



# **Generación de Trayectorias de Curvatura Continua para el Seguimiento de Líneas basado en Visión Artificial**

Tesina Máster en Automática e Informática Industrial

*Departamento de Ingeniería de Sistemas y Automática (DISA)*  
*Departamento de Ingeniería de Sistemas y Computadores (DISCA)*



Autor: D. Vicent Girbés Juan

Director: Dr. Leopoldo Armesto Ángel  
Codirector: Dr. Josep Tornero Montserrat

Diciembre 2010, Valencia



Para M<sup>a</sup> Ángeles, por su apoyo incondicional  
y por los momentos buenos y malos  
que hemos compartido juntos.



# Resumen

La presente Tesina de Máster tiene como objetivos fundamentales el estudio de los métodos existentes y el desarrollo de nuevas técnicas para la generación de trayectorias de curvatura continua en el campo de la robótica móvil. El estudio se centra en la aplicación de dichas técnicas al problema del seguimiento de línea con restricciones cinemáticas y dinámicas, mediante la utilización de sistemas de visión artificial.

Concretamente, en la Tesina se realiza un amplio estudio al respecto de las diversas técnicas para la generación de caminos y concretamente aquellas que afrontan la problemática de generar trayectorias con curvatura continua, por el hecho de introducir mejoras tanto para el propio control del vehículo y estimación, como para la mejora en la seguridad y confortabilidad para conductores y pasajeros, en su caso.

En la Tesina también se estudian aquellos trabajos de guiado automático de vehículos, especialmente los basados en sistemas de visión artificial, en los que tienen aplicación directa los desarrollos realizados en esta Tesina, esto es, generación de maniobras específicas tales como adelantamientos, aparcamientos, etc.

La Tesina desarrolla un método original para la generación de trayectorias con curvatura continua en el que se consideran las restricciones cinemáticas y dinámicas del propio vehículo, con objeto de mejorar la seguridad y confortabilidad de la conducción. El método relaciona todos estos aspectos con aceleraciones y *jerks* máximos y mínimos admisibles, que acaban traducándose a mejores adherencias de las ruedas al firme. El método tiene aplicación tanto para la generación *off-line* como planificador de trayectorias como *on-line* como algoritmo de control cinemático.

El trabajo de investigación se concreta, cara a su validación, en la problemática del seguimiento de líneas, o más específicamente de caminos, en los que utilizando un sistema de visión artificial se detecta el camino a seguir a partir de las diversas líneas pintadas en el suelo. Mediante dicho sistema de visión artificial se determina un punto objetivo a alcanzar, así como su curvatura, basada en la estimación de la curvatura del propio camino que sirve como información de entrada para el método de generación de trayectorias continuas.

Durante el transcurso de este trabajo de investigación y con el objetivo de complementar y verificar el correcto funcionamiento de los algoritmos desarrollados, se han realizado implementaciones sobre vehículos reales, así como simulaciones previas. Las experiencias reales se han realizado sobre una carretilla elevadora industrial previamente automatizada y preparada para su control automático. También se ha utilizado la plataforma robótica *Pioneer 3DX*, típicamente utilizada en numerosos laboratorios de investigación y centros docentes.

Entre los campos de la robótica móvil relacionados con esta Tesina destacan los siguientes:

- Generación de trayectorias (*Path Generation*)
- Planificación de trayectorias (*Path Planning*).
- Caminos de curvatura continua (*Continuous-Curvature Path*).
- Control cinemático (*Kinematic Control*).
- Seguimiento de líneas (*Line Tracking*).
- Visión artificial (*Computer Vision*).

Queda fuera del ámbito de la presente Tesina, el estudio y desarrollo de técnicas de evitación de obstáculos, así como los métodos de planificación de movimientos que permitan llevar de forma global al robot desde su posición actual al objetivo establecido.

# Índice General

<b>CAPÍTULO 1. Introducción</b> .....	13
<b>1.1. PLANTEAMIENTO DEL PROBLEMA</b> .....	15
<b>1.2. OBJETIVOS DE LA TESINA</b> .....	18
<b>CAPÍTULO 2. Estado del Arte</b> .....	19
<b>2.1. INTRODUCCIÓN</b> .....	21
<b>2.2. GENERACIÓN DE TRAYECTORIAS DE CURVATURA CONTINUA</b> .....	22
2.2.1. TRAYECTORIAS SUAVES EN NAVEGACIÓN AUTÓNOMA .....	22
2.2.2. TRAYECTORIAS SUAVES CON RESTRICCIONES CINEMÁTICAS Y DINÁMICAS .....	33
2.2.3. TRAYECTORIAS SUAVES EN APLICACIONES DE CONDUCCIÓN .....	40
2.2.4. TRAYECTORIAS SUAVES EN OTRAS APLICACIONES .....	44
2.2.5. MÉTODOS DE APROXIMACIÓN DE CLOTOIDES .....	45
2.2.6. CONCLUSIÓN .....	53
<b>2.3. EL PROBLEMA DEL SEGUIMIENTO DE LÍNEA</b> .....	54
2.3.1. DETECCIÓN Y MODELADO DEL CAMINO .....	54
2.3.2. EL CONTROL EN EL PROBLEMA DE SEGUIMIENTO DE CAMINOS .....	59
2.3.3. DETECCIÓN DE CARRETERA Y OBSTÁCULOS .....	66
2.3.4. DETECCIÓN DE MÚLTIPLES LÍNEAS .....	71
2.3.5. SISTEMAS AVANZADOS QUE INTEGRAN DETECCIÓN DE CAMINOS Y CONTROL DEL VEHÍCULO .....	73
2.3.6. CONCLUSIÓN .....	79
<b>CAPÍTULO 3. Generación de Trayectorias de Curvatura Continua</b> .....	81
<b>3.1. INTRODUCCIÓN</b> .....	83
<b>3.2. CONTROL CINEMÁTICO</b> .....	83
3.2.1. SEGUIMIENTO DE TRAYECTORIAS .....	84
3.2.2. SEGUIMIENTO DE CAMINOS .....	85
3.2.2.1. <i>Seguimiento de Líneas</i> .....	87
3.2.3. CONTROL CON REGULADORES CONVENCIONALES .....	88
3.2.3.1. <i>Regulador PID</i> .....	88
3.2.3.2. <i>Regulador "Fuzzy" o Borroso</i> .....	89
<b>3.3. RS-PATH</b> .....	90
<b>3.4. CLOTOIDE</b> .....	91
3.4.1. PROPIEDADES .....	92
3.4.2. MÉTODOS DE APROXIMACIÓN .....	94

<b>3.5. ELEMENTARY PATH .....</b>	<b>101</b>
<b>3.6. BIELEMENTARY PATH .....</b>	<b>102</b>
<b>3.7. CONTINUOUS CURVATURE TURN (CC-TURN) .....</b>	<b>103</b>
<b>3.8. CONTINUOUS CURVATURE PATH (CC-PATH).....</b>	<b>103</b>
<b>3.9. TÉCNICA PARA LA GENERACIÓN OFF-LINE DE TRAYECTORIAS DE CURVATURA CONTINUA, PARA EL PROBLEMA DEL SEGUIMIENTO DE LÍNEA CON CURVATURA Y BRUSQUEDAD ACOTADAS.....</b>	<b>108</b>
3.9.1. PLANTEAMIENTO DEL PROBLEMA .....	108
3.9.2. SINGLE CONTINUOUS CURVATURE PATH (SCC-PATH).....	111
3.9.3. DOUBLE CONTINUOUS CURVATURE PATH (DCC-PATH) .....	115
3.9.4. ANÁLISIS .....	118
3.9.5. CONCLUSIONES .....	125
<b>3.10. TÉCNICA PARA EL CONTROL CINEMÁTICO ON-LINE MEDIANTE LA GENERACIÓN DE TRAYECTORIAS DE CURVATURA CONTINUA, PARA EL PROBLEMA DE SEGUIMIENTO DE LÍNEA CON CURVATURA Y BRUSQUEDAD ACOTADAS .....</b>	<b>126</b>
3.10.1. CONTINUOUS CURVATURE PATH CONTROL .....	127
3.10.2. ANÁLISIS.....	130
3.10.3. CONCLUSIONES.....	135
<b>CAPÍTULO 4. Detección de Líneas mediante Sistemas de Visión Artificial .....</b>	<b>137</b>
<b>4.1. INTRODUCCIÓN .....</b>	<b>139</b>
<b>4.2. TÉCNICAS DE VISIÓN POR COMPUTADOR PARA LA DETECCIÓN DE LÍNEAS .....</b>	<b>139</b>
4.2.1. <i>Regresión Lineal Simple</i> .....	142
4.2.2. <i>Regresión Lineal Ortogonal</i> .....	144
4.2.3. <i>Regresión Lineal Ortogonal Robusta. Método RANSAC</i> .....	145
4.2.4. <i>Transformada de Hough</i> .....	147
<b>4.3. ALGORITMO DE DETECCIÓN DE LNEAS PARA SISTEMAS DE VISIÓN (CON RECURSOS ILIMITADOS).....</b>	<b>149</b>
<b>4.4. ALGORITMO DE DETECCIÓN DE LÍNEAS PARA SISTEMAS DE VISIÓN EMBEBIDOS (CON RECURSOS LIMITADOS).....</b>	<b>159</b>
<b>CAPÍTULO 5. Implementación de las Técnicas Propuestas.....</b>	<b>167</b>
<b>5.1. SISTEMA DE VISIÓN ACTIVA .....</b>	<b>169</b>
<b>5.2. EXPERIMENTACIÓN CON CMUCAM3 Y AGV INDUSTRIAL .....</b>	<b>170</b>
5.2.1. PROBLEMA DE SEGUIMIENTO DE LÍNEAS.....	176
5.2.1.1. <i>Tracking de la Línea</i> .....	177
5.2.1.2. <i>Seguimiento de la Línea</i> .....	180

