

# JUEGOS DINAMICOS DE PERSECUCION Y EVASION

por

Ollantay Medina Huaman

Tesis sometida en cumplimiento parcial  
de los requerimientos para el grado de

MAESTRO EN CIENCIAS

en

COMPUTACION CIENTIFICA

UNIVERSIDAD DE PUERTO RICO  
RECINTO UNIVERSITARIO DE MAYAGUEZ

2007

Aprobada por:

---

Robert Acar, Ph.D.  
Miembro, Comité Graduado

---

Fecha

---

Paul Castillo, Ph.D.  
Miembro, Comité Graduado

---

Fecha

---

Pedro Vasquez, D.Sc.  
Presidente, Comité Graduado

---

Fecha

---

Wilson Rivera, Ph.D.  
Representante de Estudios Graduados

---

Fecha

---

Julio Quintana, Ph.D.  
Director del Departamento

---

Fecha

# **JUEGOS DINAMICOS DE PERSECUCION Y EVASION**

Copyright 2007

por

Ollantay Medina Huaman

## **Abstract**

Pursuit-Evasion games settle the foundations for Dynamic Games, a very active area in these days due to its great applicative potential. The Hamstrung Squad Car and the Homicidal Chauffeur are two zero-sum, two-player Pursuit-Evasion games with complete information, that this work studies in order to formulate and implement a numeric solution based on Dynamic Programming for this kind of games. The resulting algorithms are implemented using Matlab and have complexity of polynomial order, depending directly on the dimension of the problem and the number of directions that players can use. These algorithms can be reused to solve similar problems.

## Resumen

Los juegos de Persecución y Evasión sentaron las bases de los Juegos Dinámicos, un área muy activa en estos tiempos debido a su gran potencial aplicativo. La Patrulla Restringida y el Chofer Homicida son dos juegos de Persecución y Evasión de suma cero de dos jugadores con información completa, que se estudian en el presente trabajo para poder plantear e implementar una solución numérica basada en Programación Dinámica para este tipo de juegos. Los algoritmos resultantes son implementados usando Matlab y son de complejidad de orden polinómico, dependiente directamente de la dimensión del problema y el número de direcciones que los jugadores pueden usar. Estos algoritmos pueden ser reusados para resolver problemas similares.

A Karen y Sofia luceros de alegría y esperanza.

A mis padres y hermanas.

A los amigos que saben ser hermanos.

## **Agradecimientos**

Quisiera expresar mi más profundo agradecimiento a todas aquellas personas que en algún momento me brindaron apoyo, a veces tan admirable y oportuno.

También quisiera agradecer a todos los profesores que he conocido durante la maestría, me han hecho querer ser un mejor profesional, una mejor persona.

# Tabla de Contenidos

<b>Lista de Figuras</b>	<b>iv</b>
<b>Lista de Tablas</b>	<b>v</b>
<b>1 Introducción</b>	<b>1</b>
<b>2 Conceptos Previos</b>	<b>5</b>
2.1 Teoría de Juegos . . . . .	5
2.1.1 Conceptos . . . . .	5
2.1.2 Análisis del juego . . . . .	10
2.2 Programación Dinámica . . . . .	12
2.2.1 Conceptos . . . . .	12
2.2.2 Principio de Optimalidad . . . . .	13
2.3 Juegos Dinámicos . . . . .	14
<b>3 Formulación y Solución de dos Juegos de Persecución y Evasión</b>	<b>17</b>
3.1 Juegos de Persecución y Evasión . . . . .	17
3.1.1 Variables de estado . . . . .	18
3.1.2 Variables de control . . . . .	18
3.1.3 Zona de captura . . . . .	19
3.1.4 Vectogramas . . . . .	19
3.1.5 Espacio realista y espacio reducido . . . . .	20
3.2 La Patrulla Restringida . . . . .	21
3.2.1 Reglas . . . . .	21
3.3 El Chofer Homicida . . . . .	23
3.3.1 Formulación del Modelo Matemático . . . . .	24
3.3.2 Entendiendo las estrategias óptimas . . . . .	27
3.3.3 Curvas singulares . . . . .	29
3.3.4 Curvas de Barrera de $V(x)$ . . . . .	30
3.4 Solución . . . . .	32
3.4.1 Principio Inductivo para el cálculo del Valor del Juego . . . . .	34
3.4.2 La Patrulla Restringida . . . . .	35

3.4.3	El Chofer Homicida discreto . . . . .	38
3.5	Algoritmos . . . . .	44
3.5.1	Consideraciones . . . . .	44
3.5.2	Valor del Juego $V(x)$ . . . . .	45
3.5.3	Trayectorias óptimas . . . . .	47
<b>4</b>	<b>Resultados</b>	<b>49</b>
4.1	La Patrulla restringida . . . . .	49
4.2	El Chofer Homicida . . . . .	51
4.2.1	Dinámica real . . . . .	52
4.2.2	Dinámica discreta . . . . .	52
<b>5</b>	<b>Conclusiones y Trabajo Futuro</b>	<b>62</b>
5.1	Conclusiones . . . . .	62
5.2	Trabajo Futuro . . . . .	64
	<b>Bibliografía</b>	<b>65</b>
<b>A</b>	<b>Código Fuente</b>	<b>67</b>
A.1	Generación de los Vectogramas . . . . .	67
A.2	Comportamiento de los vectogramas en el espacio reducido . . . . .	68
A.3	Cálculo de Border-E y Border-P . . . . .	70
A.4	Cálculo del Valor del Juego . . . . .	71
A.5	Cálculo de Trayectorias Óptimas . . . . .	72
A.6	Utilitarios . . . . .	76



# Lista de Figuras

3.1	Espacio Reducido de la Patrulla Restringida . . . . .	22
3.2	Espacio Realista del Chofer Homicida . . . . .	24
3.3	Espacio Reducido del Chofer Homicida . . . . .	26
3.4	Primer Escenario de Persecución - Maniobra Simple . . . . .	27
3.5	Segundo escenario de Persecución - Maniobra Compleja . . . . .	28
3.6	Curvas Singulares y Trayectorias Optimas del Chofer Homicida en el Espacio Reducido . . . . .	29
3.7	Vectogramas del Chofer Homicida . . . . .	38
3.8	Error en Magnitud y Angulo de los vectores de la malla . . . . .	41
4.1	PR - 1: Trayectoria Optima . . . . .	50
4.2	PR - 2: Trayectoria Optima . . . . .	51
4.3	Chofer Homicida - $V(x)$ . . . . .	53
4.4	Chofer Homicida - Barrera teórica y Barrera numérica . . . . .	54
4.5	CH 3: Trayectoria Optima en el Espacio Realista . . . . .	56
4.6	CH 3: Trayectoria Optima en el Espacio Reducido . . . . .	57
4.7	CH 4: Trayectoria Optima en el Espacio Realista . . . . .	57
4.8	CH 4: Trayectoria Optima en el Espacio Reducido . . . . .	58
4.9	CH 5: Trayectoria Optima en el Espacio Realista . . . . .	58
4.10	CH 5: Trayectoria Optima en el Espacio Reducido . . . . .	59

# Lista de Tablas

2.1	Matriz de utilidades con punto de silla . . . . .	10
2.2	Matriz de utilidades sin punto de silla . . . . .	11
3.1	P-vectograma . . . . .	35
3.2	E-vectograma . . . . .	36
3.3	Valor del juego $V(x)$ . Origen en $(0)$ . . . . .	36
4.1	Dinámica Real . . . . .	52
4.2	Dinámica Discreta . . . . .	52

# Capítulo 1

## Introducción

En la década de 1950 Rufus Isaacs inició el desarrollo de una nueva teoría denominada inicialmente como juegos diferenciales y que actualmente también es conocida como juegos dinámicos. La necesidad que originó tal investigación tuvo motivos militares, por lo que la investigación de Isaacs no fue pública sino hasta 1965.

Isaacs orientó sus esfuerzos hacia una clase de problemas específicos, denominados juegos dinámicos de persecución y evasión de suma cero de dos jugadores; a partir de la cual se sentaron las bases de los juegos dinámicos como tales. Los problemas contemplados tratan sobre toma de decisiones en entornos dinámicos competitivos que dependen del tiempo, por tal motivo se vio la necesidad de recurrir a la teoría de juegos y la teoría de optimización para poder afrontar adecuadamente problemas de este tipo. Un amplio panorama histórico-técnico del desarrollo de la teoría de juegos diferenciales puede ser encontrado en [12].

En un caso típico de persecución y evasión, un depredador acecha, acorrala y captura a su presa; un avión caza persigue y trata de fijar a un bombardero enemigo en

el área crítica que le permitirá abrir fuego y eliminarlo; o un bombardero evadirá el fuego defensivo de una batería antiaérea hasta acercarse lo suficiente a un objetivo preestablecido para liberar su carga mortal. El comportamiento que siguen los actores de estos escenarios, su viabilidad u optimalidad tratan de ser develados por la teoría de juegos dinámicos, solo se puede tener por seguro que cada uno dará lo mejor de sí mismo para prevalecer.

Los juegos dinámicos son usualmente difíciles de formular y resolver por su naturaleza dinámica y aleatoria. Se dan en espacios tanto continuos como discretos, con tiempo tanto continuo como discreto. Como consecuencia los agentes que persiguen o evaden deben mantener una buena coordinación motor sensorial la cual es compleja con respecto tanto al medio físico como al comportamiento antagónico del oponente. Además, los juegos de persecución y evasión también requieren respuesta en tiempo real, así como control dinámico para enfrentar y explotar el retraso, incertidumbre o error del adversario. Los sistemas de control natural o artificial que son lentos o fácilmente confundidos no pueden sobrevivir a un escenario de larga duración en un contexto de persecución y evasión, razón por la cual los métodos tradicionales de inteligencia artificial no pueden ser muy eficientes como métodos de solución en este tipo de juegos.

Los juegos de persecución y evasión son científicamente interesantes, debido a que los agentes envueltos en un escenario de uno contra el otro se desarrollan en forma continua a través del tiempo; además, efectiva persecución requiere predicción y en algunos casos "leer la mente del adversario", mientras que el evasor requiere del uso de tácticas que deben ser impredecibles o engañosas. El estudio de los comportamientos de persecución y evasión tiene muchas implicaciones científicas y aplicaciones prácticas que han recibido especial

atención en otras áreas de la ciencia.

En biología, donde el comportamiento depredador-presa, el delicado equilibrio que existe entre estos, o incluso las respuestas evolutivas de las especies con el fin de adaptarse y perpetuarse son aspectos que ahora tratan de ser vistos a través de los ojos de los juegos dinámicos.

En robótica, donde la persecución y la evasión están entre los más importantes y desafiantes problemas que se deben afrontar, la popular escena de pruebas del futbol robótico es un lugar donde los juegos dinámicos en general se constituyen como una técnica más robusta en comparación con otros métodos tradicionales de inteligencia artificial.

En Economía, donde las cada vez más complejas situaciones con alta dependencia del tiempo son inmanejables mediante pura teoría de juegos.

En esta tesis se analiza e implementa un algoritmo para resolver numéricamente dos problemas de persecución y evasión, la Patrulla Restringida y el Chofer Homicida, con el propósito y la esperanza de que el trabajo resultante se pueda reusar en problemas similares.

La tesis consta de 5 capítulos, en este primer capítulo se hace una introducción a los juegos dinámicos.

En el capítulo 2 se incluyen definiciones básicas de teoría de juegos, programación dinámica y juegos dinámicos; que serán importantes en la formulación y solución del problema.

El capítulo 3 especifica algunos conceptos pertinentes a juegos de persecución y evasión, para luego presentar la formulación y solución de dos juegos dinámicos de persecución y evasión; primeramente se trata la Patrulla Restringida, problema sencillo resuelto

para mostrar la mecánica de trabajo. Luego se aborda el Chofer Homicida, se plantea y explica su solución a través de programación dinámica, el algoritmo a usar y la complejidad de éste.

El capítulo 4 establece los criterios para evaluar la solución numérica obtenida para el Chofer Homicida con valores específicos.

Finalmente, el capítulo 5 presenta las conclusiones y trabajo futuro relacionado con esta tesis.

## Capítulo 2

# Conceptos Previos

### 2.1 Teoría de Juegos

Los juegos dinámicos de persecución y evasión que serán tratados en esta tesis pertenecen a los juegos no cooperativos de suma cero de dos jugadores con información completa, esta sección está enfocada desde ese punto de vista.

#### 2.1.1 Conceptos

**Juego**, es un conjunto de reglas que especifican claramente:

- Lo que un jugador puede o tiene que hacer bajo todas las circunstancias posibles.
- La cantidad de información que cada jugador recibe.
- El momento en que el juego termina.
- La cantidad que cada jugador paga o recibe al terminar el juego.

- El objetivo de cada jugador durante el juego.

**Jugador**, un agente que hace decisiones en un juego.

**Movida**, o decisión, es una acción del jugador en concordancia con las reglas del juego.

**Estrategia**, es un conjunto de movidas que especifican como jugar un juego dado de principio a fin. Este plan de movidas se puede establecer de antemano y debe cubrir todas las posibles contingencias que puedan surgir durante el juego, además de incorporar la información que pueda llegar a estar disponible según las reglas de juego. Todas las formas diferentes en las que un jugador pueda actuar de principio a fin de un juego constituirán todas sus posibles estrategias; obviamente algunas de éstas serán superiores a otras, pero todas son estrategias. Cada par de estrategias, consistente de una estrategia por cada jugador determina una forma de jugar un juego.

**Estrategia dominante**, una estrategia domina a otra si siempre da una mejor utilidad al jugador, sin importar que haga el oponente.

**Utilidad**, función de valor real establecida para cada jugador que refleja la deseabilidad de los posibles resultados de un juego.

**Valor del Juego**, es el vector de utilidades obtenidas por cada jugador cuando actúa óptimamente.

**Información Completa**, un juego es de información completa cuando cada jugador conoce exactamente el estado suyo y el del oponente en cada momento del juego. El ajedrez es un ejemplo de un juego de información completa, mientras que el poker no.



**Forma normal de un juego**, llamada también forma estratégica, suponga que el jugador A tiene  $m$  estrategias posibles y el jugador B tiene  $n$  estrategias posibles, la forma normal del juego está dada por la matriz  $P = (a_{ij}, b_{ij})$ , de dimensiones  $m \times n$ , donde  $(a_{ij}, b_{ij})$  es un par ordenado, con  $a_{ij}$  la utilidad del jugador A y  $b_{ij}$  la utilidad del jugador B, resultantes de haber usado la  $i$ -ésima y  $j$ -ésima estrategias de A y B respectivamente.

