera del moño verde-amarillo. Lo que implica que el objeto móvil está fuera de la cobertura del nodo amarillo.



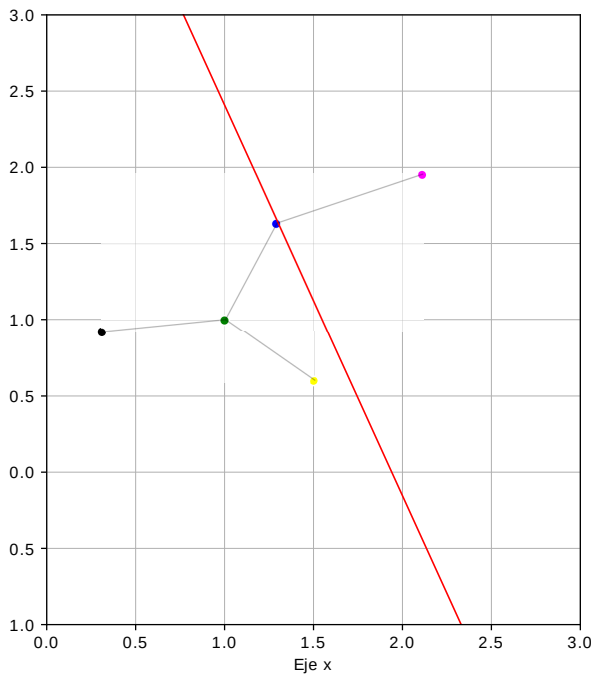
(a) Espacio primal.



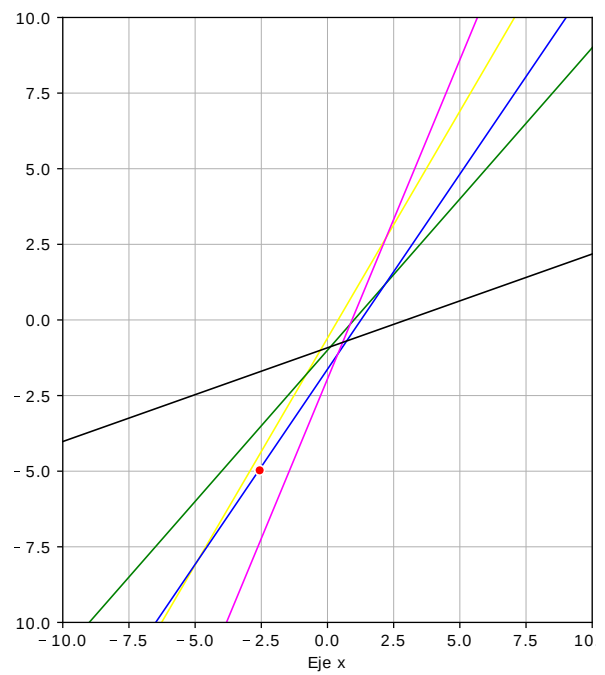
(b) Espacio dual.

Figura 4.8: Ejemplo 1: Simulación del algoritmo en tiempo  $t + 2$ .

Finalmente, en la figura 4.9(a) el objeto móvil se aproxima al sensor azul y se cumple la primera condición del algoritmo 1, donde se desactiva el sensor verde, se mantiene activo el sensor azul y se activa el sensor rosa. La línea roja se muestra perpendicular a los sensores rosa y azul.



(a) Espacio primal.



(b) Espacio dual.

Figura 4.9: Ejemplo 1: Simulación del algoritmo en tiempo  $t + 3$ .



### 4.3.2. Ejemplo 2

La figura 4.10 muestra una red de sensores, donde los sensores amarillo  $s_1$  y verde  $s_2$  detectan la presencia de un objeto móvil. La dirección del objeto móvil es representada por la línea roja  $l_i$ . En la figura 4.10(b) el punto  $*l_i$  se ubica en el moño que forman los sensores amarillo y verde. Y se ubica más próximo al sensor amarillo.