---

<b>5.3. EXPERIMENTACIÓN CON PLAYER/STAGE .....</b>	<b>180</b>
<b>5.4. EXPERIMENTACIÓN CON PLAYER Y ROBOT PIONEER 3DX.....</b>	<b>182</b>
<b>CAPÍTULO 6. Conclusiones y Trabajos Futuros .....</b>	<b>185</b>
<b>6.1. CONCLUSIONES .....</b>	<b>187</b>
<b>6.2. TRABAJOS FUTUROS .....</b>	<b>188</b>
<b>Bibliografía.....</b>	<b>191</b>
<b>ANEXOS.....</b>	<b>197</b>
<b>ANEXO 1. Métodos de Generación de Trayectorias.....</b>	<b>199</b>
<b>ANEXO 2. Control de Robots Móviles .....</b>	<b>205</b>
<b>ANEXO 3. Soluciones Industriales para AGVs.....</b>	<b>217</b>
<b>ANEXO 4. Proceso de Calibración .....</b>	<b>227</b>
<b>ANEXO 5. Protocolo de Comunicación Serie .....</b>	<b>233</b>
<b>ANEXO 6. Artículos Publicados .....</b>	<b>241</b>



# Índice de Figuras

Figura 1. (a) Patinete automático <i>Segway</i> . (b) Robot escoba <i>Roomba</i> . (c) AGV industrial. ....	15
Figura 2. Ejemplo de <i>RS-path</i> . ....	16
Figura 3. Ejemplo de <i>CC-path</i> . ....	16
Figura 4. Ejemplo de <i>path</i> que combina tramos rectos, circulares y transiciones (t0-t1, t2-t3, t4-t5).....	23
Figura 5. Curvas simples con $\alpha = \pi$ . ....	24
Figura 6. Distintos tipos de trayectoria de curvatura continua. ....	24
Figura 7. Estructura de control del vehículo autónomo. ....	25
Figura 8. Ejemplo de curva: (a) Clotoide y (b) anti-Clotoide. ....	25
Figura 9. Ejemplos de: (a) <i>turn</i> y (b) <i>loop</i> . ....	26
Figura 10. Ejemplos de: (a) <i>cusp</i> interno y (b) <i>cusp</i> externo. ....	26
Figura 11. Perfiles de la aceleración centrípeta (izquierda) y el <i> jerk</i> (derecha) para dos trayectorias distintas, una con velocidad variable (línea continua) y otra con velocidad constante (línea discontinua).....	26
Figura 12. Ejemplos de distintas trayectorias seguidas por el vehículo en un entorno lleno de obstáculos.....	27
Figura 13. <i>Elementary path</i> . ....	27
Figura 14. BiElementary Path. ....	28
Figura 15. <i>CC-turn</i> . ....	29
Figura 16. <i>SCC-path</i> (izquierda) y su perfil de curvatura (derecha). ....	29
Figura 17. Trayectoria generada con <i>Dubins' curves</i> (línea fina) y con <i>SCC-paths</i> (línea gruesa). ....	30
Figura 18. Ejemplo de trayectoria generada por el planificador propuesto en este artículo: <i>Landmarks</i> (izquierda) y traza del vehículo (derecha). ....	30
Figura 19. Ruta <i>v-shape</i> . ....	31
Figura 20. Comparación de los métodos de generación de rutas suaves: (a) es el conjunto de puntos de referencia para obtener un camino suave, (b) es el camino suave obtenido mediante interpolación de una spline cúbica, (c) es el camino obtenido mediante una <i>B-spline</i> cúbica y (d) es el resultado obtenido mediante el uso de una Bezier cúbica.....	32
Figura 21. Los 4 caminos posibles. La candidata a ser la trayectoria de mínimo tiempo está coloreada de azul. ....	33
Figura 22. Longitud $L$ correspondiente al gradiente mínimo relativo frente al radio de curvatura $R$ . ....	35
Figura 23. Gráfica de relación entre el parámetro de la Clotoide $A$ y el radio de curvatura $R$ . ....	35
Figura 24. Ejemplo de problema complejo de planificación resuelto por el método propuesto. ....	38
Figura 25. (a) Perfil de curvatura discontinuo de un <i>RS-path</i> . (b) Perfil de curvatura continuo de un <i>CC-path</i> . El tramo entre $a$ y $b$ es un arco de Clotoide.....	39
Figura 26. Posiciones inicial y final (izquierda). Solución del planificador sin restricciones holonómicas (centro). Solución final optimizada con restricciones holonómicas (derecha). ....	40
Figura 27. Trayectoria calculada sin obstáculos. ....	41
Figura 28. Trayectoria libre de colisiones. ....	41
Figura 29. Instantáneas de la trayectoria libre de colisiones. ....	41
Figura 30. Problema de cambio de carril con obstáculos móviles. ....	41
Figura 31. BiElementary path aplicado al problema de cambio de carril. ....	42
Figura 32. Trayectorias factibles obtenidas por el planificador. ....	42
Figura 33. Maniobra de cambio de carril con evitación de obstáculo. ....	43
Figura 34. Ejemplo de aparcamiento en paralelo. ....	43
Figura 35. Cuatro <i>CC-turns</i> completos y diferentes en el plano $XY$ . ....	43
Figura 36. Ejemplo de trayectoria formada por 4 Clotoides simétricas para resolver el problema de cambio de carril. ..	44
Figura 37. Modelo de 2 Clotoides contiguas. ....	49
Figura 38. Simulación del modelo de Clotoides contiguas. Geometría de carretera (izquierda) y curvatura (derecha). ....	49
Figura 39. (a) Clotoide modelo, (b) Clotoide con errores angulares y (c) Clotoide con errores de offset lateral. ....	50

Figura 40. (a) Estimación del modelo de Clotoide tras la fusión sensorial del modelo de parámetros clásico de la Clotoide. (b) Estimación del modelo de Clotoide transformado al espacio rango-offset. ....	50
Figura 41. Comparativa entre aproximaciones: (a) Taylor y (b) Hermite. ....	51
Figura 42. Ejemplo de unión recta y círculo mediante una Clotoide modelada como un arco de <i>spline</i> . ....	51
Figura 43. Aproximación de los puntos de Fresnel con una curva R-Bezier de orden $N=11$ . ....	52
Figura 44. Poses Inicial $(0, 0, 0)$ y Final $(0, 50, 1.2)$ y ruta generada. ....	53
Figura 45. Proceso de detección de carretera con el sistema RALPH. ....	55
Figura 46. Proceso de estimación de la curvatura mediante comparación con otras imágenes transformadas. ....	55
Figura 47. 2 ejemplos de detección de bordes con el método propuesto. En la parte inferior se observan los perfiles de nivel de gris de ambas imágenes. ....	55
Figura 48. Datos tomados por los sensores combinados mediante fusión sensorial. ....	56
Figura 49. Ejemplos de indicadores reflectantes. ....	56
Figura 50. Ejemplo de detección de delineadores. ....	57
Figura 51. Clotoide y función cúbica aplicadas para estimar la posición 3D de los delineadores. ....	57
Figura 52. Etapa de reconocimiento del sistema neuronal seguidor de caminos. ....	58
Figura 53. Sistema de visión para navegación por carreteras (Vision-based Road Navigation System: VICNAS). ....	58
Figura 54. El regulador se estructura en tres etapas: realimentación, pre-visualización y planificación de ganancia. ....	59
Figura 55. Comparación entre dos métodos de seguimiento de curvas de tipo recta. ....	60
Figura 56. Esquema completo del sistema de identificación y seguimiento de líneas. ....	61
Figura 57. Arquitectura de control con Estimación Multi-Frecuencia de la postura. ....	62
Figura 58. Arquitectura de control con Filtro de Kalman Extendido Multi-Frecuencia (MR-EKF). ....	62
Figura 59. Comparativa de ambas técnicas con perturbación de entrada constante. ....	62
Figura 60. Ejemplo de curva de una carretera. ....	63
Figura 61. Perfiles de curvatura (izquierda) y <i>sharpness</i> (derecha) de la trayectoria presentada en la Figura 60. ....	63
Figura 62. Estructura del algoritmo de estimación de los parámetros del camino. ....	63
Figura 63. Clotoide estimada y medida. ....	64
Figura 64. Resultados de los distintos reguladores utilizados. ....	64
Figura 65. Vehículo utilizado en la experimentación. ....	65
Figura 66. Ruta real dibujada en el pavimento mediante un sistema de pintado montado sobre el robot (izquierda). Trayectoria de referencia y trayectoria seguida por el robot (derecha), con control proporcional $Kc = -1$ y velocidad $v = 0.5 \text{ m/s}$ ( - - - escalón unitario (ruta deseada), — ruta resultante). ....	65
Figura 67. Reconocimiento de carretera mediante la técnica IPM para deshacer la perspectiva. ....	66
Figura 68. Esquema del sistema recursivo de control de trayectoria. ....	66
Figura 69. Detección de obstáculos mediante técnicas de simetría. ....	67
Figura 70. Planificación basada en visión utilizando $G^2$ -spline. ....	68
Figura 71. Ejemplo de reconocimiento de las líneas de un carril de carretera. ....	68
Figura 72. Ejemplos de reconocimiento de carril y detección de obstáculos. ....	69
Figura 73. Detección de carril, que se modela como una superficie 3D, definida mediante Clotoides. ....	69
Figura 74. Ejemplos de detección de línea y modelado de carretera. ....	70
Figura 75. Detección de las líneas de los carriles de la carretera. ....	71
Figura 76. Estimación del modelo de una línea paralela a una Clotoide, con $c_0 = -0.01$ , $c_1 = 0.0006$ y $l = 60$ . ....	72
Figura 77. Explicación de la estrategia predictiva del movimiento en un escenario de carretera urbana. ....	72
Figura 78. Resultados de la detección concurrente de múltiples líneas. ....	73
Figura 79. Ejemplo de identificación de la línea central de la carretera mediante el método probabilístico propuesto. ...	73
Figura 80. Arquitectura general del sistema <i>on-board</i> de visión DIPLODOC. ....	74
Figura 81. Resultados del reconocimiento de carretera del algoritmo. ....	75
Figura 82. Geometría de la carretera (izquierda) y curvatura en función de la longitud del arco (derecha). ....	75
Figura 83. El vehículo utilizado en los experimentos es un modelo Ford E-350 Econoline, el cual lleva instalado un sistema LADAR para obtener los datos utilizados en la navegación. ....	75