**Suma cero**, un juego de dos jugadores es de suma cero, si la matriz de la forma normal del juego se puede representar simplemente por  $P = (a_{ij})$ , con  $a_{ij}$  la utilidad del jugador A y  $-a_{ij}$  la utilidad del jugador B. Cada jugador buscará hacer su utilidad tan grande como sea posible o dejar la utilidad del oponente tan pequeña como sea posible. En este caso el Valor del juego puede ser considerado simplemente como un escalar.

En adelante los conceptos se orientan específicamente a juegos de suma cero.

**Punto de silla**, la forma normal de un juego siempre cumple:

$$\max_{i \leq m} \min_{j \leq n} a_{ij} \leq \min_{j \leq n} \max_{i \leq m} a_{ij} \quad (2.1)$$

El lado izquierdo de 2.1 representa la utilidad que puede asegurar A en tanto que el lado derecho representa la utilidad que puede asegurar B. En el caso que 2.1 se convierta en una igualdad

$$\max_{i \leq m} \min_{j \leq n} a_{ij} = \min_{j \leq n} \max_{i \leq m} a_{ij} = v \quad (2.2)$$

Se dice que el juego tiene punto de silla y que  $v$  es el valor del juego. El valor de  $v$  es único aunque puede presentarse en más de una posición de la matriz de utilidades.

La existencia del punto de silla implica que existen estrategias óptimas  $i^*$  y  $j^*$  tal que:

$$a_{ij^*} \leq a_{i^*j^*} \leq a_{i^*j} \quad (2.3)$$

$$a_{i^*j^*} = v \quad (2.4)$$

Estas estrategias óptimas tienen la propiedad de que si el jugador A juega su estrategia  $i^*$  no importa lo que haga el jugador B, el jugador A siempre ganará al menos  $v$ , similarmente para el jugador B que al jugar su estrategia  $j^*$  siempre perderá a lo más  $v$ . Estas estrategias podrían ser anunciadas de antemano y ninguno de los jugadores podría tomar ventaja para cambiar el valor del juego  $v$ .

Una forma equivalente de encontrar el punto de silla en la matriz de utilidades es buscar un elemento que sea simultáneamente el mínimo de su fila y el máximo de su columna. Si más de un elemento cumple esta condición los pares de estrategias que determinan son equivalentemente óptimos.

**Estrategia pura**, se refiere al hecho de seleccionar una estrategia específica del conjunto de todas las estrategias posibles para jugar un juego. En otras palabras, define una forma de juego específica para un jugador. Si el punto de silla existe para un juego, la estrategia óptima es una estrategia pura.

Una estrategia pura para un jugador con  $m$  estrategias disponibles puede ser representada por un vector de dimensión  $m$  cuyas componentes son todas ceros excepto una (la estrategia seleccionada).

**Estrategia mixta**, cuando al conjunto de estrategias de un jugador se le aplica una distribución de probabilidad (conjunto de pesos) que corresponde a cuan frecuentemente cada estrategia es usada si el juego se repite varias veces. Un jugador A opta por usar una estrategia mixta cuando es deseable mantener al oponente B adivinando la movida que A hará, es decir, cuando el oponente B puede beneficiarse al saber la siguiente movida de A, A opta por jugar aleatoriamente.

Una estrategia mixta para un jugador con  $m$  estrategias disponibles puede ser representada por un vector de dimensión  $m$  cuyas componentes varían entre 0 y 1; además que la suma de todas las componentes es 1. Note que una estrategia pura puede ser considerada como un caso particular de una estrategia mixta.

**Teorema Minimax**, el teorema fundamental de la teoría de juegos establece que, cada juego finito de suma cero de dos jugadores tiene estrategias óptimas mixtas.

Expresable como, sean  $X$  y  $Y$  las estrategias mixtas de los jugadores A y B respectivamente; y sea  $P$  la matriz de utilidades del juego. Entonces

$$\max_X \min_Y X^T P Y = \min_Y \max_X X^T P Y = v \quad (2.5)$$

Donde:

$v$  es llamado el valor del juego

$X$  y  $Y$  son llamadas las soluciones (estrategias óptimas) del juego.

### 2.1.2 Análisis del juego

Una vez definidos los conceptos necesarios, las ideas para el análisis y solución de un juego de suma cero de dos jugadores pueden ser resumidas en los siguientes pasos:

- Determinar las estrategias posibles para los jugadores.
- Llevar el juego a su forma normal produciendo la matriz de utilidades.
- Usar el Teorema Minimax para encontrar el valor del juego, así como las soluciones (estrategias óptimas).

Se propone un juego en forma normal para ilustrar mejor la determinación del punto de silla. Dada la matriz de utilidades del juego de suma cero, con información completa, entre los jugadores A y B:

		Jugador B	
		IV	V
Jugador A	I	8	(5)
	II	0	3
	III	-1	2

Tabla 2.1: Matriz de utilidades con punto de silla

El jugador A es el jugador maximizante, de lo mínimo que puede ganar por cada estrategia  $\{5, 0, -1\}$  debe escoger el mayor de estos valores  $\{5\}$  que corresponde a la estrategia I. El jugador B es el jugador minimizante, de lo máximo que puede perder por cada estrategia  $\{8, 5\}$  debe escoger el menor de estos valores  $\{5\}$  que corresponde a la estrategia V. Como ambos valores son iguales el punto de silla existe y el valor del juego es 5; las estrategias óptimas son A-I y B-V.

Se propone un juego sencillo en forma normal para ilustrar el uso de estrategias mixtas.

		Jugador B	
		III	IV
Jugador A	I	1	-1
	II	-1	1

Tabla 2.2: Matriz de utilidades sin punto de silla

En este caso no existe punto de silla, note como el jugador que juegue primero estará en desventaja, o aquel del que se sepa su estrategia óptima también estará en desventaja ya que el que juegue oponente siempre tiene una estrategia que le permite llevar la utilidad a 1 o -1 según le convenga. Es por este motivo por lo que cada jugador debe escoger una distribución de probabilidad para sus estrategias. Si A decide jugar sus estrategias cada una con una probabilidad de 0.5 tendrá un valor esperado de 0. Lo mismo ocurre para B que si juega sus estrategias cada una con una probabilidad de 0.5 tendrá un valor esperado de 0. Como ambos valores son iguales, el valor del juego existe y es 0. La estrategia mixta óptima para A es  $(0.5, 0.5)$  y para B es  $(0.5, 0.5)$ .

Cabe señalar que en algunos casos donde se presentan juegos complejos, se procede a descomponerlos en subjuegos más sencillos los cuales son resueltos uno a uno siguiendo un análisis como el anterior, los resultados van generando las respectivas estrategias óptimas completas para los jugadores.

En cuanto al teorema minimax hay varias pruebas, algunas están relacionadas con teoremas de punto fijo, otras basadas en la separación de conjuntos convexos, casi todas estas pruebas son solo de existencia de estrategias óptimas pero no son constructivas desde

el punto de vista práctico. Explicaciones más detalladas concernientes a teoría de juegos, conceptos complementarios y ejemplos pueden ser encontrados en [3] y [4].

## 2.2 Programación Dinámica

La Programación Dinámica es un enfoque particular de optimización. Entendiéndose por optimización el encontrar la mejor solución para un problema de un conjunto de alternativas. Esta poderosa técnica de resolución de problemas fue desarrollada por Richard Bellman [5] a finales de la década de los 50's.

### 2.2.1 Conceptos

**Estado** el espacio de estados es un conjunto no vacío  $X$  cuyos elementos  $x \in X$  son combinaciones específicas de los valores permitidos para las variables que describen la condición del sistema o proceso bajo estudio.

**Etapas** indica el orden en que ocurren las decisiones. El indicador de etapa se puede considerar como discreto o continuo, dependiendo si el enfoque de programación dinámica va a ser trabajado en tiempo discreto o continuo.

**Decisión** se toma sobre cada estado  $x \in X$ , es decir, existe un conjunto no vacío  $U_x$  llamado el conjunto decisión para  $x$ , tal que todo elemento (variable de decisión)  $u(x) \in U_x$  representa una de las opciones que el controlador puede tomar cuando el sistema se encuentra en el estado  $x \in X$ .

**Transiciones** asumiendo que el proceso bajo estudio se encuentra en el estado  $x_1 \in X$ , y se toma una decisión  $u(x_1) \in U_{x_1}$ , entonces el proceso se moverá a otro estado

$x_2 \in T(x_1, u(x_1))$  donde  $T(x, u(x))$  es el conjunto de todos los posibles estados a los que se puede mover el proceso partiendo del estado  $x \in X$  y usando la decisión  $u(x) \in U_x$ . El operador  $T(x, u(x))$  puede ser determinístico o aleatorio.

**Política** es definida como el conjunto de todos los conjuntos de decisión  $U_x$  que corresponden a todas las variables de estado  $x \in X$ .

**Costo** es la función de valor real,  $c_u(x)$ , definida como el valor que debe ser obtenido si el proceso está en el estado  $x \in X$  y se toma una decisión  $u(x) \in U_x$ .

Una idea clave en la resolución de problemas de optimización sobre tiempo es que a menudo se puede considerar una optimización sobre etapas. Si se habla en el sentido de minimizar costos, se puede tomar un curso de acciones en una etapa adecuada para obtener su menor costo, la implicancia de este resultado en etapas posteriores es que la suma de costos o costo total también está siendo minimizada al tener el menor costo posible en la etapa actual. En otras palabras, la elección óptima parcial contribuye a una elección óptima global, este concepto queda expresado en el Principio de Optimalidad.

## 2.2.2 Principio de Optimalidad

*Desde cualquier punto  $x$  de una trayectoria óptima, la trayectoria restante también es óptima para el problema correspondiente empezando ahora desde  $x$ .*

Otras definiciones pertinentes:

Para poder usar el criterio de optimalidad que se ha indicado, se requiere que el problema tenga las siguientes propiedades:

1. *Separabilidad de la función objetivo*, significa que,  $\forall k$ , el efecto de la etapa final  $k$  de

un proceso de  $n$  etapas sobre la función objetivo solo depende del estado  $x_{n-k}$  y de las últimas  $k$  decisiones.

2. *Separación de estados*, significa que,  $\forall k$ , la transición del estado  $x_k$  al estado  $x_{k+1}$  sólo depende del estado  $x_k$  y la decisión  $u_{k+1}(x_k)$  pero no de los estados anteriores. Esta propiedad es llamada la propiedad de estado Markoviana y los sistemas que tienen este tipo de propiedades son llamados, sistemas sin memoria.

Las bases formales de esta teoría pueden ser revisadas en [5], aplicaciones de programación dinámica en el área de control óptimo pueden encontrarse en [6].

## 2.3 Juegos Dinámicos

Se presentará una formalización general sobre Juegos dinámicos, cuyas ideas estarán centradas en juegos de dos jugadores con información completa.

El estado del juego en un tiempo  $t$  está dado por el vector  $x(t) \in R^n$  que está determinado por un sistema de ecuaciones diferenciales:

$$\frac{dx}{dt} = f(t, x, u(t), v(t)) \quad (2.6)$$

$$\text{Cond. Inicial} \quad : \quad x(t_0) = x_0$$

Donde:

$u(t)$  es controlado por el jugador A en cada tiempo  $t$  y  $v(t)$  es controlado por el jugador B en cada tiempo  $t$ , además de estar restringidos a  $u(t) \in Y$  y  $v(t) \in Z$ ; con  $Y$  y  $Z$  como conjuntos preestablecidos en el espacio euclidiano.



La elección de  $u(t)$  es dictada por un conjunto de reglas o estrategia  $U$  seleccionada por el jugador A antes de empezar el juego, similarmente la elección de  $v(t)$  es dictada por la estrategia  $V$  seleccionada por el jugador B antes de empezar a jugar.

El juego inicia acciones a partir del punto inicial  $(t_0, x_0)$ , continuará hasta el punto  $(t, \phi(t))$ , con  $\phi$  solución de la ecuación diferencial, en el cual se alcanza el conjunto terminal preestablecido  $T$ . El punto  $(t, \phi(t))$  en el cual se alcanza  $T$  es llamado punto terminal y será denotado como  $(t_f, x_f)$ .

La utilidad se define de la forma:

$$P(t_0, x_0; U, V) = g(t_f, x_f) + \int_{t_0}^{t_f} f^0(s, \phi(s), u(s), v(s)) ds \quad (2.7)$$

Donde:

$g$  y  $f^0$  son funciones de valor real preestablecidas.  $g$  determina la utilidad que se agrega al terminar el juego.

El jugador A busca escoger un conjunto de decisiones  $U$  de tal forma que  $P$  se maximice mientras que el jugador B busca un conjunto de decisiones  $V$  que minimice  $P$ .

El valor del juego, será definido como  $W(t_0, x_0)$ , que puede ser calculado (y siempre que se cumpla) mediante:

$$\sup_U \inf_V P(t_0, x_0; U, V) = \inf_V \sup_U P(t_0, x_0; U, V) \quad (2.8)$$

Se dice que un par de estrategias  $(U^*, V^*)$  constituyen un punto de silla, si para todo  $U$  sobre el conjunto donde se toma el supremo y para todo  $V$  sobre el conjunto donde se toma el ínfimo se cumple:

$$P(t_0, x_0; U, V^*) \leq P(t_0, x_0; U^*, V^*) \leq P(t_0, x_0; U^*, V) \quad (2.9)$$

Las estrategias  $U^*$  y  $V^*$  son denominadas estrategias óptimas, y se concluye que si existe un punto de silla entonces el valor del juego es

$$W(t_0, x_0) = P(t_0, x_0; U^*, V^*) \quad (2.10)$$

La solución del juego busca encontrar  $U^*$  y  $V^*$ , para ello se impone la *condición de Isaacs*.

Sea  $u(t) = y$ ,  $v(t) = z$  y  $\langle \cdot \rangle$  el producto interior de vectores; para cada punto inicial  $(t, x)$  y cada  $s \in R^n$  se debe cumplir:

$$\min_{y \in Y} \max_{z \in Z} \langle s, f(t, x, y, z) \rangle = \max_{z \in Z} \min_{y \in Y} \langle s, f(t, x, y, z) \rangle \quad (2.11)$$

La función  $W$  es entonces una solución de:

$$\max_{y \in Y} \min_{z \in Z} [W_t(t, x) + \langle W_x(t, x), f(t, x, y, z) \rangle] = 0 \quad (2.12)$$

$$W(t_f, x_f) = g(t_f, x_f)$$

Para todos los puntos iniciales  $(t, x)$  y puntos  $(t_f, x_f)$  en la frontera de  $T$ .