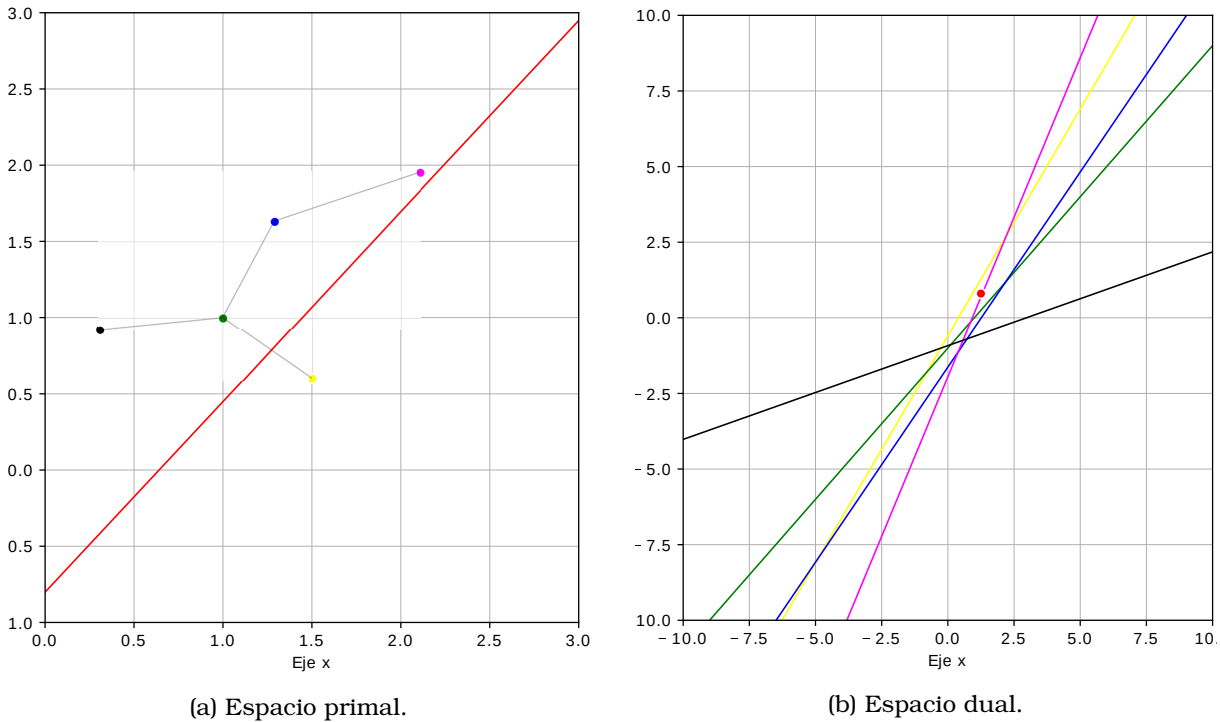
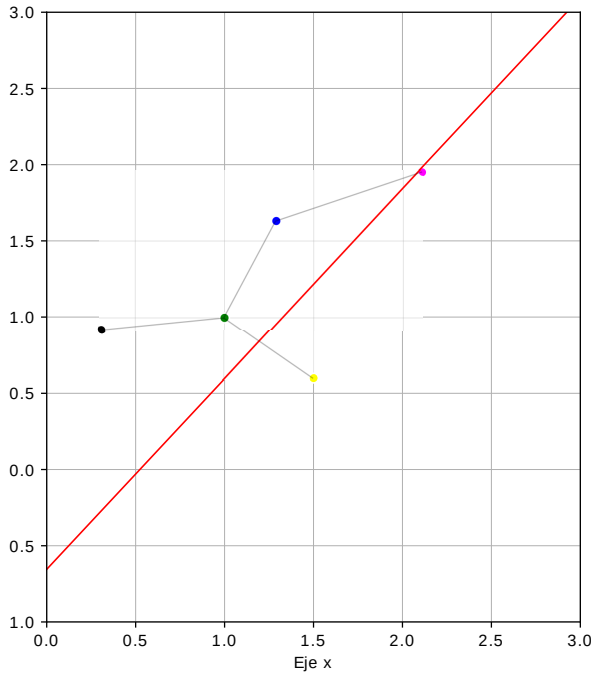


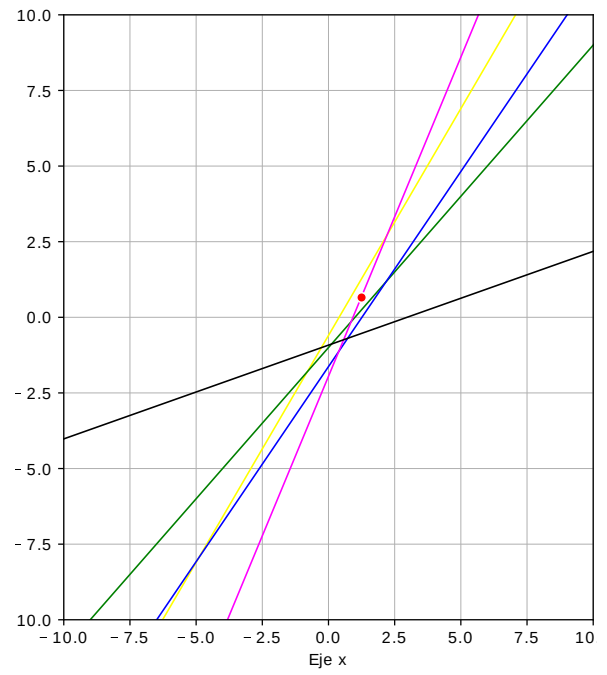
Figura 4.10: Ejemplo 2: Simulación del algoritmo en tiempo  $t$ .