Figura 84. Para detectar el camino pintado y los bordes de la carretera el sistema utiliza varios sensores que aportan streams de datos asíncronos y heterogéneos (amarillo) y estimar la línea central de múltiples carriles (cian). .....	76
Figura 85. Detector de las líneas pintadas en la carretera: (a) Imagen original. (b) Imagen segmentada por filas mediante un filtro horizontal. (c) Imagen con los máximos locales de la respuesta del filtro y sus orientaciones, en amarillo se observan las orientaciones perpendiculares a cada máximo. (d) Los máximos cercanos se conectan mediante <i>splines</i> cúbicas de Hermite. ....	76
Figura 86. Vista aérea de los caminos recorridos de forma autónoma por el vehículo. Regiones en las que el seguimiento es preciso, robusto y seguro (azul). Zonas en las que la seguridad es baja (rojo). Algunas de las zonas de seguridad baja corresponden a intersecciones. ....	77
Figura 87. (a) Imagen típica de color de un camino sucio. (b) Error de desplazamiento $\epsilon(\mathbf{y})$ del centro del vehículo cuando el seguimiento se realiza con un único segmento de Clotoide (línea roja) en una carretera (gris) a partir del área de visión de la cámara (cian). ....	77
Figura 88. Generación sucesiva de segmentos estáticos de carretera (áreas marrones), con la predicción de las covarianza el error (elipses azules) y la estimación de los segmentos locales de Clotoide (línea gris). ....	78
Figura 89. (a) Error en la detección de la carretera mediante el uso del EKF. (b) Detección correcta de la carretera tras la utilización del filtro de partículas propuesto. ....	78
Figura 90. Bucle de control para seguimiento de trayectorias. ....	84
Figura 91. Simulación de control de seguimiento de trayectorias. ....	84
Figura 92. Modelo para el control de persecución pura. ....	85
Figura 93. Esquema para calcular la curvatura de la trayectoria con persecución pura. ....	86
Figura 94. Modelo de seguimiento de línea. ....	87
Figura 95. Esquema de control de posición con PID (izquierda) y referencias cinemáticas (derecha). ....	88
Figura 96. Esquema de control de ruedas con PID para un modelo diferencial de robot. ....	89
Figura 97. Ejemplo de evaluación de las reglas: (a) en función del ángulo o (b) en función de la distancia. ....	89
Figura 98. Ejemplo de <i>RS-path</i> . ....	90
Figura 99. Ejemplo de curva Clotoide. ....	91
Figura 100. Ejemplo de curva anti-Clotoide. ....	91
Figura 101. Aproximación de los puntos de Fresnel con una curva Bezier de orden $N=11$ . ....	97
Figura 102. Aproximación de los puntos de Fresnel con una curva R-Bezier de orden $N=11$ . ....	99
Figura 103. Aproximación de los puntos de Fresnel con Bezier y R-Bezier para $N=11$ en el rango $[0 - \pi/2]$ . ....	99
Figura 104. Detalle de la diferencia entre la aproximación Bezier y la R-Bezier. ....	100
Figura 105. Segmento clotoidal. ....	101
Figura 106. <i>Elementary path</i> . ....	101
Figura 107. <i>BiElementary Path</i> . ....	102
Figura 108. <i>CC-turn</i> . ....	103
Figura 109. <i>SCC-path</i> (izquierda) y su perfil de curvatura (derecha). ....	104
Figura 110. Perfiles de curvatura: (a) <i>RS-path</i> y (b) <i>SCC-path</i> . El tramo $\mathbf{a} - \mathbf{b}$ es un arco de Clotoide. ....	104
Figura 111. Modelo de tipo coche del vehículo: $\mathbf{R} = (x, y)$ es el punto de referencia y $\theta$ es la orientación del vehículo, $\phi$ es el ángulo de orientación de las ruedas y $\mathbf{b}$ es la separación entre los ejes de las ruedas. ....	105
Figura 112. Caso general de <i>CC-turn</i> . ....	106
Figura 113. Casos particulares de <i>CC-turn</i> : $\delta = 0$ y $0 < \delta < \delta_{min}$ . ....	107
Figura 114. Caso particular de <i>CC-turn</i> : $\delta_{min} + \pi \leq \delta < 2\pi$ . ....	107
Figura 115. Espacio de soluciones con (a) dos Clotoides y (b) opcionalmente un arco circular para compensar el ángulo de deflexión y un segmento de línea para compensar el error de <i>bias</i> . ....	110
Figura 116. Espacio de soluciones: (a) con cuatro Clotoides y (b) opcionalmente arcos circulares y segmentos de línea para compensar el ángulo de deflexión y el error de <i>bias</i> respectivamente. ....	110
Figura 117. Ejemplo representativo de un <i>SCC-path</i> . ....	111
Figura 118. Selección de curvas de curvatura para distintos valores de $\sigma$ . ....	112
Figura 119. <i>Paths</i> de Clotoides en los cuatro cuadrantes. ....	113

Figura 120. Ejemplo representativo de un <i>DCC-path</i> .	115
Figura 121. Perfil de curvatura del <i>DCC-path</i> de la Figura 120.	115
Figura 122. Seguimiento de línea con $dh > 0, \delta > 0, \kappa A > 0$ y $\kappa B > 0$ .	118
Figura 123. Seguimiento de línea con $dh > 0, \delta > 0, \kappa A < 0$ y $\kappa B > 0$ .	119
Figura 124. Seguimiento de línea con $dh < 0, \delta < 0, \kappa A < 0$ y $\kappa B < 0$ .	119
Figura 125. Seguimiento de línea con $dh < 0, \delta < 0, \kappa A > 0$ y $\kappa B < 0$ .	120
Figura 126. Seguimiento de línea con $dh > 0$ pero de valor pequeño, $\delta > 0, \kappa A < 0$ y $\kappa B > 0$ .	120
Figura 127. Seguimiento de línea con $dh < 0, \delta = 0, \kappa A < 0$ y $\kappa B > 0$ .	121
Figura 128. Seguimiento de línea con $dh > 0, \delta < 0, \kappa A > 0$ y $\kappa B > 0$ .	121
Figura 129. Seguimiento de línea con $dh < 0, \delta < 0, \kappa A > 0$ y $\kappa B > 0$ .	122
Figura 130. Máximo valor de aceleración normal $\alpha N$ para distintos valores de $\alpha A$ y $\alpha B$ .	123
Figura 131. Máximo valor de <i> jerk </i> normal $jN$ para distintos valores de $\alpha A$ y $\alpha B$ .	123
Figura 132. Máximo valor de <i> jerk </i> tangencial $jT$ para distintos valores de $\alpha A$ y $\alpha B$ .	123
Figura 133. Longitud del <i> path </i> para distintos valores de $\alpha A$ y $\alpha B$ .	124
Figura 134. Variación de la longitud del <i>DCC-path</i> frente a la variación del parámetro $\alpha C$ .	125
Figura 135. Variación de la curvatura del <i>DCC-path</i> frente a la variación del parámetro $\alpha C$ .	125
Figura 136. Ejemplo de <i>DCC-path</i> para control.	127
Figura 137. Perfil de curvatura del <i>DCC-path</i> de la Figura 136.	127
Figura 138. Ejemplo de <i>DCC-path</i> en el problema de control para configuraciones inicial $qA$ y final $qB$ arbitrarias.	130
Figura 139. Ejemplos de seguimiento de caminos basados en métodos de persecución pura con distintas configuraciones iniciales y para varios valores de <i>sharpness</i> , con $D=10m$ .	131
Figura 140. <i>Sharpness</i> , curvatura y ángulo de tangencia para las trayectorias de la Figura 139 (a).	132
Figura 141. <i>Sharpness</i> , curvatura y ángulo de tangencia para las trayectorias de la Figura 139 (b).	132
Figura 142. <i>Sharpness</i> , curvatura y ángulo de tangencia para las trayectorias de la Figura 139 (c).	133
Figura 143. <i>Sharpness</i> , curvatura y ángulo de tangencia para las trayectorias de la Figura 139 (d).	133
Figura 144. Maniobras típicas de conducción resueltas con <i>DCC-paths</i> : (a) cambio de carril y (b) adelantamiento.	134
Figura 145. Ejemplo de trayectoria generada mediante control con <i>DCC-paths</i> en un entorno con obstáculos.	134
Figura 146. Etapas de procesamiento de un sistema de visión artificial.	140
Figura 147. Error de estimación método de mínimos cuadrados.	143
Figura 148. Recta de regresión. Método mínimos cuadrados.	144
Figura 149. Representación de la recta por el método Distancia-Ángulo.	144
Figura 150. Regresión ortogonal. Variables consideradas.	144
Figura 151. Caracterización de puntos con RANSAC.	145
Figura 152. Conjunto de observaciones de partida, algoritmo RANSAC.	146
Figura 153. Resultado iteración intermedia, algoritmo RANSAC.	147
Figura 154. Resultado final. Estimación modelo con algoritmo RANSAC.	147
Figura 155. Ejemplo de transformada de Hough para reconocimiento de rectas.	148
Figura 156. Diagrama de flujo del procesamiento.	149
Figura 157. Ejemplos de imágenes de líneas: (a) y (b) originales en color, (c) y (d) transformadas a escala de gris.	150
Figura 158. Ejemplos de líneas tras aplicar un suavizado.	150
Figura 159. Ejemplos de líneas tras aplicar un umbralizado.	151
Figura 160. Ejemplos de líneas (a) y (b) tras realizar una erosión, (c) y (d) tras una dilatación.	152
Figura 161. Ejemplos de líneas en las que se ha detectado un ROI (rectángulo azul).	152
Figura 162. Ejemplos de líneas en las que se ha realizado una detección de bordes.	153
Figura 163. Ejemplos de líneas en las que mediante Hough se han detectado los bordes (líneas verdes), la línea central (rojo) y las intersecciones entre rectas (puntos azules).	155
Figura 164. Ejemplos de líneas proyectadas sobre el plano de tierra $z = 0$ .	155
Figura 165. Ejemplos de líneas proyectadas sobre la imagen (rojo) y estimación del camino mediante RBC (azul).	157
Figura 166. Ejemplo de carril de carretera: (a) imagen original y (b) imagen emborronada en escala de gris.	157