Para una explicación más detallada de la deducción de la solución ver [2].

## Capítulo 3

# Formulación y Solución de dos Juegos de Persecución y Evasión

Se establecerá de forma breve algunos conceptos específicos a juegos de persecución y evasión, luego se introducen los dos juegos específicos a ser tratados, la Patrulla Restringida y el Chofer Homicida. En el primer caso se describe el juego usando simplemente un conjunto de reglas, en el segundo caso se hará una formulación matemática más completa, detallando algunas características importantes que deben ser consideradas al resolver el problema, finalmente estos juegos son resueltos numéricamente.

### 3.1 Juegos de Persecución y Evasión

Diferentes situaciones de persecución pueden ser encontradas en el campo militar, buque-torpedo, caza-misil, bombardero-misil, etc., para una mejor especificación se tomará la abreviatura P para referirse al Perseguidor y E para referirse al Evasor, P y E serán

considerados como individuos, es decir, no más de un perseguidor ni de un evasor.

P y E representan los jugadores enfrentados en el juego, la persecución tendrá como escenario el plano euclidiano, en general un espacio  $n$ -dimensional y el objetivo del juego es que a partir de posiciones iniciales para P y E se logre una *captura* la cual consiste en hacer que la distancia PE sea menor que una cierta cantidad preestablecida. P buscará la captura lo mas pronto posible mientras que E tratará de posponerla indefinidamente.

Algunos conceptos importantes:

### 3.1.1 Variables de estado

Se denomina así al vector  $x \in R^n$  con componentes  $x_1, x_2, \dots, x_n$  las cuales describen de forma completa el estado de los jugadores, en nuestro caso el estado de P y E. Si hipotéticamente un juego fuera detenido en un momento dado, solo se necesita guardar el valor de las variables de estado para poder reanudarlo posteriormente y continuar el juego como si no hubiera habido interrupción alguna.

### 3.1.2 Variables de control

Se denomina así a  $u$  y  $v$ , en general vectores de dimensiones menores que la del espacio de juego, que describen las decisiones que toman P y E respectivamente en un momento  $t$  del juego, cada jugador tiene en todo momento independencia completa de sus controles respectivos. La transición de un estado del juego a otro depende completamente de los  $u$  y  $v$  escogidos por P y E respectivamente.

### 3.1.3 Zona de captura

En un juego de persecución P busca lograr la captura de E lo antes posible; de manera general esta captura se entiende como hacer que la distancia PE sea menor que una cantidad preestablecida. Se opta por el término *zona de captura* en vez de simplemente captura, para definir de forma más específica una región cerrada fijada a P (P esta en el interior de esta región), de manera que el objetivo de P es hacer que E quede dentro de esta zona lo antes posible. La zona de captura cumple el papel del conjunto terminal preestablecido  $T$  del planteamiento general de un juego dinámico.

### 3.1.4 Vectogramas

El estado del juego definido por 2.6 puede ser representado tambien de la forma

$$\frac{dx_i}{dt} = f_i(x, u, v) \quad i = 1, \dots, n \quad (3.1)$$

Considerando a  $x$  y  $v$  fijos en el espacio de juego, el conjunto de vectores  $f_i(x, u, v)$  para todo  $u$  aceptable será denominado vectograma o P-vectograma; análogamente se ve que existe un E-vectograma.

Estos vectogramas constituyen el conjunto de todas las direcciones posibles que pueden seguir tanto P como E, según las consideraciones anteriores son continuos; posteriormente se tomarán simplificaciones discretas para intentar resolver de forma aproximada juegos de persecución y evasión..

### 3.1.5 Espacio realista y espacio reducido

Al modelar un juego de persecución y evasión a partir de un prototipo físico, el número de variables de estado a menudo será tal que éstas pueden dar una descripción directa y completa de la situación. Este número de variables dará la dimensión del espacio realista  $n$ ; no obstante, puede resultar que se tienen más variables de las necesarias. Es aquí que surge el concepto de espacio reducido, cuando las variables de estado son escogidas más cuidadosamente pudiendo reflejar de manera menos directa pero aun completa la situación física, reduciendo la dimensión del espacio de juego resultante.

Las ventajas de un espacio reducido radican en el menor número de variables a considerar, menor cantidad de ecuaciones a trabajar y en general una representación más compacta, pudiendo incluso si  $n$  es reducido a 3 ó menos tener la posibilidad de una representación geométrica visual que pueda ayudar a la síntesis de la solución buscada. Por otro lado en el espacio realista, aunque las ecuaciones son más numerosas, suelen ser más simples además de que representan de forma directa lo que ocurre físicamente; en un espacio reducido las ecuaciones resultantes pueden ser considerablemente más complejas.

Solo se considerarán juegos de persecución y evasión de información completa, lo que significa que en todo momento cada jugador conoce los valores de todas las variables de estado.

## 3.2 La Patrulla Restringida

Este es un juego discreto de persecución y evasión sencillo, básicamente permite mostrar el uso de los conceptos antes descritos.

### 3.2.1 Reglas

- La patrulla P persigue al auto de los criminales E a través de la ciudad, cuyas calles forman una cuadrícula ilimitada.
- P tiene el doble de velocidad de E.
- P debe obedecer las leyes de tráfico, las cuales prohíben las vueltas en U y giros hacia la izquierda.
- E no obedece ninguna ley de tráfico y se mueve en cualquiera de las 4 direcciones posibles.
- La captura ocurre cuando P y E coinciden o son adyacentes.
- Las movidas son hechas alternativamente, un turno para P, luego otro para E y así sucesivamente.
- La utilidad está definida como el número de movimientos que P necesita para capturar a E.

La figura 3.1 es una representación del espacio reducido del problema, bajo la convención de que P siempre está en el origen mirando hacia el norte, siendo solo necesarias un par de coordenadas  $(x, y)$  para ubicar a E de forma relativa a P. Se aprecian los vectogramas

de  $P$  (en el origen) y  $E$ ; la Zona de captura está formada por los 9 vértices (incluyendo el origen) marcados con "x". Note como afecta el P-vectograma a un punto  $x$  en este espacio reducido, si  $P$  usa el vector  $A$ ,  $x$  se mueve a  $A'$  y si usa  $B$ ,  $x$  pasa a  $B'$ ; en el caso de  $E$  el punto  $x$  es modificado directamente por el vector usado del E-vectograma, es decir,  $x$  se desplaza una posición en una de las 4 direcciones posibles.

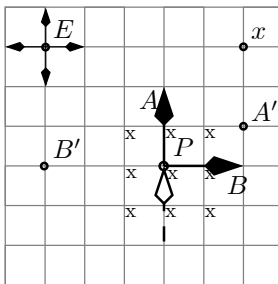


Figura 3.1: Espacio Reducido de la Patrulla Restringida



### 3.3 El Chofer Homicida

El famoso y bien conocido Chofer Homicida es un problema clásico en el ámbito de juegos de persecución y evasión, reúne las características más saltantes comunes a muchos de estos juegos.

Un auto cuyo chofer intenta atropellar a un transeúnte en un estacionamiento infinito sin obstáculos. El auto asume el rol del perseguidor P, mientras que el transeúnte toma el rol del evasor E. P intentará lograr su objetivo lo antes posible en tanto que E buscará demorar tanto como pueda la captura.

P y E se mueven por el plano con velocidades  $v_P$  y  $v_E$  respectivamente, siendo  $v_P > v_E$ ; estas velocidades son fijas y no simplemente acotadas, ya que las estrategias óptimas demandan velocidad máxima todo el tiempo.

P tiene una restricción en su movimiento, el radio de curvatura de su trayectoria nunca debe ser menor que un  $R$  fijo dado. En tanto que E puede optar por ir en cualquier dirección en cualquier momento siendo su única limitación su velocidad máxima, en otras palabras, la trayectoria de E no necesariamente tiene tangente en todo momento.

El tiempo que tome la captura es la utilidad para los jugadores y la zona de captura usualmente se considera como una región circular de radio  $l$  fijada a P, la cual se mueve con él todo el tiempo, la captura ocurre cuando E entra en esta región. Ambos jugadores disponen de información completa para realizar sus movimientos.

La figura 3.2 muestra el espacio realista del Chofer Homicida, P no puede alcanzar los puntos que estén a una distancia menor de  $2R$  sobre el segmento perpendicular a su dirección debido a su radio de curvatura mínima.

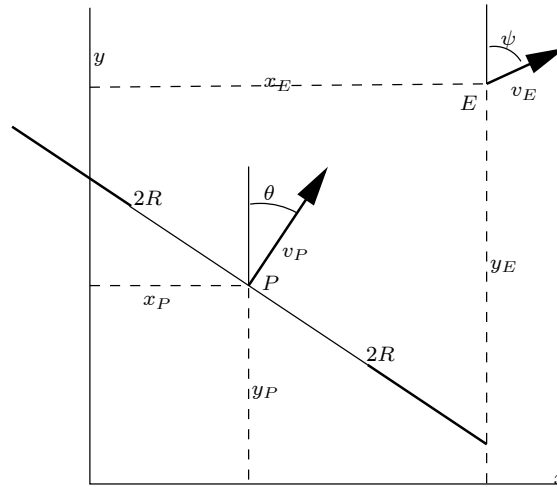


Figura 3.2: Espacio Realista del Chofer Homicida

### 3.3.1 Formulación del Modelo Matemático

La dinámica de los jugadores de este juego puede ser representada por cinco variables de estado y dos variables de control

$$\dot{x}_P = v_P \sin \theta \quad (3.2)$$

$$\dot{y}_P = v_P \cos \theta$$

$$\dot{x}_E = v_E \cos \psi$$

$$\dot{y}_E = v_E \sin \psi$$

$$\dot{\theta} = \frac{v_P}{R} \phi \quad -1 \leq \phi \leq 1$$

Donde:

$x_P, y_P, \theta$  son variables de estado y  $\phi$  es variable de control para P.

$x_E, y_E$  son variables de estado y  $\psi$  es variable de control para E.

$R$  es el radio de curvatura mínima de P.

El planteamiento inicial 3.2 tiene 5 variables de estado y puede considerarse como un espacio realista. No obstante, se puede lograr un espacio reducido con solo 2 variables de estado bajo la consideración de que el origen de coordenadas estará fijado a P y que el eje  $Y$  coincidirá siempre con la dirección de la velocidad de P; en otras palabras, E quedaría siempre en coordenadas relativas a P. Finalmente, la dinámica del juego en este espacio reducido queda como:

$$\begin{aligned} \dot{x} &= -\frac{v_P}{R}y\phi + v_E \sin \psi \\ \dot{y} &= \frac{v_P}{R}x\phi - v_P + v_E \cos \psi \quad -1 \leq \phi \leq 1 \end{aligned} \quad (3.3)$$

La figura 3.3 muestra el espacio reducido del Chofer Homicida. P siempre en el origen de coordenadas con dirección sobre el eje  $Y$ , E en coordenadas relativas a P, la variable de control  $\phi$  permite a P escoger un punto C que será el centro de su círculo de curvatura, nunca menor a la distancia  $R$ .

Este problema no siempre tiene solución para cualquier conjunto de parámetros dado. El cumplimiento de la siguiente condición garantiza la solución del problema:

$$\frac{l}{R} > \sqrt{1 - \gamma^2} + \arcsin \gamma - 1 \quad (3.4)$$

Donde:

$\gamma = v_E/v_P < 1$  es la razón entre la velocidad de E y P.

$R$  es el radio de curvatura mínima de P.

$l$  es el radio de la zona de captura circular.

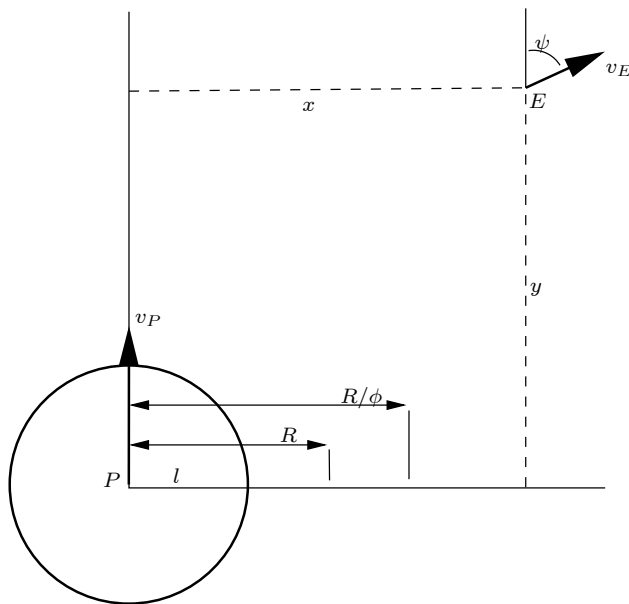


Figura 3.3: Espacio Reducido del Chofer Homicida

Para mayores detalles sobre la construcción e interpretación de esta condición remítase al capítulo 9 de [1].

La necesidad de definir una zona de captura y no simplemente tratar a P como un punto queda de manifiesto al analizar 3.4. Una zona de captura colapsada a un punto sería equivalente a tomar  $l = 0$ , con lo cual 3.4 queda reducida a:

$$0 > \sqrt{1 - \gamma^2} + \arcsin \gamma - 1 \quad (3.5)$$

Donde  $0 < \gamma < 1$ , cerca a 0 si  $v_P$  es muy grande en comparación con  $v_E$  y cerca a 1 para velocidades similares, obviamente siempre en cumplimiento de  $v_E < v_P$ . Por lo

tanto el lado derecho de 3.5 estaría acotado por:

$$0 < \sqrt{1 - \gamma^2} + \arcsin \gamma - 1 < \frac{\pi}{2} - 1 \quad (3.6)$$

Con la cual 3.5 nunca podría cumplirse. Note que  $R$  no afecta en esta conclusión, podría tenerse un  $P$  más ágil ( $R$  pequeño) pero sin una zona de captura diferente a un punto, la captura nunca ocurriría.

### 3.3.2 Entendiendo las estrategias óptimas

Bajo las condiciones planteadas para el chofer homicida, el comportamiento óptimo que deben tener  $P$  y  $E$  en un espacio realista, se puede separar básicamente en dos escenarios:

- Si  $E$  está más o menos en frente o suficientemente lejos de  $P$ , en este caso  $P$  enruta hacia  $E$  tan rápido como se lo permita su radio de curvatura mínima, mientras que  $E$  trata de huir en dirección opuesta a  $P$  como se aprecia en la figura 3.4.