En la figura 4.11(a) la línea roja se mueve hacia el punto verde. Se puede observar que el punto rojo en el plano dual se acerca a la línea verde, sin embargo no se encuentra tan cerca como para utilizar alguna de las dos primeras condiciones del algoritmo 1. Por lo tanto, se utiliza la tercera condición. Esta condición consiste en identificar dentro de que otros moños se encuentra el punto rojo. En este ejemplo, el punto rojo además de estar en el moño amarillo-verde, también está dentro del moño rosa-azul. Esto quiere decir que el objeto móvil se puede dirigir hacia la arista rosa-azul sin entrar en el área de cobertura de verde-azul.

En el espacio primal se observa que la línea roja atraviesa el enlace de los sensores rosa y azul, además del enlace verde y amarillo. Por lo tanto, los sensores amarillo y verde se desactivan y se activan los sensores azul y rosa.



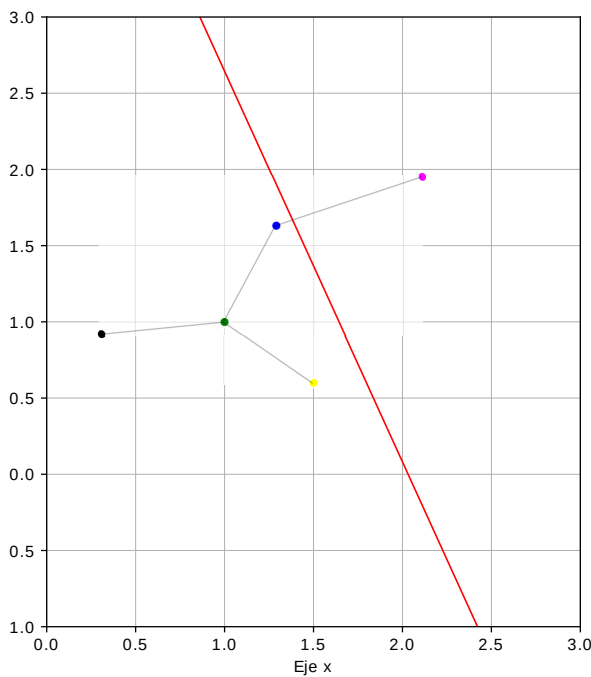
(a) Espacio primal.



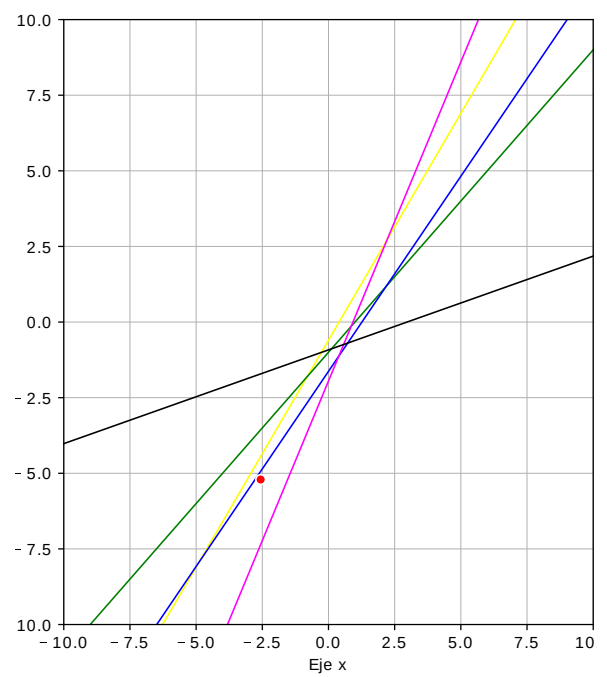
(b) Espacio dual.

Figura 4.11: Ejemplo 2: Simulación del algoritmo en tiempo  $t + 1$ .

Posteriormente, los sensores azul y rosa tienen en su área de cobertura al objeto móvil (ver figura 4.12(a)). En la figura 4.12(b), es fácil observar que si el punto rojo se aproxima a la línea rosa, éste saldrá del área de cobertura de la red y del moño azul-rosa.



(a) Espacio primal.



(b) Espacio dual.