Figura 167. Ejemplo de carril de carretera: (a) imagen tras varias operaciones morfológicas y (b) imagen después del proceso de detección de bordes. ....	158
Figura 168. Ejemplo de carril de carretera: (a) imagen con el ROI (cuadro azul) y la región de interés resaltada en rojo, (b) resultado final con los puntos centrales (verde) y la estimación del camino con la RBC (azul). ....	158
Figura 169. Aproximación de los puntos de la carretera mediante una RBC cúbica. ....	158
Figura 170. Diagrama de flujo principal de la aplicación. ....	160
Figura 171. Diagrama de flujo de la función de eliminación de fondo mediante <i>blurring</i> . ....	162
Figura 172. Ejemplos de compensación de fondo. (a) y (c) imágenes originales con fondo gris claro y gris moteado, respectivamente. (b) y (d) resultado de la imagen con el fondo compensado. ....	162
Figura 173. Diagrama de flujo de la función de detección de bordes. ....	163
Figura 174. Representación de la recta en el espacio de distancia-ángulo. ....	164
Figura 175. Ejemplos de detección de líneas. ....	165
Figura 176. (a) Torre PTU para la CMUcam3. (b) Cámara IP con sistema PTZ. ....	169
Figura 177. AGV industrial utilizado en la experimentación de seguimiento de líneas. ....	170
Figura 178. Representación de la recta en el espacio de distancia-ángulo. ....	171
Figura 179. Experimento para la obtención de resultados que sirvan como <i>ground-truth</i> al utilizar el sistema de visión CMUcam3 junto con un robot Kuka KR/15 y una mesa giratoria. ....	172
Figura 180. (a) Ejemplo de imagen de línea con fondo gris claro. (b) Ejemplo de imagen con fondo gris moteado. ....	172
Figura 181. Parámetros angulares para calibración de la PTU. ....	173
Figura 182. Ejemplo de (a) TP y (b) FP en una situación similar de elevada complejidad. ....	175
Figura 183. Sensibilidad de la robustez ante errores de calibración. ....	175
Figura 184. Sensibilidad de la precisión ante errores de calibración. ....	176
Figura 185. Proceso de detección y seguimientos de línea. ....	178
Figura 186. Seguimiento de línea con $\rho = \mathbf{0}$ . ....	178
Figura 187. Seguimiento de línea con $\phi = \mathbf{0}$ . ....	179
Figura 188. Seguimiento de línea para valores arbitrarios de $\rho$ y $\phi$ . ....	179
Figura 189. Esquemas de funcionamiento de la plataforma Player para un robot real y para Stage. ....	181
Figura 190. Ejemplos de seguimiento de trayectoria realizados con la plataforma Player/Stage. ....	182
Figura 191. Robot Pioneer con diversos sensores: cámara, láser, sónar y encoders. ....	182
Figura 192. Cámara IP de la marca <i>Edimax</i> instalada en la parte superior del robot <i>Pioneer</i> . ....	183
Figura 193. Láser de telemetría de la marca <i>Hokuyo</i> instalado en la parte frontal del robot <i>Pioneer</i> . ....	183
Figura 194. Anillo de sensores de tipo sónar que incorpora el robot <i>Pioneer</i> . ....	183
Figura 195. Motor y encoder de una de las ruedas del robot <i>Pioneer</i> . ....	183
Figura 196. Ejemplo de mapa de carreteras en entorno industrial. ....	201
Figura 197. Descomposición celular regular. ....	202
Figura 198. Descomposición celular trapezoidal. ....	202
Figura 199. Descomposición celular por árboles octales. ....	202
Figura 200. Ejemplo de Grafo de Visibilidad. ....	203
Figura 201. Ejemplo de Diagrama de Voronoi. ....	203
Figura 202. Ejemplo de Campo de Potencial Artificial. ....	204
Figura 203. Varios modelos de robots con configuración diferencial. ....	208
Figura 204. Modelo de la configuración diferencial. ....	208
Figura 205. Modelo de la configuración diferencial con punto descentralizado. ....	210
Figura 206. Carretilla elevadora con configuración triciclo. ....	211
Figura 207. Modelo de la configuración triciclo. ....	212
Figura 208. Resultados de diferentes algoritmos de <i>Scan Matching</i> : MbICP (izq.), IDC (centro) e ICP (dcha.). ....	215
Figura 209. Estimación de la posición del robot: <i>Scan Matching</i> (verde) y odometría (rojo). ....	215
Figura 210. Sistemas de seguridad de un AGV. ....	220
Figura 211. Esquema de funcionamiento del sistema de guiado por cable magnético. ....	222

Figura 212. AGVs guiados mediante visión. ....	225
Figura 213. Sistemas de coordenadas de la torreta de la cámara. ....	229
Figura 214. Ejemplo de transformación mediante el método IPM. ....	232
Figura 215. Transformación mediante el método IPM para la cámara IP montada sobre el robot <i>Pioneer 3DX</i> . ....	232
Figura 216. Campos del telegrama de comunicación. ....	235
Figura 217. Proceso de comunicación del protocolo ....	235
Figura 218. Proceso de formación telegrama de comunicación desde PC. ....	236
Figura 219. Proceso de formación telegrama de comunicación desde CMUcam3 ....	236
Figura 220. Proceso de decodificación del telegrama ....	238
Figura 221. Aplicación del ordenador para visualización de la línea detectada ....	239

# **CAPÍTULO 1**

## **Introducción**



# 1. Introducción

## 1.1. Planteamiento del Problema

En la actualidad el campo de la automática y más concretamente el de la robótica están en expansión, gracias a los avances en la integración electrónica y la informática. Especialmente interesantes son los robots móviles, ya que se trata de vehículos capaces de moverse por el entorno de forma completamente autónoma.

La robótica móvil, sobre todo la de vehículos guiados con ruedas, ha avanzado mucho en las últimas décadas, tanto tecnológicamente como en su algorítmica. Por ese motivo cada vez son más los sistemas comerciales que incorporan los desarrollos realizados por los investigadores en el campo de la robótica móvil. En la Figura 1 se muestran algunas de las aplicaciones comercializadas más importantes relacionadas con la robótica móvil.



Figura 1. (a) Patinete automático *Segway*. (b) Robot escoba *Roomba*. (c) AGV industrial.

Por otro lado, entre los avances más importantes a nivel de investigación destacan las técnicas de auto-localización y construcción de mapas o SLAM (*Simultaneous Localization And Mapping*), basadas en técnicas de registro de datos y/o *Scan Matching*. Mediante dichas técnicas el robot es capaz de determinar su posición con bastante exactitud en entornos desconocidos a priori y a medida que explora el entorno es capaz de construir un mapa del mismo.

También se ha avanzado significativamente con objeto de dotar al robot de la capacidad de generar, planificar y decidir su propia trayectoria en función de las variables propias y las del entorno, de modo que sea capaz tanto de llegar a su objetivo como de evitar obstáculos. La diferencia de estas técnicas con respecto a otras como por ejemplo el SLAM, es el hecho de que normalmente se trabaja en entornos desconocidos, mientras que aquí se suele tener conocimiento total o parcial del entorno. Son muchas y muy variadas las técnicas utilizadas para la generación de trayectorias: Mapas de Carreteras (*Roadmap*), Grafos de Visibilidad, Diagramas de Voronoi, Descomposición Celular, Campos Potenciales, etc. No obstante, estas técnicas sólo permiten determinar un conjunto de puntos intermedios o configuraciones libres de colisión, pero no el camino que une dos o más configuraciones en su totalidad, es decir, como una trayectoria completa. Esto significa que dichos algoritmos sólo son capaces de unir los distintos puntos mediante segmentos de recta, de modo que el camino para ir de una configuración libre de colisión a la siguiente no queda claramente definido.