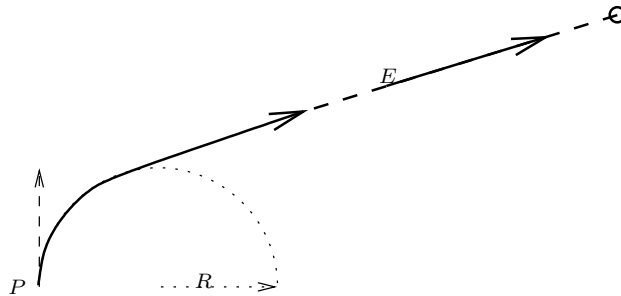


Figura 3.4: Primer Escenario de Persecución - Maniobra Simple

- Si  $E$  está más o menos detrás de  $P$ , en este caso  $P$  debe tener cuidado de intentar enrutar directamente a  $E$ , ya que  $E$  puede frustrar su intento entrando a cualquiera

de los círculos de curvatura mínima de P que tiene a los lados como se aprecia en la figura 3.5(a); P primeramente debe alejarse lo suficiente de E tomando la dirección opuesta a la de éste, en este transcurso E intentará alcanzar uno de estos círculos de curvatura mínima siguiendo a P por detrás. En el momento que E ya no pueda alcanzar los círculos de curvatura mínima, E debe emprender la huida tomando la dirección opuesta a P, en tanto que P empieza a girar hasta tener a E en la primera situación antes descrita como se aprecia en la figura 3.5(b).

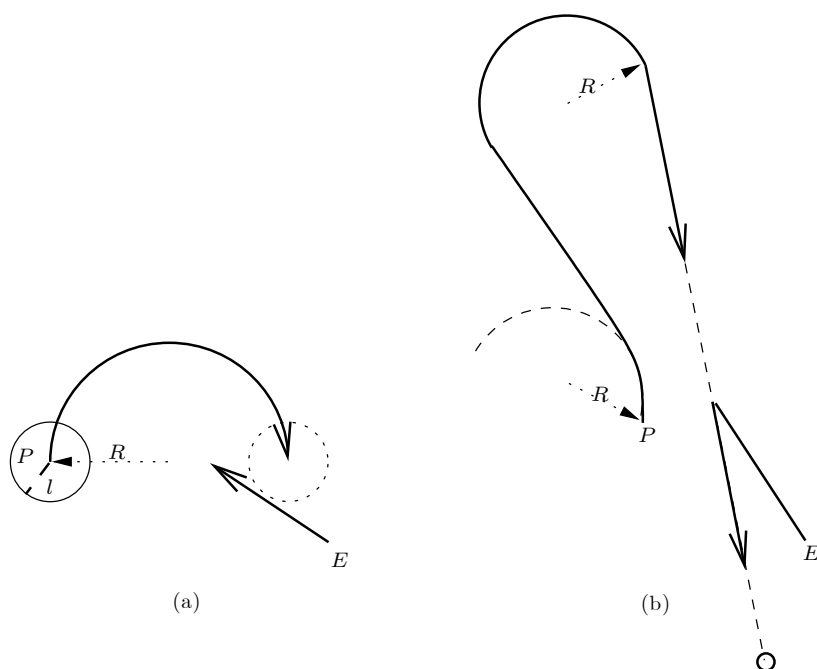


Figura 3.5: Segundo escenario de Persecución - Maniobra Compleja

La figura 3.6 muestra las trayectorias óptimas del Chofer Homicida en el espacio reducido, las interpretaciones antes hechas ahora se transforman en el propósito de P de llevar el punto  $x$  a la porción positiva del eje vertical, si es necesario  $x$  tendrá que ser conducido primero hacia la porción negativa del eje vertical hasta que haya una trayectoria

que permita evitar la barrera. Obviamente E hará todo lo posible por demorar el avance de  $x$ .

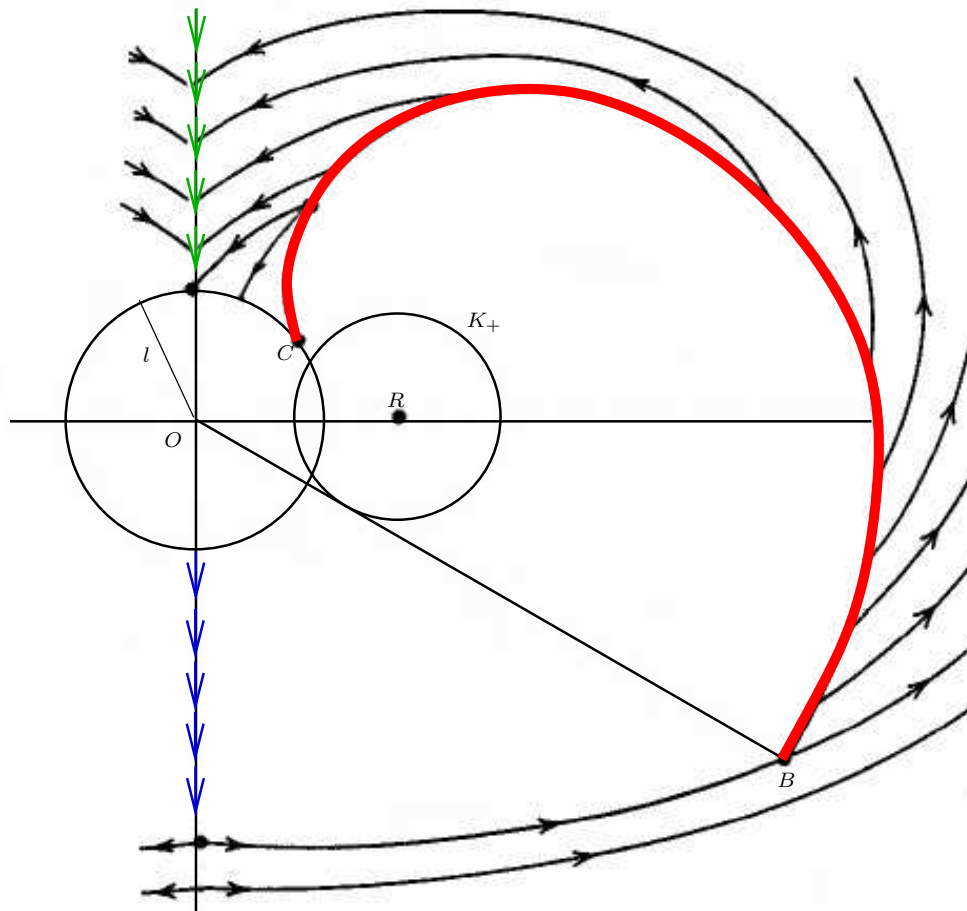


Figura 3.6: Curvas Singulares y Trayectorias Óptimas del Chofer Homicida en el Espacio Reducido

### 3.3.3 Curvas singulares

El Chofer Homicida presenta en su solución 3 tipos de curvas singulares (superficies en el caso general) muy importantes para los juegos de persecución y evasión.

- **Universal:** Una vez que los jugadores están en la curva universal, la estrategia óptima

para ambos es mantenerse en la curva. P sigue directamente a E sin hacer giros.

- **Dispersión:** Donde E puede decidir por un periodo de tiempo si ambos jugadores deben mantenerse en esta curva, o si quiere virar hacia algún lado.
- **Barrera:** Donde escenarios similares son separados por utilidades significativamente diferentes.

Estas curvas y la zona de captura (considerada usualmente circular) pueden ser apreciadas en la figura 3.6.

- La zona de captura tiene  $V(x) = 0$
- La línea delante de P (en verde) es la curva universal.
- La línea detrás de P (en azul) es la curva de dispersión.
- Las curvas laterales a P (en rojo) son las barreras.

### 3.3.4 Curvas de Barrera de $V(x)$

Se verá de forma más detallada esta curva singular ya que servirá como criterio de evaluación para la solución numérica propuesta para el Chofer Homicida.

Las curvas de barrera que aparecen en el valor del juego  $V(x)$  del Chofer Homicida son las involutas de los círculos  $K_+$  y  $K_-$  que tocan la Zona de captura.  $K_+$  y  $K_-$  en el *espacio reducido* tienen centro en  $(\pm R, 0)$  respectivamente, ambos de radio  $\frac{v_E}{v_P} R$ . Las ecuaciones que definen a la barrera de  $K_+$  son:



$$x = (l - v_P \tau) \sin(S + c\tau) + R(1 - \cos c\tau) \quad (3.7)$$

$$y = (l - v_P \tau) \cos(S + c\tau) + R \sin c\tau$$

Donde:

$l$  es el radio de la Zona de captura.

$$\cos S = \frac{v_E}{v_P}, \quad 0 < S < \frac{\pi}{2}$$

$$c = \frac{v_P}{R}$$

$R$  es el radio del círculo de curvatura mínima de P.

$\tau$  tiempo, en el sentido de cuanto le toma a  $(x, y)$  llegar a la zona de captura.

La figura 3.6 muestra la barrera derecha del valor del juego  $V(x)$ . Estas curvas comienzan en la frontera de la Zona de captura, punto C, y terminan en el punto B, tal que el segmento que une el origen y B es tangente a  $K_+$  ó  $K_-$  según sea el caso. Para un tratamiento más detallado de las curvas de barrera vea los capítulos 9 y 10 de [1].

### 3.4 Solución

A continuación se dará solución numérica a los juegos formulados. Por resolver el juego debemos entender que a partir de los datos:

1. Posiciones de P y E en el plano.
2. Dinámica de P y E, velocidad máxima y curvatura mínima para P y velocidad máxima de E. O en su defecto los vectogramas respectivos.
3. El radio de la zona de captura circular.

Se puedan establecer estrategias óptimas aproximadas tanto para P y E que permitan obtener las trayectorias óptimas aproximadas mediante las cuales P captura a E.

Se presentará un esquema de tiempo-discreto y espacio-discreto sobre el espacio reducido con la ayuda de una malla de estructura uniforme; una vez establecida la dinámica aproximada se procederá a obtener el valor del juego  $V(x)$  sobre la malla mediante un enfoque basado en programación dinámica; a partir de este resultado, se podrán calcular las estrategias óptimas tantas veces como se desee para situaciones particulares del juego.

Primeramente se representa el sistema de forma discreta:

$$x_{k+1} = f(x_k, u_k) \tag{3.8}$$

Para entender de mejor forma la discretización de  $f$  recurrimos a los vectogramas tanto para P y E que fueron previamente definidos, inicialmente estos eran conjuntos continuos de vectores, ahora estos vectogramas serán transformados en conjuntos discretos

tratando de preservar las propiedades mas importantes tanto para P y E; magnitud de velocidad y curvatura mínima en el primer caso y magnitud de velocidad en el segundo caso. Más adelante en la sección *el Chofer Homicida discreto*, se mostrarán cálculos pertinentes para la transformación de los vectogramas continuos en discretos.

Las restricciones de los controles tanto para P y E ahora están agrupadas como un solo conjunto:

$$u_k(x_k) \in U_x \quad (3.9)$$

El valor del juego representará el tiempo que toma llevar el estado inicial  $x_0$ , al estado final  $x_N$  que es cuando ocurre la captura. La relación para encontrar el valor del juego queda definida por:

$$V_N(x_N) = c_N(x_N) \quad (3.10)$$

$$V_k(x_k) = \min_{u_k \in U_x} \{c_k(x_k, u_k) + V_{k+1}(f(x_k, u_k))\} \quad k = 0, 1, 2, \dots, N - 1 \quad (3.11)$$

El tiempo que representa el valor del juego puede entenderse en el sentido de turnos, es decir, cuantos turnos le toma a P capturar a E, en cuyo caso  $V_N(x_N) = 1$ , lo cual representa el hecho de que le toma un turno a P realizar la captura cuando ya no hay más decisiones que hacer. También se puede entender en el sentido de que con la velocidad que representan los vectogramas de P y E cuantas unidades de tiempo toma a P realizar la captura.

### 3.4.1 Principio Inductivo para el cálculo del Valor del Juego

Aunque la ecuación de optimalidad 3.11 está planteada en forma recursiva, usualmente no es resuelta de esta forma, ya que demanda trabajo que aumenta exponencialmente; en cambio tomaremos un enfoque "*bottom-up*" que irá calculando progresivamente el valor del juego usando resultados óptimos anteriores.

Para cada celda  $x$  en la malla del espacio reducido,  $V(x)$  representa el valor del juego, el cual se puede entender como el número de pasos que le cuesta a P (en el origen) capturar a E (en  $x$ ), como trabajamos en el espacio reducido el origen de coordenadas y las celdas de la zona de captura centrada en el origen tendrán  $V(x) = 0$ . Se asumirá jugadas por turnos, siendo P el primero en mover colocará utilidad 1 en los puntos que puede llevar a la zona de captura en una movida.

En el turno de E se calculará un frente que incluirá a todos los puntos para los cuales todos los vecinos (desde el punto de vista de E) tengan utilidad asignada y al menos uno de estos vecinos con la utilidad más alta hasta ese momento.

Seguidamente se incluirán aquellos nuevos puntos para los cuales una movida de P los lleva al frente antes calculado y se les asignará la utilidad más alta incrementada en uno.

Generalizando la idea tenemos:

1. Suponga que tenemos encontrados y marcados todos los puntos con utilidad  $P = 0, 1, 2, \dots, n$ .
2. Sea  $S_n$  el conjunto de los puntos  $x_n$ , tal que todos los vecinos (desde el punto de vista de E) de  $x_n$  han sido encontrados y marcados y al menos uno de estos vecinos tiene

asignada utilidad  $n$

3. El conjunto  $S_{n+1}$  de puntos  $x_{n+1}$ , consta de aquellos puntos que no hayan sido marcados aún, los cuales pueden ser llevados a  $S_n$  mediante una movida de P, estos puntos ahora son marcados con utilidad  $n + 1$ .

### 3.4.2 La Patrulla Restringida

La solución a este juego utiliza el espacio reducido del juego previamente descrito, el cual nos permite lograr una representación más compacta del estado de los jugadores; la desventaja usual como ya se mencionó es que genera mayor complejidad en las movidas de los jugadores.