Figura 4.12: Ejemplo 2: Simulación del algoritmo en tiempo  $t + 2$ .



# Capítulo 5

## Conclusiones

En este capítulo, se presentan las conclusiones generales, así como la verificación de la hipótesis y finalmente, una sección de perspectivas de investigación.

### 5.1. Conclusiones generales

En este trabajo se propone un algoritmo de minimización de energía utilizando las propiedades de la transformada al espacio dual, donde cada sensor se representa como un punto y el objeto móvil como una línea que indica su dirección de movimiento, ya que al ser detectado por solo dos sensores no es posible saber su ubicación exacta. Sin embargo, el mapeo al espacio dual permite identificar propiedades que no son visibles en el espacio primal y de esta manera simplificar el problema.

El algoritmo se divide en tres condiciones. Esto se debe a que al dividir el espacio dual en polígonos, el objeto móvil en el espacio dual ( $*l_i$ ) puede quedar ubicado en alguno de estos tres casos. Si el objeto móvil se acerca a alguno de los dos sensores que detectan su presencia caen en el primer o segundo caso. Mientras que si el objeto móvil se aleja de forma perpendicular a la línea que une a los sensores activos y no cae en ninguna de las dos primeras condiciones, se usa la tercera condición. Para esta última condición se requiere identificar en que otro nódo se encuentra el nódo móvil. Este proceso toma  $O(n)$ , mientras que las dos primeras condiciones toma  $O(1)$ .

Aunque muchos trabajos [5] [10] [11] [6], son capaces de predecir patrones de movimiento, estos requieren de tiempos largos de entrenamiento o de un conjunto grande de entradas para poder realizar su predicción. En cambio, nuestro algoritmo logra predecir que sensores deben activarse en tiempo lineal. Lo que lo hace atractivo para el uso de sensores alimentados por pequeñas baterías y pocos recursos de memoria, ya que solo debe almacenar su información a dos saltos y hacer una búsqueda lineal.

## 5.2. Verificación de la hipótesis

Retomando la hipótesis presentada en la sección 1.2:

*“Modelos basados en aprendizaje automatizado, minería de datos o diagramas de Voronoi requieren de un cómputo exhaustivo para seleccionar cuál sensor debe activarse o desactivarse. Por el contrario, modelos basados en la transformada al espacio dual permiten encontrar propiedades para predecir qué sensores deben activarse de manera muy simple y así reducir el consumo de energía de la red”.*

Para verificar la hipótesis previa, esta tesis se desarrolló en dos etapas. En la primera etapa se encontraron propiedades de la transformada al espacio dual. A partir de estas propiedades, en la segunda etapa se propuso un algoritmo para minimizar el tiempo en el que los sensores se mantienen activos. Nuestras simulaciones (ver Sección 4.3.1 y Sección 4.3.2) muestran que el punto  $*l_i$  puede caer en tres casos distintos. Por ejemplo, en la figura 4.7(b) se puede utilizar alguna de las dos primeras condiciones mostradas en la sección 4.2, ya que el objeto móvil se acerca a alguno de los dos sensores que lo detectan. Mientras que la figura 4.11(b), muestra la tercera condición donde el objeto móvil se encuentra dentro de dos o más moños a la vez. Las propiedades encontradas del espacio dual así como nuestras simulaciones nos permiten concluir que la hipótesis es verdadera debido a que la predicción es a lo más lineal y no se requieren periodos prolongados de entrenamiento o de un cómputo exhaustivo, el cual va en contra del ahorro de energía en una red de sensores IoT. Este es el caso de [6], donde la complejidad algorítmica es  $O(n^3)$  y además requiere de al menos 20 entradas para realizar una predicción correcta.

## 5.3. Perspectivas de investigación

En el capítulo 4 se realizó el desarrollo de la investigación de forma teórica, donde el área de cobertura de los sensores son circunferencias y se comunican de forma instantánea y correcta. Por esta razón como perspectiva de investigación se realizará una simulación aproximada a un escenario real en un software de simulación especializado, donde las áreas de cobertura y la comunicación entre los sensores este definida por modelos probabilísticos para así calcular la eficiencia energética, medir el tiempo en el que los sensores se mantienen activos y la precisión del algoritmo en un ambiente más cercano a la realidad.

Finalmente, se implementará el algoritmo en una red de sensores reales utilizando un estándar predefinido con el fin de medir el ahorro energético real.