Existen muchas opciones para generar un camino, siendo la más sencilla la concatenación de líneas rectas y arcos de circunferencia (ver Figura 2), lo que se conoce como *RS-paths* (*Reeds and Shepp's paths*).

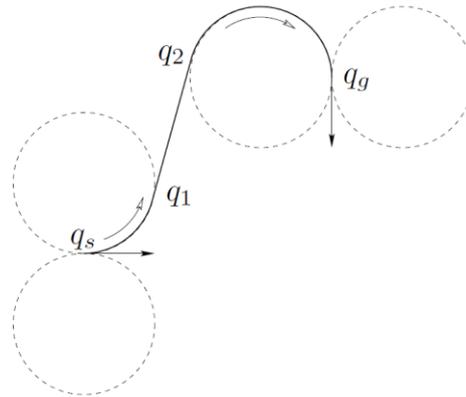


Figura 2. Ejemplo de *RS-path*.

Aunque la trayectoria generada mediante *RS-paths* es continua en su trazada en realidad no lo es para su curvatura, ya que en el cambio de trayectoria recta a circular existe una discontinuidad de la curvatura. Esto es cierto por el hecho de que se pasa de forma instantánea de una curvatura nula (recta) a otra de valor constante (arco circular). Dicha discontinuidad impone zonas de aceleración infinitas, provocando que el robot no siga la trayectoria de forma correcta, lo cual puede resultar inaceptable tanto en el problema de seguimiento de trayectorias como en el de auto-localización.

Por ese motivo los investigadores han estudiado diversas formas de generar trayectorias continuas, llegando a la conclusión de que para solucionar la discontinuidad lo ideal es la inclusión de tramos clotoidales, los cuales pueden combinarse con rectas y arcos circulares (ver Figura 3). Las Clotoides tienen la propiedad de que la variación de su radio de curvatura es constante, permitiendo una transición suave desde la línea hasta el círculo. Además de por sus características geométricas también conviene utilizar trayectorias de este tipo por el hecho de que el radio de la curva Clotoide es fácilmente trasladable a un parámetro controlable del vehículo como es el giro de las ruedas.

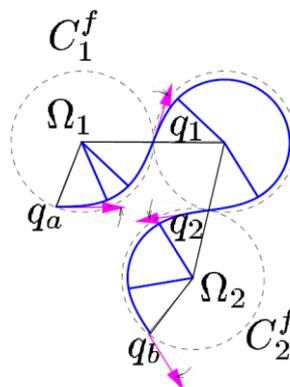


Figura 3. Ejemplo de *CC-path*.

El inconveniente de este tipo de trayectorias es el alto coste computacional que conlleva trabajar con ellas de forma exacta, lo cual hace más difícil su inclusión en el planificador de trayectorias del robot. No obstante, los últimos avances en este campo permiten la aproximación

de segmentos clotoïdales obteniendo un error despreciable con un coste computacional muy bajo, lo cual permite generar trayectorias continuas para la navegación de vehículos de forma rápida y sencilla, pudiéndose incluir fácilmente en los planificadores de trayectorias.

Otra área ampliamente desarrollada y utilizada en robótica móvil es la visión artificial (*Computer Vision*). Son muchas las aplicaciones que se pueden desarrollar mediante un sistema de visión, pero entre todas ellas destacaremos la del seguimiento de líneas o caminos. Muchos investigadores han desarrollado técnicas de detección de bordes para poder determinar con precisión el camino que debe seguir el robot. Una vez detectada la región de interés se puede aproximar el camino mediante tramos rectos o curvos, fáciles de parametrizar, de modo que se defina un modelo que sirva como referencia al vehículo autónomo. Por último, una vez detectado y modelado el camino se puede determinar un punto objetivo y aplicar una ley de control para reducir el error. Por lo tanto se podría dividir este tipo de aplicaciones en tres etapas:

- Detección de línea o camino (Visión u otro sistema sensorial).
- Modelado del camino y obtención de un punto objetivo (Modelado de curvas).
- Generación de la trayectoria y/o aplicación de la ley de control (Control cinemático).

Existen diversas técnicas de control cinemático, que garantizan estabilidad y convergencia, capaces de dotar al robot de la suficiente autonomía como para seguir una línea o un camino. Entre dichas técnicas cabe mencionar tanto las de seguimiento de trayectorias mediante punto descentralizado, como las de seguimiento de caminos o persecución pura (*Pure Pursuit*). El problema de estas técnicas es que no garantizan continuidad en la curvatura, puesto que la acción de control se obtiene aplicando una ley de control que depende del error entre la posición actual y la referencia a seguir, sin tener en cuenta la posición y la orientación actual de las ruedas. Además tampoco se genera una trayectoria o secuencia de acciones de control sino que se obtiene una acción de control única que se recalcula cada iteración. Por lo tanto, la implementación de este tipo de técnicas provoca que el robot tenga ligeros deslizamientos en las ruedas incrementándose los errores de odometría, los cuales en muchas aplicaciones son inaceptables o poco deseables.

Las técnicas anteriores tampoco tienen en cuenta la variación de la curvatura generada en el control, ni tampoco las restricciones cinemáticas y dinámicas del vehículo autónomo. Entre las principales restricciones cinemáticas destaca la limitación física que tiene un robot en el ángulo de giro de su rueda o ruedas directrices. Entre las principales limitaciones dinámicas (aceleraciones y sus derivadas) están las impuestas por motivos de seguridad, por ejemplo en vehículos industriales para que estos no vuelquen cuando están cargados, o de confortabilidad en vehículos que transportan pasajeros. Por ese motivo existen diversos trabajos de investigación que abordan dicha problemática que en muchos casos se pasa por alto, pero que en otros casos no puede hacerse.

## **1.2. *Objetivos de la Tesina***

El objetivo fundamental de la presente Tesina de Máster es el desarrollo de algoritmos que integren tanto la detección y el modelado de líneas o caminos, como la generación de trayectorias de curvatura continua, de modo que se tengan en cuenta restricciones tanto cinemáticas como dinámicas.

En vista de la ausencia de técnicas de control que garanticen la continuidad de la curvatura en el problema del seguimiento de líneas y dado que la utilización de Clotoides para la generación de trayectorias en robótica móvil nos ofrece diversas ventajas con respecto a cualquier otra curva, ya que al ser una trayectoria suave en todo su recorrido se minimiza el deslizamiento de las ruedas y con ello el error de odometría, se decide fusionar ambos ámbitos: el de generación de trayectorias continuas y el de seguimiento de líneas mediante sistemas de visión artificial.

En el Capítulo 2 se realiza un estudio del estado del arte tanto desde el punto de vista de la generación de trayectorias de curvatura continua como del problema del seguimiento de línea. Como en cualquier trabajo de investigación, en la presente Tesina hay una parte muy importante de búsqueda de información y comprensión de las distintas técnicas existentes, es por eso que en este capítulo se detallarán los aspectos más significativos de las dos principales áreas que se van a fusionar.

En el Capítulo 3 se abordará de forma detallada todo lo relativo a la generación de trayectorias de curvatura continua con restricciones cinemáticas y dinámicas. Se presentarán tanto las técnicas existentes como las nuevas técnicas desarrolladas en el presente trabajo de investigación, tanto desde el punto de vista de la planificación como del control.

El Capítulo 4 se centra en la explicación de las distintas técnicas de detección de líneas mediante sistemas de visión artificial, determinando sus principales características, así como sus ventajas e inconvenientes. Posteriormente se abordarán las técnicas implementadas desde un punto de vista genérico, tanto para sistemas de visión basados en computador con recursos ilimitados, como para sistemas embebidos con recursos limitados de procesamiento y memoria.

Los experimentos de validación de las técnicas y algoritmos desarrollados se explicarán en el Capítulo 5. Se mostrarán los resultados obtenidos en las implementaciones tanto para una plataforma de simulación robótica, como para un vehículo autónomo industrial y un robot móvil de investigación.

Finalmente, en el Capítulo 6 se extraen algunas conclusiones sobre las principales aportaciones de la Tesina, así como sobre los posibles trabajos futuros para el desarrollo de la Tesis Doctoral.

Además de los capítulos anteriores también se añaden varios anexos con la intención de complementar la documentación de la presente Tesina de Máster.

# **CAPÍTULO 2**

## **Estado del Arte**



## 2. Estado del Arte

### 2.1. Introducción

En robótica móvil existen básicamente dos tipos de entornos por los cuales puede moverse el robot: los escenarios conocidos o estructurados y los desconocidos o desestructurados. En ambos casos es muy importante generar trayectorias de curvatura continua para el control de movimiento del robot por el hecho de que ofrecen una serie de ventajas, como por ejemplo la reducción del deslizamiento de las ruedas y en consecuencia la reducción de los errores de odometría y de estimación. Además, al generar trayectorias suaves y eliminar las discontinuidades de la curvatura se reduce lo que se conoce como *sharpness* o brusquedad y por lo tanto el *jerk* o sobre-aceleración (derivada de la aceleración). Dicha magnitud física está directamente relacionada con las fuerzas normales al movimiento, las cuales, si no se limitan, pueden empeorar la confortabilidad del pasajero en vehículos que transportan personas o aumentar el riesgo de caídas de los vehículos industriales que transportan productos.