En el espacio reducido se establece el origen de coordenadas sobre P, con P en dirección norte. El P-vectograma consta de 2 vectores:

	E. Realista	E. reducido
Vector $ v  = 2$	Como vector	Transforma $(x, y)$
Avanzar	$(0, 2)$	$(x, y - 2)$
Girar derecha	$(2, 0)$	$(-y, x - 2)$

Tabla 3.1: P-vectograma

Note como cambia  $(x, y)$  cuando P realiza un giro a la derecha debido a las condiciones del espacio reducido. El E-vectograma consta de 4 vectores:

#### Valor del juego $V(x)$

Aplicando el principio inductivo bajo las consideraciones anteriores arribamos al resultado mostrado a continuación, note que más allá de  $V = 11$  el principio inductivo no

Vector $ v  = 1$	E. Realista	E. reducido
	Como vector	Transforma $(x, y)$
Adelante	$(0, 1)$	$(x, y + 1)$
Derecha	$(1, 0)$	$(x + 1, y)$
Atrás	$(0, -1)$	$(x, y - 1)$
Izquierda	$(-1, 0)$	$(x - 1, y)$

Tabla 3.2: E-vectograma

produce nuevos puntos a ser considerados, lo que implica que el problema no tiene solución (E puede escapar indefinidamente) si se encuentra o consigue llegar a las posiciones no marcadas.

			8	11					
			5	8					
		3	4	7	10				
		2	3	4	7				
	1	1	1	3	6	9			
	1	1	1	2	3	6			
	0	0	0	1	1	5	8		
	0	(0)	0	1	1	2	3		
9	0	0	0	1	1	3	4	5	8
	7	4	3	2	3	4	7	8	11
	10	7	4	3	6	7	10		
		8	5	6	9				
		11	8						

Tabla 3.3: Valor del juego  $V(x)$ . Origen en (0)

### Trayectorias óptimas

El siguiente paso en la resolución del juego consiste en encontrar las trayectorias óptimas. Teniendo en cuenta que  $V(x)$  está en el espacio reducido se encontrarán las trayectorias óptimas en este espacio, para luego transformarlas en las trayectorias óptimas del espacio realista, las cuales reflejan de mejor forma como transcurre el juego. El proceso

sigue los siguientes pasos:

- A partir de las posiciones de P y E en el espacio realista se obtiene el punto  $x$  en el espacio reducido.
- Usando el cuadro 3.3 obtenemos, si existe, el valor del juego  $V(x)$  el cual nos dice cuantos turnos le toma a P atrapar a E. Para las posiciones en las que no hay valor del juego, se hace  $V(x) = \infty$ , lo cual es consistente con el hecho de que a P le toma tiempo indefinido capturar a E, además de que E buscará ocupar una de estas posiciones.
- Construimos  $M$ , la matriz de utilidades de  $2 \times 4$  donde  $m(i, j)$  contiene el nuevo valor del juego una vez que P usa su  $i$ -ésimo vector y luego E su  $j$ -ésimo vector.
- Se encuentra el punto de silla de  $M$ , el cual nos dirá que vectores deben usar P y E respectivamente.
- El proceso se repite hasta que alguna de las matrices  $M$  tiene como punto de silla el valor 0 el cual indica captura.
- Se transforma la lista de movimientos (vectores) que realizaron P y E al espacio realista.

Note que el punto de silla si existe tiene un valor único, no obstante puede presentarse en más de una posición en  $M$  siendo igualmente óptima cualquiera de las combinaciones respectivas para P y E. En caso de no existir el punto de silla, P y E deben elaborar estrategias mixtas lo cual básicamente implica escoger las movidas más prometedoras y de éstas tomar una al azar.

### 3.4.3 El Chofer Homicida discreto

El juego del Chofer Homicida ha sido resuelto mediante varios métodos numéricos, entre ellos podemos mencionar los conjuntos de nivel [10], las soluciones de viscosidad [7]-[8], el método de las características y programación dinámica<sup>1</sup>. Este último enfoque es el que será usado para obtener una solución numérica del problema.

Primeramente se encontrarán equivalentes discretos satisfactorios de los vectogramas tanto de P y E. A partir de la dinámica planteada en 2.6 y teniendo en cuenta los posibles valores de las variables de control, se puede ver que en una fracción de tiempo los vectogramas de P y E se pueden considerar como se muestra en 3.7.

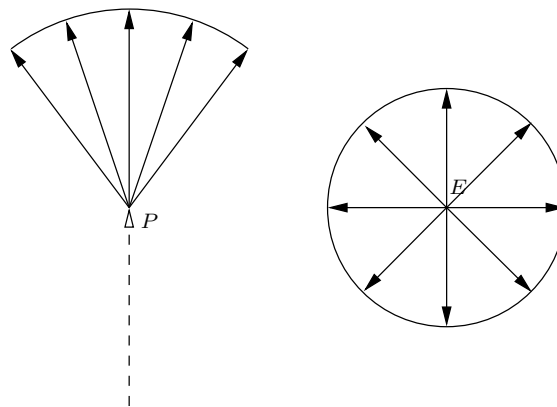


Figura 3.7: Vectogramas del Chofer Homicida

Tomaremos subconjuntos finitos de estos vectogramas para los equivalentes discretos bajo las siguientes consideraciones:

---

<sup>1</sup>La ecuación de Hamilton-Jacobi-Bellman y el Principio del Máximo de Pontryagin son dos métodos exitosos para resolver en general juegos dinámicos [11].



### P-vectograma discreto

- Cada vector tendrá magnitud  $v_P$  o un múltiplo de acuerdo al paso de tiempo que se tome.
- Estarán equiespaciados por el mismo ángulo entre vectores consecutivos.
- Estarán limitados por ángulos  $\pm\alpha$  con respecto a la vertical de tal forma que preserven lo mejor posible el círculo de curvatura mínima.

En resumen:

$$P = \{P_i : P_i = v_P \Delta t [\cos(\frac{\pi}{2} - \alpha + i\beta), \sin(\frac{\pi}{2} - \alpha + i\beta)]; i = 0, \dots, n; \beta = \frac{2\alpha}{n}\} \quad (3.12)$$

$$\alpha = \frac{v_P}{R} \Delta t \quad (3.13)$$

Donde:

$v_P$  es la magnitud de la velocidad de P.

$\Delta t$  es el paso de tiempo.

$\alpha$  es el ángulo que preserva el círculo de curvatura mínima. Para encontrarlo se usó 3.2; integrando para  $\theta$  se obtiene  $\theta = \frac{v_P}{R} \phi t$ , luego se toma el control extremo  $\phi = 1$ .

$R$  es el radio del círculo de curvatura mínima.

$n+1$  es la cantidad de vectores que tendrá el P-vectograma discreto. Se recomienda  $n$  par para tener un vector vertical (para avanzar en línea recta).

### E-vectograma discreto

- Cada vector tendrá magnitud  $v_E$  o un múltiplo de acuerdo al paso de tiempo que se tome.
- Estarán equiespaciados por el mismo ángulo entre vectores consecutivos hasta que todos estos ángulos sumen  $2\pi$ .

En resumen:

$$E = \{E_i : E_i = v_E \Delta t [\cos(i\beta), \sin(i\beta)]; i = 0, \dots, m - 1; \beta = \frac{2\pi}{m}\} \quad (3.14)$$

Donde:

$v_E$  es la magnitud de la velocidad de E.

$\Delta t$  es el paso de tiempo.

$m$  es la cantidad de vectores que tendrá el E-vectograma discreto. Se recomienda  $m$  par para tener vectores verticales (para avanzar y retroceder en línea recta).

El siguiente paso es establecer una relación entre las unidades de espacio tanto físicas como de la malla, para escalar adecuadamente los vectogramas con el propósito de lograr mejor precisión. Se usará:

$$\text{Unidad de espacio física} = k * \text{Unidad de espacio malla} \quad (3.15)$$

Donde  $k$  es un entero positivo.

Con la dinámica discreta establecida, los pasos a seguir para encontrar soluciones numéricas son similares a los vistos en el juego de la Patrulla Restringida. Note que el valor

del juego solo necesita ser calculado una vez, en adelante se puede reusar para determinar trayectorias óptimas a partir de diferentes posiciones iniciales de P y E.

### Errores de aproximación

Debido a la naturaleza discreta de la malla, donde los vértices a los que se ajustarán los puntos tienen coordenadas enteras, se debe tener en cuenta que el efecto de los vectogramas discretos obtenidos para P y E sobre un punto  $x$  en la malla producirá un nuevo conjunto de puntos, que tendrá que ser reajustado dentro de la malla. Para ello se usa redondeo al entero más cercano para cada una de las componentes de un punto dado, este redondeo puede producir un error en la magnitud del vector posición y en el ángulo con la vertical como se aprecia en la figura 3.8.

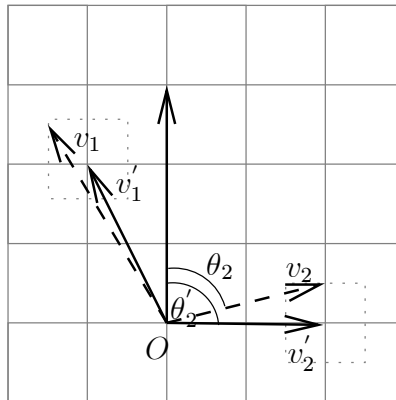


Figura 3.8: Error en Magnitud y Angulo de los vectores de la malla

El error máximo por redondeo en la magnitud del vector posición de un punto  $x$  diferente del origen de coordenadas es:

$$\varepsilon_m = \frac{\sqrt{2}}{2} \text{unidades malla}$$

En tanto que el error relativo máximo por redondeo en la magnitud es:

$$\eta_m = \frac{\sqrt{2}}{2 \|\vec{x}\|}$$

El error máximo por redondeo en el ángulo del vector posición de un punto  $x$  es:

$$\sin \varepsilon_a = \frac{1}{2 \|\vec{x}\|}$$

Note que los errores son menos significativos en tanto el punto  $x$  esté más alejado del origen. Aunque la malla representa un espacio reducido del Chofer Homicida y necesita mejor precisión justamente en las cercanías del origen de coordenadas.

Los parámetros a considerar para intentar controlar los errores serán el paso de tiempo  $\Delta t$  y el factor de escalamiento  $k$  de 3.15, teniendo en cuenta que:

- Un  $\Delta t$  pequeño favorece a una mejor aproximación de la solución, pero requiere mayor precisión en los cálculos lo que debe ser compensado tomando  $k$  grande, lo cual finalmente aumenta la cantidad de operaciones a realizar.
- Un  $\Delta t$  grande perjudica a la aproximación de la solución, pero requiere menor precisión en los cálculos lo que permite tomar  $k$  pequeño, lo cual finalmente disminuye la cantidad de operaciones a realizar.
- Lo ideal es encontrar un  $\Delta t$  no muy pequeño y un  $k$  no muy grande que permitan mantener los errores por redondeo dentro de una tolerancia aceptable según el contexto del problema.

Por otro lado, la manipulación de  $\Delta t$  y  $k$  debe contemplar como se comportan P y E en la generalidad de los casos cuando usan sus estrategias óptimas. La sección 3.3.2 describe los dos escenarios principales que ocurren durante una persecución y evasión, al inspeccionar las figuras 3.4 y 3.5 no es difícil darse cuenta que:

- En el espacio realista, P sigue directamente a E o viaja a través del círculo de curvatura mínima la mayor parte del tiempo. En el espacio reducido esto es equivalente a ir verticalmente o usar los vectores extremos del P-vectograma.
- En el espacio realista, E huye directamente de P o lo sigue directamente la mayor parte del tiempo. En el espacio reducido esto es equivalente a ir verticalmente.

De donde se puede extraer una condición necesaria que debe cumplir la combinación ideal de  $\Delta t$  y  $k$ :

$$(v_P \Delta t)k \text{ y } (v_E \Delta t)k \text{ deben ser enteros.} \quad (3.16)$$

## 3.5 Algoritmos

### 3.5.1 Consideraciones

**Malla:** Matriz de  $n \times n$  elementos que representará el espacio reducido y tendrá los valores de  $V(x)$ . El origen de coordenadas está fijado en el centro de la malla. La malla es de estructura uniforme.

**Zona de captura:** Círculo con centro en el origen y radio  $l$ . Los puntos correspondientes de la Zona de captura en la malla son marcados con  $V(x) = 0$ .

**P-vectograma:** Conjunto de  $p$  vectores que representa la dinámica discreta de P.

**E-vectograma:** Conjunto de  $e$  vectores que representa la dinámica discreta de E.

**Frontera E:** Conjunto de puntos sobre la malla, tal que para cada punto todos sus vecinos según el E-vectograma tienen valor asignado y al menos uno de estos vecinos tiene utilidad  $V$

**Frontera P:** Conjunto de puntos sobre la malla que no tienen utilidad asignada, tal que mediante el P-vectograma estos puntos pueden ser llevados a la Frontera E con utilidad  $V$ .

### 3.5.2 Valor del Juego $V(x)$

El valor del juego  $V(x)$  será calculado a partir de la zona de captura donde empieza con  $V(x) = 0$ , luego mediante cálculos sucesivos de la Frontera E y la Frontera P se obtienen las nuevas regiones de la malla con el valor del juego incrementando, este proceso puede continuar hasta que el valor del juego llegue a los bordes de la malla.

#### Algoritmo

1. Inicializar toda la malla con  $V = -1$ ;
2. Marcar la Zona de captura con  $V = 0$ ;
3. Marcar la Zona después de la primera movida de P con  $V = 1$ ;
4. Mientras se pueda calcular utilidades en la malla
5.     Determinar Frontera E con utilidad  $V$ ;
6.     Marcar Frontera P con utilidad  $V + 1$ ;
7.     Incrementar  $V$ ;

#### Complejidad

1.  $O(n^2)$ : La malla es una matriz cuadrada de dimensión  $n \times n$ , en cada elemento se realiza una operación de tiempo constante.
2.  $O(l^2)$ : La cantidad de puntos en la malla correspondientes a la Zona de captura (círculo de radio  $l$ ) es proporcional a  $l^2$ , en cada elemento se realiza una operación de tiempo constante.

3.  $O(pl^2)$ : Cada uno de los puntos de la Zona de captura será sometido al P-vectograma mediante una operación de tiempo constante..
4.  $O(n)$ : Cada nuevo valor del juego genera en la malla curvas de nivel crecientes cada vez más semejantes a aros circulares.
5.  $O(ne)$ : Teniendo en cuenta que la Frontera E debe ser cercana a la curva de nivel para el valor del juego más alto calculado. Se explora una región cercana a esta última curva de nivel aplicando el E-vectograma.
6.  $O(nep)$ : A la Frontera E antes obtenida se le aplica el P-vectograma para obtener la nueva frontera P que es la nueva curva de nivel para el siguiente valor del juego que se marca en la malla.
7.  $O(1)$ : Operación de tiempo constante.