# Bibliografia

- [1] Rathin Chandra Shit, Suraj Sharma, Deepak Puthal, and Albert Y. Zomaya, "Location of Things (LoT): A Review and Taxonomy of Sensors Localization in IoT Infrastructure". IEEE Communications Surveys & Tutorials, 2018.
- [2] Sachchidanand Singh, and Nirmala Singh, "Internet of Things(IoT): Security Challenges, Business Opportunities & Reference Architecture for E-commerce". 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), 2015.
- [3] Hyunsook Kim et al., "An energy efficient method for tracking mobile ubiquitous robots using wireless sensor network", IFIP International Conference on Personal Wireless Communications, 2006.
- [4] Yingqi Xu, Julian Winter, Dual prediction-based reporting for object tracking sensor networks, IEEE, 2004.
- [5] N. A. Ansari, U. A. Deshpande, A. M. Tapas, and A. A. Jejani A, "DISTRIBUTED ALGORITHM FOR OBJECT TRACKING IN WIRELESS SENSOR NETWORKS USING DATA MINING BASED PREDICTION", 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2017.
- [6] Lorenzo Paris and M. H. Anisi, "An Energy-efficient Predictive Model for Object Tracking Sensor Networks". 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), 2019.
- [7] Manel Abdelkader, Mohamed Hamdi, and Nouredine Boudriga, "Voronoi-based Sensor Network Engineering for Target Tracking Using Wireless Sensor Networks". IEEE 2008 New Technologies, Mobility and Security, 2008.
- [8] Qianqian Ren, Jinbao Li and Beibei Sun, "TTVT: A Two-Tier Voronoi Diagram based Tracking Algorithm in Wireless Sensor Networks". IEEE 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2018.
- [9] Hsu, Jenq-Muh, Chao-Chun Chen, and Chia-Chi Li. "POOT: An efficient object tracking strategy based on short-term optimistic predictions for face-structured sensor networks". Computers & Mathematics with Applications 63.2 (2012): 391-406.
- [10] S. Samarah, M. Al-Hajri, and A. Boukerche, "An Energy Efficient Prediction-based Technique for Tracking Moving Objects in WSNs". IEEE ICC 2011 proceedings, 2011.
- [11] Dr.Thangarajan, Prakash kumar Sakthivel, and Jai Balaji Padmanaban, "An Energy Efficient Technique For Object Tracking in Wireless Sensor Networks". IEEE 2013 International Conference on Communication Systems and Network Technologies, 2013.
- [12] Zhibo Wang, Hongbin Li, Xingfa Shen, Xice Sun, and Zhi Wang, "Tracking and Predicting Moving Targets in Hierarchical Sensor Networks". 2008 IEEE International Conference on Networking, Sensing and Control, 2008.
- [13] Khuong Vu and Rong Zheng, "Geometric Algorithms for Target Localization and Tracking Under Location Uncertainties in Wireless Sensor Networks". 2012 Proceedings IEEE INFOCOM, 2012.

- [14] Jie Liu, Feng Zhao, P. Cheung, and L. Guibas, "Apply geometric duality to energy-efficient non-local phenomenon awareness using sensor networks". IEEE Wireless Communications, 2004.
- [15] Shereen S. Ismail, Eman Alkhader, Said Elnaffar, "Object Tracking in Wireless Sensor Networks: Challenges and Solutions", JCS, vol. 12, no. 4, pp. 201-212, 2016.
- [16] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. (2008). Computational Geometry. Algorithms and Applications.. Berlin: Springer.
- [17] Matplotlib. (s.f.). Matplotlib: Python plotting — Matplotlib 3.1.1 documentation. Recuperado 1 enero, 2020, de <https://matplotlib.org/>
- [18] D. Matula and R. Sokal. "Properties of gabriel graphs relevant to geographic variation research and the clustering of points in the plane". Geographical Analysis, 12(3):205 - 222, July 1980.
- [19] Ke Wang, Liqiang Wang, Shiyu Cai, and Song Qu, "An Energy-Saving Algorithm of WSN Based on Gabriel Graph". 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing, 2009.
- [20] Petr Felkel. "DUALITY AND APPLICATIONS OF ARRANGEMENTS". Katedra Pocatocove Grafiky A Interlace, 2017.
- [21] Lorenzo Paris, Mohammad Hossein Anisi, "An Energy-efficient Predictive Model for Object Tracking Sensor Networks". 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), 2019.
- [22] Santosh Pandey, Prathima Agrawal, "A SURVEY ON LOCALIZATION TECHNIQUES FOR WIRELESS NETWORKS". Journal of the Chinese Institute of Engineers, Vol. 29, No. 7, pp. 1125-1148, 2006.
- [23] N. Fathima, A. Ahammed, Rajashekarappa, R. Banu, Parameshachari, N. Naik, "Optimized Neighbor Discovery in Internet of Things (IoT)". 2017 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEEC-COT), 2017.