Por lo tanto, la intención principal del Capítulo 2 será la de definir con claridad las técnicas existentes en los distintos campos de investigación relacionados con la Tesina, resaltando sus características tanto positivas como negativas. No obstante, antes de centrarnos en la generación de trayectorias de curvatura continua y la detección de caminos mediante sistemas de visión, se va a introducir el concepto de generación de rutas o caminos como un espacio de configuraciones libres de colisión y también algunos conceptos clásicos de control cinemático en robótica móvil.

Son varios los aspectos que hay que considerar para hacer una buena planificación de movimientos a la hora de controlar un robot móvil. En primer lugar está el modelado del móvil y del entorno, necesario para responder a preguntas tales como ¿dónde estoy? y ¿qué me rodea? Después está el aspecto de detección de colisiones y evitación, que responde parcialmente a la pregunta ¿cómo llego a mi objetivo? Lo siguiente es la planificación del movimiento, que permite completar esta última pregunta desde una perspectiva global. Por ejemplo, en caso de moverse por zonas en las cuales no se ha estado previamente o bien aparecer obstáculos imprevistos, es necesario un razonamiento lógico, capaz de hacer que se tomen las decisiones adecuadas para alcanzar el objetivo final.

Dentro de este ámbito de la robótica están los algoritmos y técnicas que se basan en el conocimiento previo del entorno [1], [2]. Ya sea con modelos gráficos o mediante visión computacional, se determina el lugar por el que se va a desplazar el robot y en función de los resultados se delimitan los posibles caminos que éste puede seguir. En el Anexo 1 se presenta una clasificación detallada de los distintos métodos de generación de trayectorias existentes en robótica móvil.

También es interesante recordar algunos conceptos relacionados con el control cinemático en robótica móvil. Los métodos más extensamente utilizados son aquellos que en función de la situación actual del robot respecto a la trayectoria de referencia determinan un punto objetivo y aplican una acción sobre las ruedas que se obtiene a partir de una determinada ley de control. En el Anexo 2 se introducen las técnicas clásicas en cuanto a control de robots móviles se refiere.

## ***2.2. Generación de Trayectorias de Curvatura Continua***

Pese a que las técnicas de generación y de control introducidas en los Anexos 1 y 2 se utilizan desde hace mucho tiempo y de forma muy extendida, debido a su facilidad de implementación y su bajo coste computacional, ninguna de ellas es capaz de generar trayectorias de curvatura continua. Eso se debe al hecho de que basan su funcionamiento en la teoría de control clásico, definiendo configuraciones libres de colisión que sirven como referencia a leyes de control que cada iteración calculan una acción de control totalmente independiente de la anterior. Por lo tanto, se trata de expresiones matemáticas que sólo garantizan estabilidad y convergencia del sistema dinámico considerando la configuración del robot, pero que no consideran en ningún momento la configuración de las ruedas del robot, es decir, la posición y orientación de las mismas.

Debido a las carencias existentes en cuanto a generación de trayectorias de curvatura continua y con el objetivo de mejorar los métodos clásicos, en las últimas tres décadas algunos investigadores se han centrado en la búsqueda de soluciones capaces de integrar tanto la parte de planificación de movimiento como la de generación de trayectorias continuas en curvatura.

### **2.2.1. Trayectorias Suaves en Navegación Autónoma**

Muchos investigadores han usado las Clotoides por sus características geométricas para aplicaciones muy diferentes entre sí. En robótica móvil se han utilizado para la generación de trayectorias en problemas de navegación, evitación de obstáculos, adelantamiento y cambio de carril, aparcamiento, etc. Además, los ingenieros las utilizan para el diseño de autopistas, ferrocarriles y montañas rusas mediante las conocidas *Clothoid loops*. Otra posibilidad de uso de este tipo de curvas, que también se ha trabajado bastante en la bibliografía, es utilizar modelos clotoidales para la identificación de carreteras, utilizando diversos sistemas sensoriales entre los que destacan los sistemas de visión.

En la generación de trayectorias, el método más simple y rápido es el uso segmentos de línea y círculos, concatenados para la obtención de las trayectorias que solucionen el problema [3], [4], [5], [6]. La desventaja de esta técnica es que la curvatura resultante es discontinua, puesto que en el cambio de recta a arco de circunferencia se pasa de radio infinito a radio constante y en consecuencia la curvatura cambia su valor de forma instantánea. Para solucionarlo se introducen tramos de transición entre ambos tipos de trayectorias, pero entre las muchas propuestas que los investigadores han realizado destacan las curvas Clotoides, ya que su principal característica es el cambio de curvatura constante, de modo que permite pasar suavemente de tramos rectos a circulares y viceversa.

El primer trabajo relacionado con la generación de trayectorias continuas lo podemos encontrar en [7]. Este artículo describe la problemática del control cinemático de un robot móvil en configuración diferencial, describiendo un método para la generación de trayectorias suaves. Plantea la situación en la que el robot debe cambiar de un tramo recto a otro curvo, destacando que son los dos movimientos básicos que puede realizar cualquier vehículo. No obstante, se comenta que para realizar el cambio de trayectoria el robot debe parar y reorientar las ruedas puesto que es incapaz de hacerlo instantáneamente. Es por ese motivo por el que se introduce el

concepto de curva de transición y se plantea la posibilidad de incluirlas entre ambos tipos de trayectoria, de modo que se garantice la suavidad en toda la trayectoria sin que aparezcan discontinuidades en curvatura (ver Figura 4). No obstante en ningún momento se plantea una solución determinada, sino que se nombra el nuevo elemento como estado de transición planteando una especie de metodología para realizar el control.

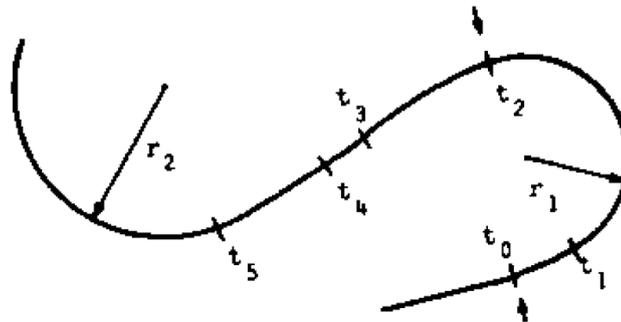


Figura 4. Ejemplo de *path* que combina tramos rectos, circulares y transiciones ( $t_0$ - $t_1$ ,  $t_2$ - $t_3$ ,  $t_4$ - $t_5$ ).

Para resolver el problema de las transiciones, en [8] se plantea la utilización de la Clotoide dadas sus propiedades geométricas. De este modo se garantiza un cambio de curvatura suave y constante de modo que se reduzcan los errores de odometría del vehículo autónomo. Se plantea toda la matemática para su inclusión como segmento de transición en el planificador de caminos.

Por lo tanto, la aportación básica de este artículo es que se propone utilizar segmentos clotoidales como método para interpolar y unir dos puntos. No obstante, el problema del uso de Clotoides es que aunque se garantiza suavidad (*smoothness*) y linealidad, no es trivial generar segmentos clotoidales para posturas arbitrarias de inicio y fin. Esto se debe a que las expresiones para esas curvas están sobre-restringidas, y por tanto, no existe una expresión de forma cerrada (se requieren métodos numéricos para evaluar la expresión).

Este método es defendible bajo la simplificación de que la curvatura inicial y final sea siempre cero. El modelo es adecuado para *paths* simples que pueden unirse mediante parejas de Clotoides simétricas o para *paths* con curvaturas leves que se pueden dividir fácilmente mediante segmentos clotoidales separados.

Otro trabajo relacionado es [9], aunque a diferencia del anterior, para resolver el problema de las transiciones entre los tramos rectos y los tramos circulares se utilizan espirales cúbicas. Según se argumenta, se trata de curvas cuya dirección tangente se describe mediante una función cúbica que depende de la distancia  $s$ , lo cual da ciertas facilidades de implementación. Este tipo de funciones se utiliza para resolver problemas de *path planning*, pero con un único *path* sólo se pueden resolver problemas de posturas simétricas y por eso para situaciones de posturas no simétricas se utilizan dos *paths* simples. Además, se introducen dos funciones de coste para obtener caminos suaves y cuantificarlos: la curvatura y su derivada, el *sharpness*.

Como resultado de los experimentos realizados se demuestra que el uso de espirales cúbicas en los tramos de transición da como resultado una trayectoria de mayor suavidad que al utilizar Clotoides (ver Figura 5).

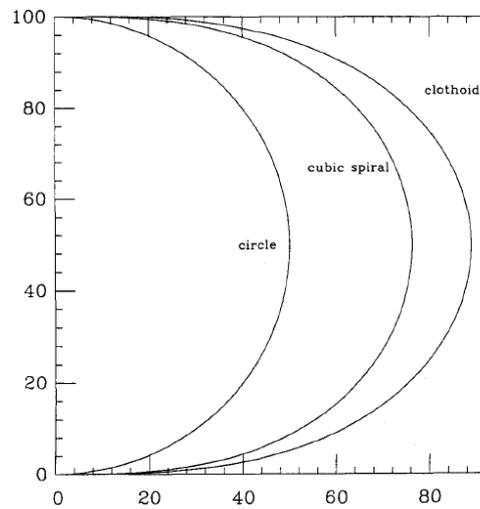


Figura 5. Curvas simples con  $\alpha = \pi$ .