**Total:** Dado que  $n > l$ , las operaciones dominantes están en las líneas (4)x(5) y (4)x(6) siendo la dominante la segunda combinación. El algoritmo es de orden  $O(n^2ep)$ .



### 3.5.3 Trayectorias óptimas

Se obtiene el valor del juego en base al punto  $x$  en el espacio reducido,  $x$  es calculado a partir de las posiciones de P y E en el espacio realista. En este caso asumimos  $V(x) = V$  un número entero, en el sentido de la solución aproximada propuesta nos indica cuantos turnos le toma a P capturar a E; donde cada turno tiene una duración de tiempo preestablecida.

#### Algoritmo

1. Realizar  $V$  veces
2. Construir la matriz de utilidades  $M_{pxe}$  para el punto  $x$ ;
3. Encontrar el punto de silla  $m(i, j)$ ;
4. Aplicar a  $x$  el  $i$ -ésimo vector de P y luego el  $j$ -ésimo vector de E;
5. Transformar la lista de vectores de P y E al espacio realista;
6. Aplicar los vectores resultantes a las posiciones iniciales de P y E respectivamente.

#### Complejidad

1.  $O(V)$ : Las trayectorias están compuestas por tantos vectores como turnos le tome a P capturar a E.
2.  $O(pe)$ : Para un punto  $x$ , P tiene  $p$  posibilidades para jugar y en cada una E tiene  $e$  posibilidades para responder.

3.  $O(pe)$ : Encontrar el punto de silla sobre la matriz  $M_{p \times e}$  demanda calcular

$$\max_{1 \leq i \leq p} \min_{1 \leq j \leq e} \{m(i, j)\} = \min_{1 \leq i \leq p} \max_{1 \leq j \leq e} \{m(i, j)\}$$

4.  $O(1)$ : Operación de tiempo constante.
5.  $O(V)$ :  $2V$  vectores son transformados a través de operaciones de tiempo constante.
6.  $O(V)$ :  $2V$  vectores son aplicados a las posiciones iniciales de P y E.

**Total:** Las operaciones dominantes son (1)x(2) y (2)x(3), ambas tienen un orden similar. El algoritmo es de orden  $O(Vpe)$

## Capítulo 4

# Resultados

La solución numérica de los problemas descritos en el capítulo anterior es desarrollada mediante la implementación de los algoritmos descritos en las secciones 3.5.2 y 3.5.3 usando Matlab.

### 4.1 La Patrulla restringida

La solución de este juego no trae mayores complicaciones, el juego en sí es discreto. La dinámica que se usa en el algoritmo del valor del juego refleja perfectamente la original, los movimientos de los jugadores también son discretos, por lo cual no hay problemas de redondeo al operar sobre la malla.

Usando el Valor del juego de la Patrulla Restringida en el espacio reducido (ver Tabla 3.3), se mostrarán dos ejemplos de trayectorias óptimas en el espacio realista a partir de posiciones iniciales de P y E.

#### **Ejemplo 1:**

Espacio Realista:  $P=(2,5)$ ,  $E=(6,8)$ ; Espacio Reducido:  $x=(4,3)$ ,  $V(x)=9$

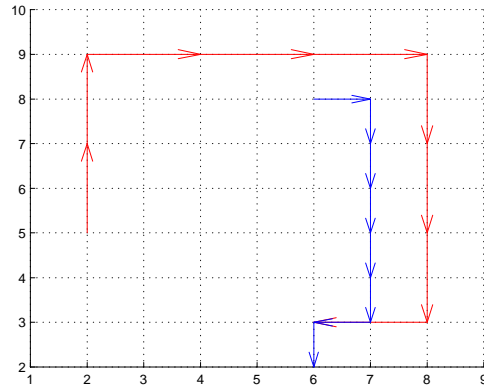


Figura 4.1: PR - 1: Trayectoria Optima

**Ejemplo 2:**

Espacio Realista:  $P=(3,5)$ ,  $E=(2,2)$ ; Espacio Reducido:  $x=(-1,-3)$ ,  $V(x)=10$

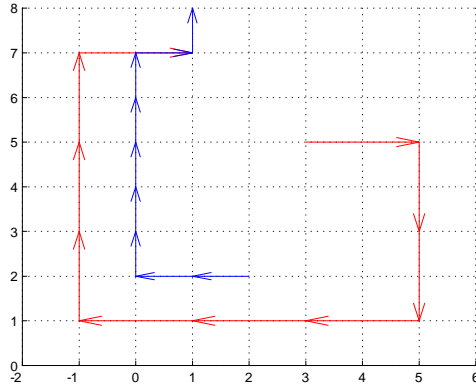


Figura 4.2: PR - 2: Trayectoria Optima

## 4.2 El Chofer Homicida

Para evaluar el éxito de la solución numérica propuesta se planteará una dinámica de prueba para el Chofer Homicida, se encontrará la solución numérica y luego se aplicarán dos criterios de evaluación:

- Comparación de las barreras de la solución numérica con las barreras teóricas de  $V(x)$ .
- Comportamiento de las trayectorias óptimas obtenidas para P y E comparadas con el comportamiento teórico ideal mencionado anteriormente.

Con la dinámica que será propuesta a continuación se busca reflejar un escenario real, luego se encuentra la contraparte discreta para proceder a obtener una solución numérica una vez establecidos los parámetros para la malla.

### 4.2.1 Dinámica real

Descripción	Variable	Valor	Unidad
Radio de la Zona de captura	$l$	2	metros
Velocidad de P	$v_P$	2	metros/segundo
Velocidad de E	$v_E$	1	metros/segundo
Radio del círculo de curvatura mínima	$R$	$16/\pi$	metros

Tabla 4.1: Dinámica Real

### 4.2.2 Dinámica discreta

#### Parámetros $\Delta t$ y $k$

Teniendo en cuenta la condición 3.16 para la dinámica propuesta, se usan los valores:

$$\Delta t = 0.25$$

$$k = 32$$

Descripción	Variable	Valor	Unidad
Equivalencia 1 metro		32	unidades malla
Paso de tiempo	$\Delta t$	0.25	segundos
Radio de la Zona de Captura	$l$	64	unidades malla
Velocidad de P	$v_P$	16	unidades malla/ $\Delta t$
Velocidad de E	$v_E$	8	unidades malla/ $\Delta t$
Radio del círculo de curvatura mínima	$R$	$512/\pi$	unidades malla
Máxima separación de la vertical	$\alpha$	$\pi/32$	radianes
P-vectograma	$p$	41	vectores
E-vectograma	$e$	52	vectores

Tabla 4.2: Dinámica Discreta

Usando los datos anteriores se procedió a generar la aproximación del Valor del juego del Chofer Homicida en el espacio reducido. La malla (matriz) usada tiene dimensiones

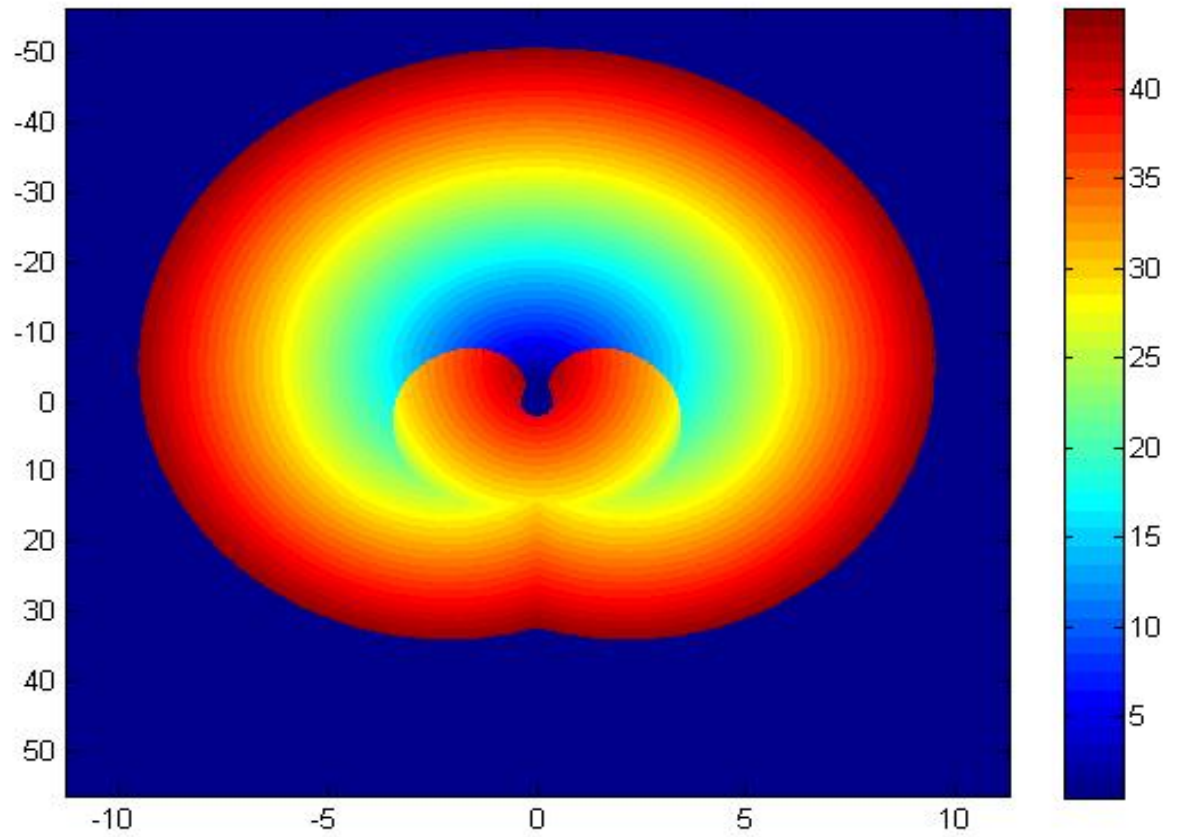


Figura 4.3: Chofer Homicida -  $V(x)$

de  $3601 \times 3601$ , se realizaron 200 iteraciones, lo que según el paso de tiempo puede ser considerado como lo necesario para resolver persecuciones de hasta 50 segundos de duración.

La generación de esta aproximación tomó 1279.54 segundos.

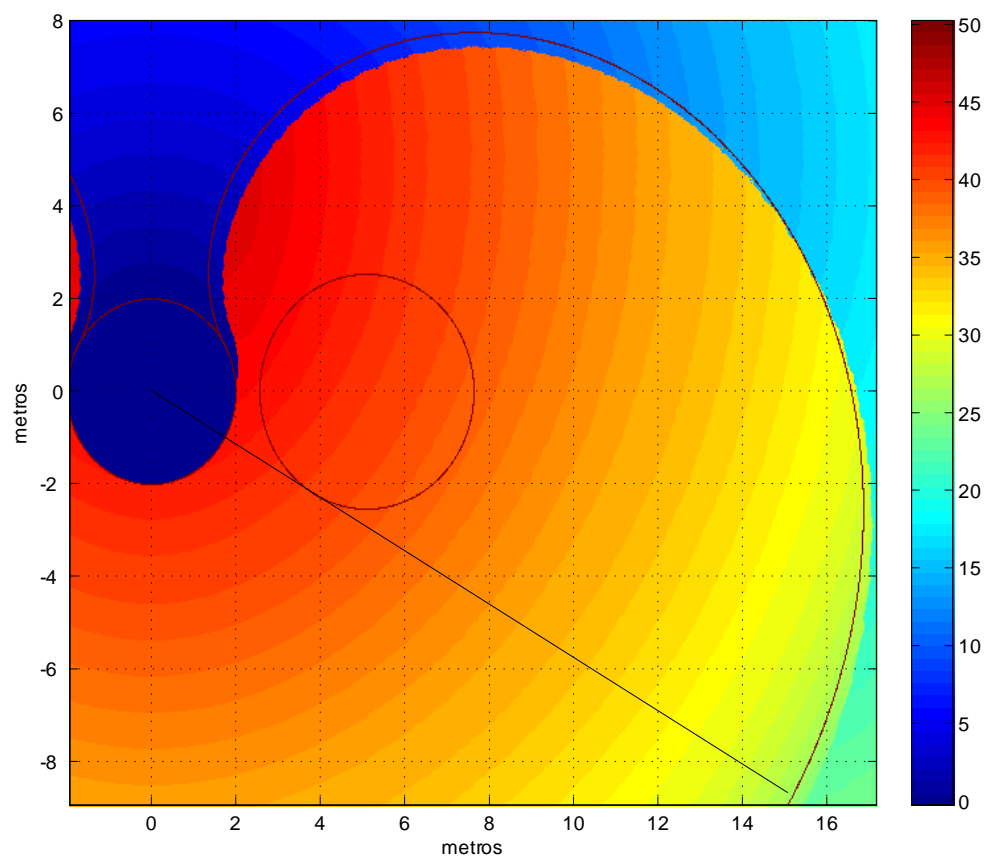


Figura 4.4: Chofer Homicida - Barrera teórica y Barrera numérica



### **Comparación de las barreras teóricas con las barreras numéricas de $V(x)$**

En la figura 4.4 se aprecia que la barrera numérica tiene una buena aproximación con la barrera teórica; por buena aproximación considérese la distancia entre las barreras comparada con la velocidad de P o la zona de captura. Visualmente la aproximación se aprecia igual o mejor que la mostrada en [9].

Sin olvidar que las barreras determinan cambios drásticos en las trayectorias óptimas, se puede apreciar que inicialmente la barrera numérica está después de la barrera teórica, esta situación no es recomendable ya que P puede pretender numéricamente alcanzar a E cuando en la realidad su círculo de curvatura mínima frustrará su intento, además de que ahora P está en peor posición que al inicio ya que permitió que E se acerque por el lado del círculo de curvatura mínima. Posteriormente la barrera numérica está antes de la barrera teórica lo que aunque no es óptimo es más deseable. Se podría alterar (disminuir) ligeramente el radio de curvatura mínima para que las barreras numéricas estén antes de las barreras teóricas.

Por otro lado se observa que la barreras a medida que pasa el tiempo se van alejando una de otra, este hecho se menciona en [9] y se describe como un error de redondeo acumulado para el escenario completo debido al uso de pasos de tiempo pequeños.

### **Comparación del comportamiento de las trayectorias óptimas numéricas con las trayectorias óptimas teóricas.**

Se presentarán algunas trayectorias óptimas numéricas correspondientes a diferentes posiciones iniciales de P y E, en todos los ejemplos se asume que P apunta al norte

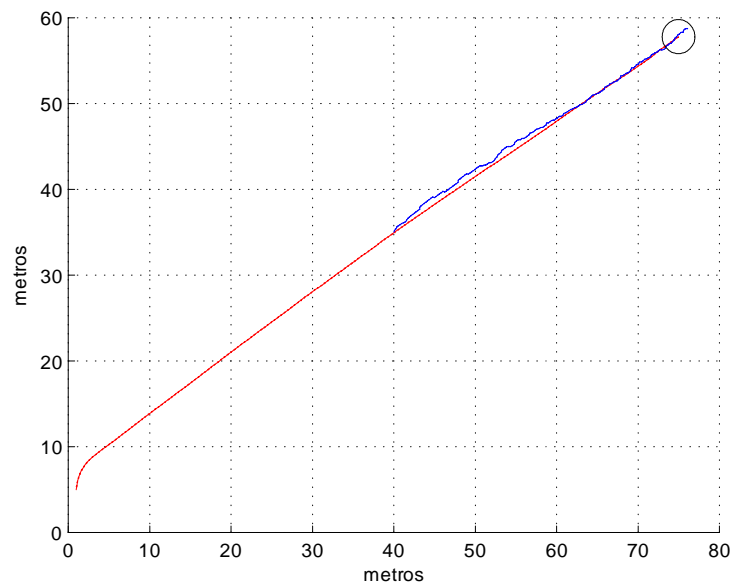


Figura 4.5: CH 3: Trayectoria Optima en el Espacio Realista

inicialmente.

### Ejemplo 3

Espacio Realista:  $P=(1,5)$ ,  $E=(40,35)$ , Espacio Reducido:  $x=(39,30)$

Pasos realizados: 183, Tiempo total: 3.8 seg, Tiempo por paso: 0.02 seg

### Ejemplo 4

Espacio Realista:  $P=(1,1)$ ,  $E=(0,-20)$ , Espacio Reducido:  $x=(-1,-21)$

Pasos realizados: 142, Tiempo total: 2.79 seg, Tiempo por paso: 0.019 seg

### Ejemplo 5

Espacio Realista:  $P=(0,0)$ ,  $E=(5,-5)$ , Espacio Reducido:  $x=(5,-5)$

Pasos realizados: 149, Tiempo total: 2.86 seg, Tiempo por paso: 0.019 seg

Las trayectorias óptimas calculadas tienen el comportamiento esperado de forma

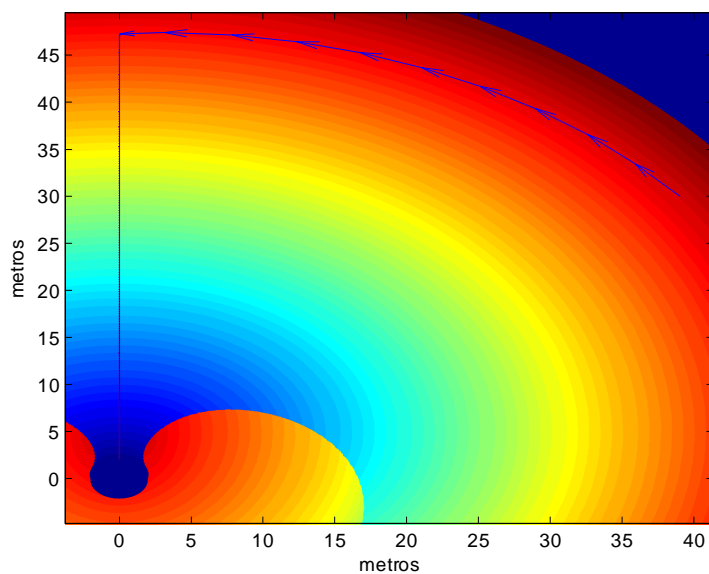


Figura 4.6: CH 3: Trayectoria Optima en el Espacio Reducido

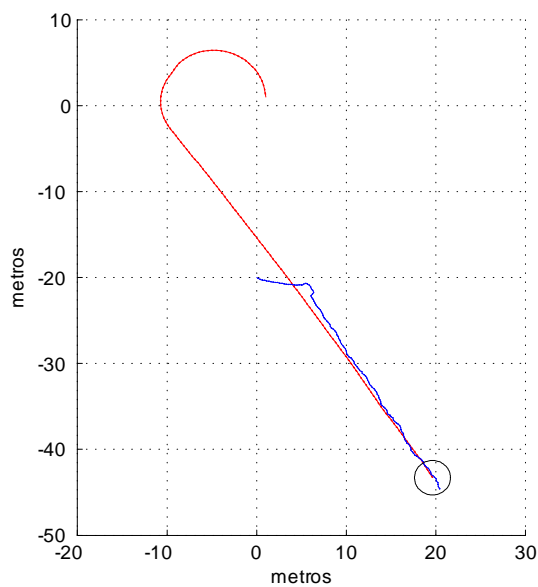


Figura 4.7: CH 4: Trayectoria Optima en el Espacio Realista

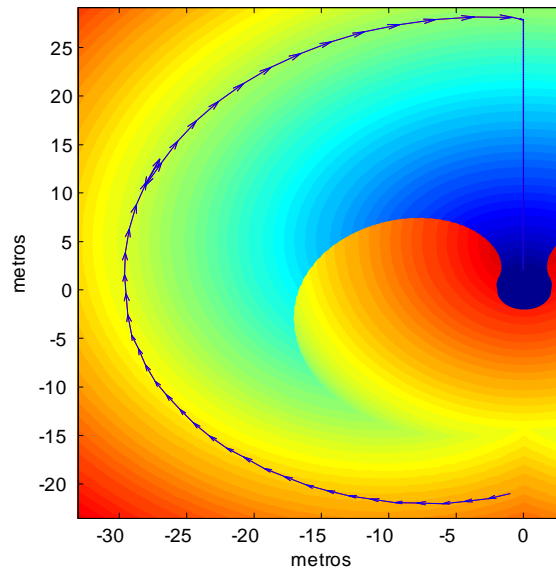


Figura 4.8: CH 4: Trayectoria Optima en el Espacio Reducido

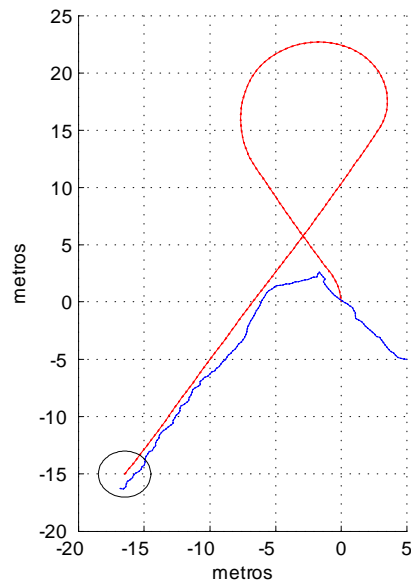


Figura 4.9: CH 5: Trayectoria Optima en el Espacio Realista

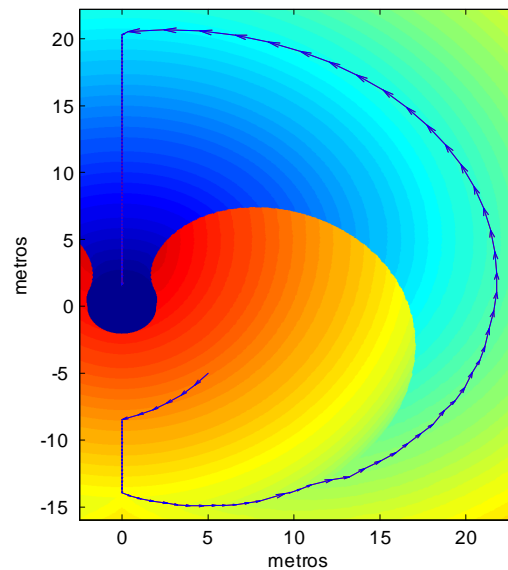


Figura 4.10: CH 5: Trayectoria Optima en el Espacio Reducido

más satisfactoria para P que para E, a continuación algunos comentarios al respecto:

- Debido al paso de tiempo tomado, el cual obviamente dista mucho de ser un valor infinitesimal, el Valor del Juego se ve afectado, convirtiéndose de su versión exacta continua en una función constante por partes.
- Debido al hecho anterior, al momento de calcular la matriz de utilidades para un punto  $x$  en busca de las direcciones que deben tomar P y E, surgen múltiples puntos de silla, que para la aproximación del Valor del Juego son soluciones aceptables.
- El tomar cualquiera de las "soluciones aceptables" puede acarrear posteriormente un error acumulado que empobrece el comportamiento óptimo de P y E.
- El empobrecimiento del comportamiento óptimo, en este punto afecta más a P que

a E; puesto que P avanza más que E, si no lo hace tan acertadamente, no se estará acercando a E tan rápido como debería.

- El algoritmo de las trayectorias óptimas propuesto inicialmente mostraba frecuente inestabilidad en las trayectorias de P y E comparados con el comportamiento óptimo teórico. En algunos casos la captura no ocurría al cumplirse la cantidad de pasos estipulada por el valor del juego inicial.
- Para estabilizar las trayectorias se modificó el algoritmo agregando una condición más, si se obtienen múltiples puntos de silla en la matriz de utilidades, de todos ellos se toma aquel que en el espacio reducido minimice el valor absoluto de la primera componente del punto  $x$ .
- La validez del criterio incorporado al algoritmo la encontramos en la equivalencia de optimalidad de las estrategias asociadas a cada punto de silla, y en el comportamiento óptimo teórico que deberían tener P y E; P alejándose de E, ambos en la misma dirección o P acercándose a E, ambos también en la misma dirección; lo que en el espacio reducido es equivalente a tener el punto  $x$  en la vertical que pasa por el origen. Cabe señalar que cuando P empieza a realizar giros, el punto de silla es generalmente único.
- Después de la modificación del algoritmo, el empobrecimiento del comportamiento óptimo afecta más a E que a P, esto debido a que ahora P tiene menos libertad de escoger entre los elementos de su vectograma debido a la rotación que ocasiona, con el consiguiente alejamiento que puede ocasionarle al punto  $x$  de la vertical. En cambio

E solo desplaza al punto  $x$ , pudiendo tener varias direcciones aparentemente óptimas según el Valor del Juego aproximado. En algunos casos la captura ocurre antes de cumplirse la cantidad de pasos estipulada por el valor del juego inicial.

- En algunos puntos de la trayectoria puede ocurrir que no exista punto de silla, para ello se incorporó un mecanismo aleatorio de selección de direcciones para P y E, que primeramente aplica el concepto de estrategia dominante para reducir la matriz de utilidades, descartando combinaciones nada útiles de vectores. Sobre la matriz reducida se aplica una selección aleatoria donde los vectores que quedaron de P tienen la misma probabilidad de ser seleccionados, similarmente para los vectores que quedaron de E.
- Finalmente, en cuanto al uso práctico de una solución numérica como ésta, el algoritmo para el cálculo de las trayectorias óptimas trabaja lo suficientemente rápido, haciendo factible su uso. Para los ejemplos vistos, cada decisión tomada nos da un tiempo de al menos 0.25 seg. para tomar la siguiente decisión y el algoritmo decide el siguiente paso en un promedio de 0.02 seg., esperando que la detección de las nuevas posiciones sea lo suficientemente rápida y precisa. Se asume que la dinámica de P y E es fija y conocida de antemano, lo cual permitiría calcular el Valor del Juego antes de empezar el juego en sí.

## Capítulo 5

# Conclusiones y Trabajo Futuro

### 5.1 Conclusiones

Se estudiaron e implementaron dos algoritmos para resolver numéricamente dos juegos de Persecución y Evasión de suma cero de dos jugadores. Entre las conclusiones más relevantes se pueden mencionar.

Se logró una solución numérica que resuelve aceptablemente los juegos tratados, su implementación no es muy compleja y el esquema de trabajo es reusable en problemas similares, siendo particularmente apropiado para dinámicas discretas o dinámicas irregulares.

Para la solución de los juegos de Persecución y Evasión se siguen principalmente dos fases, el cálculo del valor del juego y el cálculo de las trayectorias óptimas. El valor del juego puede ser reusado libremente para volver a jugar en nuevas posiciones en tanto la dinámica de los jugadores no varíe. Se usó Programación Dinámica para calcular el valor del juego.



Se pudo apreciar en el Chofer Homicida que el valor del juego en el espacio reducido requiere más precisión en las cercanías del origen de coordenadas; a medida que aumenta la distancia, la precisión puede ser disminuida, visto de otra manera si el evasor está muy distante, tiene poca importancia la maniobra que tenga que realizar el perseguidor antes de dirigirse directamente hacia su objetivo.

## 5.2 Trabajo Futuro

El uso de una malla con estructura no uniforme para el valor del juego sería muy beneficioso tanto para la precisión de los cálculos como para el tiempo de ejecución. Este tipo de malla aplicada al espacio reducido tendría el interesante efecto de aparentar refinamiento y aglomeración en el espacio real a medida que  $P$  avance.

Una adaptación a tres dimensiones es factible, aunque la generación del valor del juego demandará más tiempo dada la dimensión del espacio y la cardinalidad de los vectogramas.

La solución numérica resuelve aceptablemente los juegos planteados, no obstante las condiciones de estos son muy ideales o simples. Una solución numérica más robusta debe incluir aspectos como espacio de juego con fronteras, obstáculos e incertidumbre.

Es necesario un estudio exhaustivo sobre los estimados de errores y precisión para el enfoque usado.

# Bibliografía

- [1] *R. Isaacs*. Differential Games. A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization. Dover. 1999.
- [2] *L. D. Berkovitz*. A Theory of Differential Games. Advances in Dynamic Games and Applications. 1: 3-22. 1994.
- [3] *M. Dresher*. The Mathematics of Games of Strategy. Theory and Applications. Dover. 1981.
- [4] *M. Davis*. Game Theory. A Nontechnical Introduction. 1997.
- [5] *R. E. Bellman*. Dynamic Programming. Princeton University Press. 1957.
- [6] *R. Luus*. Iterative Dynamic Programming. Chapman and Hall/CRC. 2000.
- [7] *M. Falcone*. Numerical Methods for Differential Games based on Partial Differential Equations. AMS. 2005.
- [8] *M. Bardi, M. Falcone, P. Soravia*. Fully discrete schemes for the Value Function of Pursuit-Evasion Games. Advances in Dynamic Games and Applications. 1:89-105.1994

- [9] *A. Meyer, M. H. Breitner, M. Kriesel.* A Pictured Memorandum on Synthesis Phenomena Occurring in the Homicidal Chauffeur Game. Proceedings of the Fifth International ISDG Workshop. International Society of Dynamic Games.17:32. 2005.
- [10] *V. S. Patsko, V. L. Turova.* Homicidal Chauffeur Game. Computation of the Level Sets of the Value Function. Advances in Dynamic Games and Applications. 6: 295-322. 2001.
- [11] *E. Dockner, S. Jorgensen, N. V. Long, G. Sorger.* Differential Games in Economics and Management Science. Cambridge University Press. 2006.
- [12] *M. H. Breitner.* The Genesis of Differential Games in Light of Isaacs' Contributions. Journal of Optimization Theory and Applications. 124: 523-559. 2005.