En [10] se plantea el problema de la generación de trayectorias suaves desde el punto de vista del planificador. Se menciona que la respuesta del vehículo cuando realiza el seguimiento de un determinado camino depende parcialmente de la naturaleza del camino. Por ese motivo se propone un método para la generación de caminos fáciles de seguir mediante el uso de Clotoides. De este modo se consigue tener trayectorias continuas en posición, orientación y curvatura, a la vez de ser lineales en curvatura. Se introduce un método de dos pasos que es capaz primero de determinar una serie de posturas únicas a partir de una secuencia de puntos de control o *waypoints* y después se interpola entre dichas posturas uniendo cada par de puntos vecinos mediante la sucesión de tres tramos clotoidales. Además de esto, también se demuestra que cualquier par de configuraciones están inscritas en dos circunferencias que intersectan, como se observa en la Figura 6.

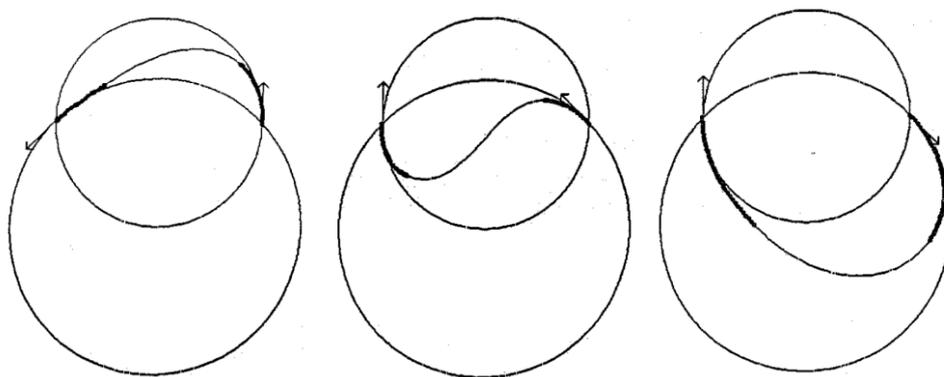


Figura 6. Distintos tipos de trayectoria de curvatura continua.

Otra idea interesante que introduce este artículo y se observa en el esquema de la Figura 7, es el hecho de que cualquier problema de navegación de un robot móvil se puede dividir en 4 sub-problemas: percepción del mundo, planificación de caminos, generación de caminos y seguimiento de caminos; los cuales se pueden resolver de forma independiente.

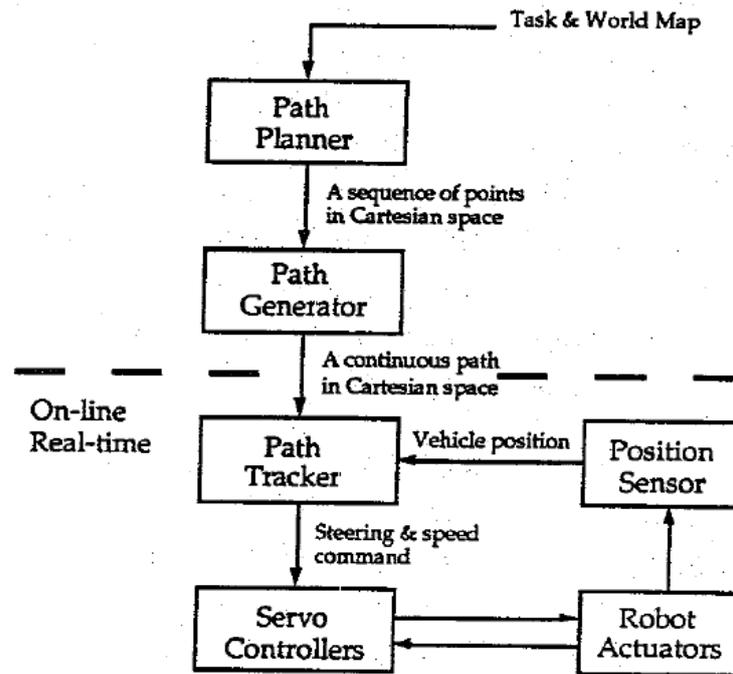


Figura 7. Estructura de control del vehículo autónomo.

En [11] se desarrolla la matemática necesaria para determinar si un tramo es de tipo Clotoide (ver Figura 8 (a)), o si por el contrario es, según el concepto que introducen, de tipo anti-Clotoide (la inversa de la Clotoide) (ver Figura 8 (b)).

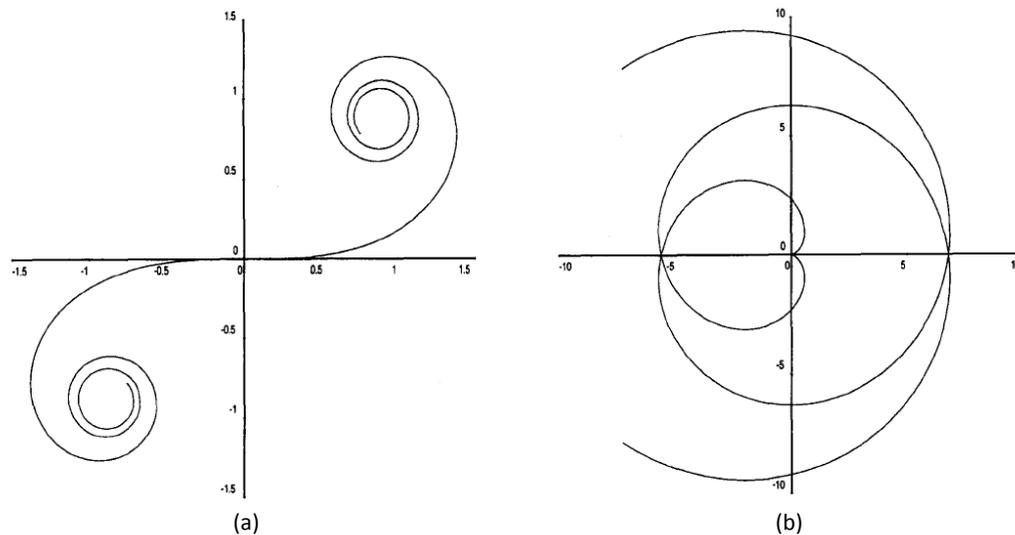


Figura 8. Ejemplo de curva: (a) Clotoide y (b) anti-Clotoide.

La anti-Clotoide se define como la trayectoria que pasa de un tramo de curvatura de un valor determinado (tramo circular) a otro tramo de curvatura nula (tramo rectilíneo). Utilizan estas curvas para vehículos en configuración diferencial. Cuando la aceleración de las dos ruedas es igual pero de signo contrario tenemos una Clotoide; en cambio, si las aceleraciones de la rueda son iguales y del mismo signo se trata de una anti-Clotoide.

Además de los conceptos anteriores, también se introducen algunas ideas que aclaran bastante ciertas situaciones que se aplican en este y en otros artículos, como por ejemplo los conceptos de *turn*, *loop* y *cuspl* (ver Figura 9 y Figura 10).

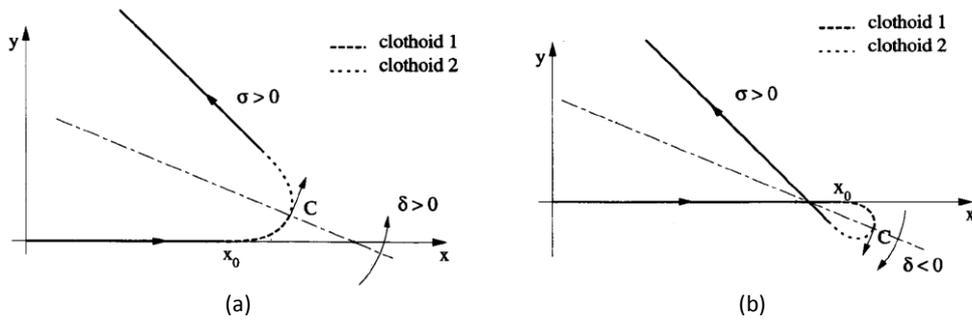


Figura 9. Ejemplos de: (a) *turn* y (b) *loop*.

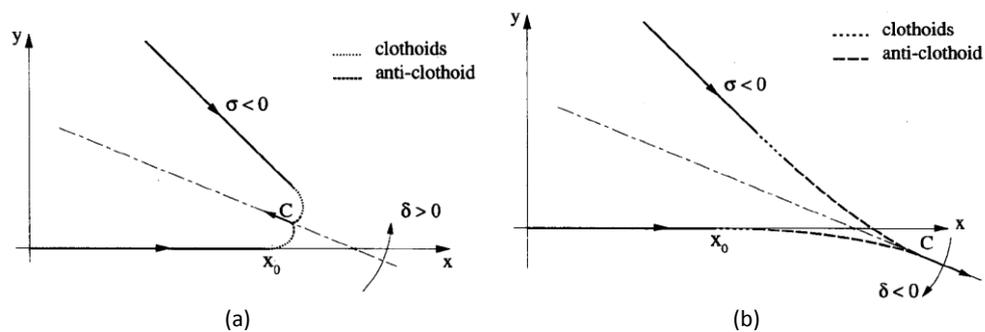


Figura 10. Ejemplos de: (a) *cusp interno* y (b) *cusp externo*.