## Apéndice A

# Código Fuente

Se incluye el código fuente en Matlab resultante de esta investigación.

### A.1 Generación de los Vectogramas

```
function [vectP, angleP] = gen_vectP(vP, R, dt, undmalla, n)
% Generar P-vectograma, sin redondeos
% Eje Y como base, generara n vectores de tamaño vP*dt*undmalla
% desde -angle/2 hasta +angle/2

r=vP*dt*undmalla;
angle=vP*dt/R;
inc=2*angle/(n-1);
angleP=pi/2-angle:inc:pi/2+angle;

%% Enfoque: Preservar el radio de los vectores (sin redondear)
vectP = r*[cos(angleP)' sin(angleP)'];
angleP=angleP'-pi/2;

function [vectE,angleE] = gen_vectE(vE, dt, undmalla, n)
% Generar E-vectograma, no incluye redondeos
% Eje Y como base, generara n vectores de tamaño vE*dt*undmalla
% Si n es cero se calcula internamente cuantos vectores se necesitan
% para tener un circulo completo en la malla
```

```

r=vE*dt*undmalla;
if n==0
    %Asegurarnos de que sea multiplo de 4
    n1=set_round(pi*r/2)*4;
    inc = 2*pi/n1;
else
    inc = 2*pi/n;
end
angleE=pi/2:inc:5*pi/2-inc;

%% Enfoque: Preservar el radio de los vectores (sin redondear)
vectE = r*[cos(angleE)' sin(angleE)'];
angleE=angleE'-pi/2;

```

## A.2 Comportamiento de los vectogramas en el espacio reducido

```

function L= set_vectP(Points, step, angleP)
% A cada punto se le aplicara cada uno de los vectores del P-vectograma
% L: Lista de Points operados con primer vectP, Points con segundo vectP ...
% step: Distancia que avanza P (constante para todos los vectores)
% angleP: Angulos de los vectores

p=size(Points,1);
n=size(angleP,1);

x= repmat(Points(:,1),1,n);
y= repmat(Points(:,2),1,n);

% Unir coordenadas x,y
aux=zeros(2*n,p);
aux(1:2:2*n,:)=x(:,1:n)';
aux(2:2:2*n,:)=y(:,1:n)';

% Matriz diagonal por bloques para rotacion
rotor=get_rot(-angleP);

% Aplicar rotacion y construir el resultado

```

```
L_=rotor*aux;
L=set_round([reshape(L_(1:2:2*n,:))',n*p,1) reshape(L_(2:2:2*n,:))',n*p,1)-step]);
```

```
function L= set_vectE(Points,vectE)
% A cada punto se le aplicara cada uno de los vectores del E-vectograma
% L: Lista de Points operados con primer vectE, Points con segundo vectE ...
% vectE: vectograma de E
```

```
p=size(Points,1);
n=size(vectE,1);
```

```
% Aplicar a cada punto cada vector
x= repmat(Points(:,1),1,n)+repmat(vectE(:,1))',p,1);
y= repmat(Points(:,2),1,n)+repmat(vectE(:,2))',p,1);
```

```
L=set_round([reshape(x,p*n,1) reshape(y,p*n,1)]);
```

```
function L= set_ivectP(Points, step, angleP)
% A cada punto se le aplicara el P-vectograma inverso
% L: Lista bajo la sgte estructura
% Points operados con primer vectP, Points con segundo vectP ...
% step: Distancia que avanza P (constante para todos los vectores)
% angleP: Angulos de los vectores
```

```
p=size(Points,1);
n=size(angleP,1);
```

```
% Aplicar a cada punto cada vector
x= repmat(Points(:,1),1,n);
y= repmat(Points(:,2),1,n)+step;
```

```
% Unir coordenadas x,y
aux=zeros(2*n,p);
aux(1:2:2*n,:)=x(:,1:n)';
aux(2:2:2*n,:)=y(:,1:n)';
```

```
% Matriz diagonal por bloques para rotacion
rotor=get_rot(angleP);
```

```
% Aplicar rotacionn y construir el resultado
L_=rotor*aux;
L=set_round([reshape(L_(1:2:2*n,:))',n*p,1) reshape(L_(2:2:2*n,:))',n*p,1]]);
```

### A.3 Cálculo de Border-E y Border-P

```

function L = get_borderE( BorderP, Space, vectE )
% Obtener el conjunto de puntos ya marcados tal que:
% Segun E todos los vecinos estan marcados
% y al menos uno (pueden ser todos) esta en BorderP
% BorderP: Puntos con V(x)=n, se suponen ya marcados
% vectE: E-vectograma
% Space: Malla del espacio reducido

%% Aplicar el E-vectograma inverso a Border para obtener puntos tentativos
% Alerta: Como el E-vectograma es simetrico sera usado en vez del inverso
tmp = set_vectE(BorderP, vectE);
% Descartar duplicados y aquellos no marcados fuera de BorderP
dupli = xy2mat(tmp,Space);
nondupli = unique(dupli);
acceptable = logical(~(Space(nondupli)==-1));

%% Verificar vecinos de los puntos tentativos
tentative = mat2xy(nondupli(acceptable),Space);
m = size(tentative,1);
% Obtener vecinos
% Alerta: Orden invertido para reducir calculos
tentativeP = set_vectE(vectE,tentative);
% Conveter a indice lineal los vecinos
ind_ = xy2mat(tentativeP,Space);
% Reacomodar como matriz de vecinos
p = size(ind_,1);
ind = reshape(ind_,p/m,m);
% Indice de Puntos tentativos con Vecinos aceptables (todos marcados)
L_=all((Space(ind)+1));

%% Retornar BorderE
L = tentative(L_,:);

function L = get_borderP(BorderE, Space, step, angleP)
% Encontrar los puntos no marcados tal que una movida de P los lleva a BorderE

%% Enfoque: Aplicar el P-vectograma inverso directamente a BorderE

```



```

tmp=set_ivectP(BorderE,step,angleP);
aux=xy2mat(tmp,Space);
% Descartar duplicados
ind=unique(aux);
captured=logical(Space(ind)==-1);

% Retornar BorderP
L=mat2xy(ind(captured),Space);

```

## A.4 Cálculo del Valor del Juego

```

%% Valor del Juego
clear;
clc;

%% Ejemplo: Datos reales l=2, vP=2m/s, vE=1m/s, R=16/pi; 32undmalla=1m, t=0.25
% Crear variables
n=3601;
l=2;
vP=2;
vE=1;
R=16/pi;
dt=0.25;

undmalla=32;
zoneinc=1;

cardP=41;
cardE=0;

Space = gen_space(n,-1);

% Generar Vectogramas
[vectP angleP] = gen_vectP(vP, R, dt, undmalla, cardP);
[vectE angleE] = gen_vectE(vE, dt, undmalla, 0);

% Generar zona de captura
Zone=get_capturezone(l,undmalla,zoneinc);
Zone=set_round(Zone);

```

```

% V(x)=0 en la zona de captura
ind=xy2mat(Zone,Space);
Space(ind)=0;

% V(x)=1, P mueve primero
L2=get_borderP(Zone,Space, vP*dt*undmalla, angleP);
ind = xy2mat(L2,Space);
Space(ind)=1;

for i_ = 1:200
    L1=get_borderE(L2,Space,vectE);
    L2=get_borderP(L1,Space, vP*dt*undmalla, angleP);
    ind = xy2mat(L2,Space);
    Space(ind)=i_+1;
end

```

## A.5 Cálculo de Trayectorias Optimas

```

function [pathP, angles, pathE, pathx, lastV] =
    get_path( P, E, Space, radio, vectP, angleP, vectE )
% Devolver las trayectorias optimas en el espacio reducido

pathP = [];
angles = [];
pathE = [];

% Calcular x en el espacio reducido
x=E-P;
pathx = x;

% V(x) nos permite saber de antemano cuantas movidas deben ser realizadas
% Si no esta en area de captura se procede
total = xy2mat(x, Space);
v=Space(total);

% Suponemos que todo el vectograma tiene el mismo avance
step=get_norms(vectP(1,:));
lastV=0;

if v>0
    nP=size(vectP,1);

```

```

nE=size(vectE,1);
value=zeros(nP,nE);
normx=zeros(nP,nE);

for steps=v:-1:0
% Enfoque: Teoria de juegos, Matriz de payoffs de suma cero
  tmpP = set_vectP(x, step, angleP);
  for iP=1:nP
    pointP=tmpP(iP,:);
    tmpE = set_vectE(pointP,vectE);
    k = xy2mat( tmpE, Space);
    value(iP,:)=Space(k);
    normx(iP,:)=abs(tmpE(:,1));
  end

  minE=min(value); % Minimos por columnas
  maxP=max(value,[],2); % Maximos por filas, maxE es una columna

  vE=max(minE);
  vP=min(maxP);
% Seleccionar el saddle point que estabilice mejor a x
  if (vP==vE) \QTR{it}{% Es saddle point}
    tentE=find(minE==vE);
    tentP=find(maxP==vP)';

    lE=length(tentE);
    lP=length(tentP);
    if (lE>1)|| (lP>1)
      % Calcular pares de estrategias y los x a estabilizar
      pairs=[];
      absx=[];
      for indP=tentP
        for indE=tentE
          if value(indP,indE)==vP
            pairs=[pairs; indP indE ];
            absx= [absx; normx(indP,indE)];
          end
        end
      end
      end
      minx=min(absx);
      posx=find(absx==minx);
      selected=ceil(length(posx).*rand);
      finalP=pairs(posx(selected),1);
      finalE=pairs(posx(selected),2);
    end
  end
end

```

```

        else
            finalP = tentP;
            finaleE = tentE;
        end
    else % Si no hay saddle point randomizar
        indP=1:nP;
        indE=1:nE;
        [finalP,finaleE]=mixed(indP,indE,value);
    end
    x=set_vectE(tmpP(finalP,:),vectE(finaleE,:));
    pathP = [pathP; vectP(finalP,:)];
    angles = [angles; get_angle(vectP(finalP,:))];
    pathE = [pathE; vectE(finaleE,:)];
    pathx=[pathx; x];

    % Por si ocurre captura antes de tiempo
    if norm(x)<=radio
        lastV=steps;
        break;
    end
end
end
end

function [finalindP,finalindE]=mixed( indexP, indexE, value)
% A partir de la matriz de utilidades
% Descartar por dominancia filas y columnas,
% Devolver estrategias al random con igual probabilidad

valuerand=value;
valuerand=sortrows(valuerand);
indrow=1;
while indrow<size(valuerand,1)
    a=valuerand(indrow,:);
    b=valuerand(indrow+1,:);
    % P busca minimizar, descartar estrategias con valores mayores
    if a<=b
        valuerand(indrow+1,:)=[];
        indexP(indrow+1)=[];
    else
        indrow = indrow+1;
    end
end
end
end

```

```

valuerand=sortrows(valuerand');
indcol=1;
while indcol<size(valuerand,1)
    a=valuerand(indcol,:);
    b=valuerand(indcol+1,:);
    % E busca maximiar, descartar estrategias con valores menores
    if a>=b
        valuerand(indcol+1,:)=[];
        indexE(indcol+1)=[];
    else
        indcol = indcol+1;
    end
end
finalindP=indexP(ceil(length(indexP).*rand));
finalindE=indexE(ceil(length(indexE).*rand));
end

```

```

function [pointsP, pointsE] = get_truepath( P, E, pathP, angleP, pathE )
% En funcion de los puntos P y E mas las trayectorias obtenidas calculara
% la sucesion de puntos en el espacio realista

```

```

% Obtener la cantidad de movidas que se haran
lP=size(pathP,1);
lE=size(pathE,1);
pointsP=zeros(lP+1,2);
pointsE=zeros(lE+1,2);

pointsP(1,:)=P;
pointsE(1,:)=E;

stepsP = pathP';
stepsE = pathE';

% Calcular puntos en base a los vectogramas
for k_=1:lP
    pointsP(k_+1,:)=pointsP(k_,:)+stepsP(:,k_）';
    % Efecto de rotacion debido a las movidas de P
    stepsP=get_rot(angleP(k_))*stepsP;
    stepsE=get_rot(angleP(k_))*stepsE;
    if k_ == lP
        break;
    end
    pointsE(k_+1,:)=pointsE(k_,:)+stepsE(:,k_）';

```

```
end
    pointsE(end,:)=pointsE(end-1,:);
```

## A.6 Utilitarios

```
function rot = get_rot(angle)
% Devuelve la matriz de rotacion de angle grados
% Si angle es un arreglo rot es una matriz diagonal de bloques
% los bloques en la diagonal seran las respectivas matrices de rotacion 2x2

n=length(angle);
matrix_ = cell(n,1);
for i_=1:n
    matrix_{i_} = [ cos(angle(i_)) -sin(angle(i_));
                  sin(angle(i_))  cos(angle(i_))];
end
rot=blkdiag(matrix_{1:n});
clear matrix_;
```

```
function L=xy2mat(Points,Space)
% Convertir coordenadas en formato xy a indice lineal para la matriz

% Asumiendo siempre Space cuadrado solo consideramos n
n=size(Space,1);

offset = get_offset(Space);

% invertir Fila=y, Columna=x y aplicar el offset
i_=(n+1)-(Points(:,2)+offset);
j_=Points(:,1)+offset;

L=sub2ind(size(Space),i_,j_);
```

```
function L=mat2xy(Points,Space)
% Convertir coordenadas en indice lineal de la matriz a formato xy
%Asumiendo siempre Space cuadrado solo consideramos n
n=size(Space,1);

offset = get_offset(Space);
```

```
% Indice lineal a puntos
[i_,j_] = ind2sub(size(Space),Points);
% Invertir Fila=y, Columna=x; luego aplicarle el offset
L=[ j_, (n+1-i_) ]-offset;
```