En [12] se comparan los diferentes tipos de curvas utilizados hasta los años 90 para la generación de trayectorias en robótica móvil: líneas rectas, arcos circulares y funciones polinomiales (polinomios polares, polinomios cartesianos, polinomios Bezier, espirales de Cornu o Clotoides, espirales cúbicas, etc.). Después se propone un método de cálculo de rutas suaves que minimizan la integral del cuadrado del *jerk*. Finalmente, destaca el hecho de que se introduce una velocidad variable a lo largo de la ruta planificada, la cual es función de la curvatura, de modo que de forma directa se puedan controlar y limitar las aceleraciones centrífugas (ver Figura 11).

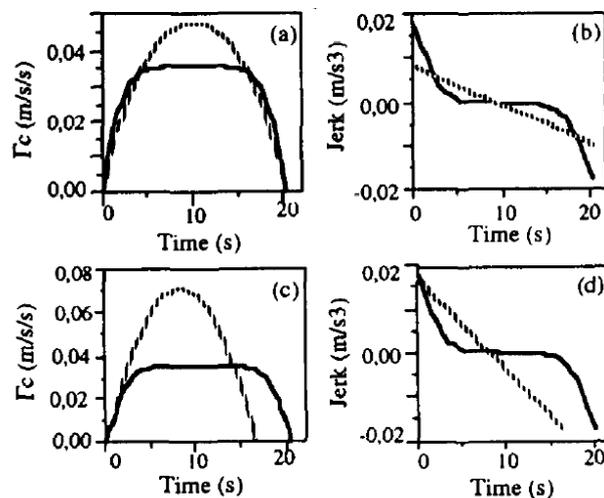


Figura 11. Perfiles de la aceleración centrípeta (izquierda) y el *jerk* (derecha) para dos trayectorias distintas, una con velocidad variable (línea continua) y otra con velocidad constante (línea discontinua).

En [13] se presenta por primera vez un planificador de caminos de curvatura continua para vehículos autónomos. Anteriormente se habían realizado trabajos de planificación que generaban el espacio de soluciones mediante tramos rectos y arcos circulares, pero ninguno había conseguido integrar un planificador que tuviera funciones de evitación de colisiones y un generador de trayectorias continuas que incorporase elementos de transición entre las rectas y los arcos (ver Figura 12). El planificador utilizado es el algoritmo de *Ariadne's Clew*, que se basa en el uso de dos funciones complementarias: búsqueda y exploración (*Search and Explore*).

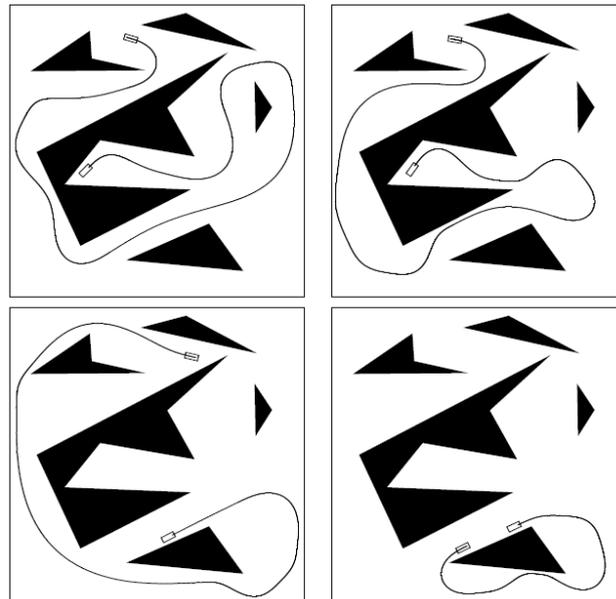


Figura 12. Ejemplos de distintas trayectorias seguidas por el vehículo en un entorno lleno de obstáculos.

Además, en este trabajo también se introduce el término *Elementary path*, para solucionar el problema de la discontinuidad en el punto de tangencia entre un segmento recto y un arco de circunferencia. Dichos elementos son una combinación de dos Clotoides iguales, es decir, con el mismo factor de homotecia. Además, para que dos posturas puedan ser enlazadas, estas deben de ser simétricas. En la Figura 13 se muestra un *Elementary path*.

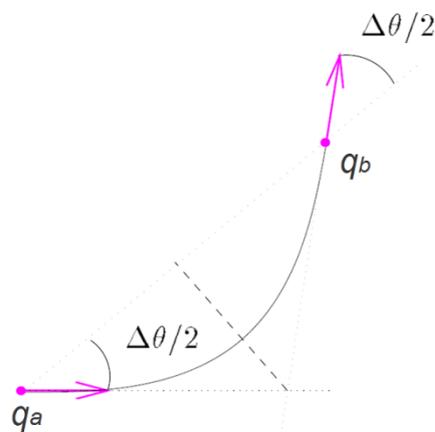


Figura 13. *Elementary path*.

Las coordenadas  $q_a$  y  $q_b$  deben verificar:

$$(x_b - x_a) \cdot \sin\left(\frac{\tau_b + \tau_a}{2}\right) = (y_b - y_a) \cdot \cos\left(\frac{\tau_b + \tau_a}{2}\right) \quad (1)$$

Siendo  $\frac{\tau_b + \tau_a}{2}$  el ángulo  $\frac{\Delta\theta}{2}$  que aparece en la Figura 13, así como  $(x_a, y_a)$  y  $(x_b, y_b)$  las coordenadas cartesianas de las configuraciones  $q_a$  y  $q_b$ , respectivamente.

Además de la ecuación anterior, para que la curva sea trazable no debe de superar el límite físico impuesto por el giro de las ruedas del vehículo, lo cual es una restricción adicional. No obstante, aunque se menciona en el artículo, no se aborda la problemática de las restricciones cinemáticas.

La aportación fundamental de [14] es la introducción del concepto de *BiElementary path*, que es la combinación de dos *Elementary paths*, tal y como aparece en la Figura 14. Del desarrollo de las características y principales propiedades de los *BiElementary paths* se demuestra que pese a que las configuraciones inicial y final no sean simétricas éstas pueden unirse mediante una configuración intermedia que garantice que es simétrica con ambas, de modo que existen infinitas candidatas para la localización de dicha configuración. No obstante, tal y como ya se ha comentado anteriormente, en [10] se demuestra que las 3 configuraciones están localizadas en una misma circunferencia, la que une la configuración inicial y final. Por lo tanto se reduce considerablemente el espacio de búsqueda de la configuración intermedia.

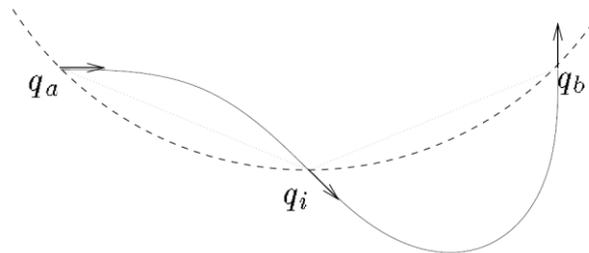


Figura 14. BiElementary Path.

Como ya se ha dicho, las poses inicial y final no son simétricas pero pueden unirse mediante la pose o configuración intermedia. La pose intermedia debe de cumplir la condición de simetría con la pose inicial y final para que sea enlazable mediante dos *Elementary paths*. La búsqueda de la configuración intermedia correcta es bastante simple, una vez dado el punto  $(x_i, y_i)$  de dicha configuración, el objetivo es encontrar la orientación que haga cumplir las ecuaciones siguientes:

$$(x_i - x_a) \cdot \sin\left(\frac{\tau_i + \tau_a}{2}\right) = (y_i - y_a) \cdot \cos\left(\frac{\tau_i + \tau_a}{2}\right) \quad (2)$$

$$(x_b - x_i) \cdot \sin\left(\frac{\tau_b + \tau_i}{2}\right) = (y_b - y_i) \cdot \cos\left(\frac{\tau_b + \tau_i}{2}\right) \quad (3)$$

Además, se introducen dos nuevas limitaciones: el giro y la curvatura máxima que el robot podría realizar. Aunque de momento sólo se menciona el problema, ya que aquí tampoco se aborda la problemática desde un punto de vista de aplicación real de las restricciones sobre la planificación del movimiento del robot.

Los autores de [15] se inspiran en las *Dubins' curves* [3] para simplificar el problema de búsqueda del camino óptimo mediante la creación de los *CC-turns* (*continuous-curvature turns*) (ver Figura 15) y los *SCC-paths* (*simple continuous-curvature paths*) compuestos por dos *CC-turns* unidos por un segmento de recta (Figura 16). Se define cada elemento fundamental de unión entre dos configuraciones consecutivas como la combinación de como máximo 7 piezas entre Clotoides, arcos circulares y segmentos de línea.

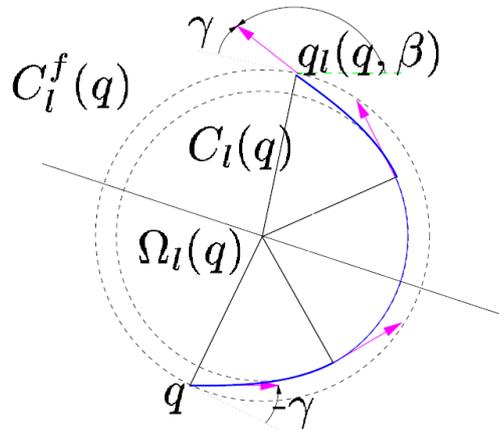


Figura 15. *CC-turn*.

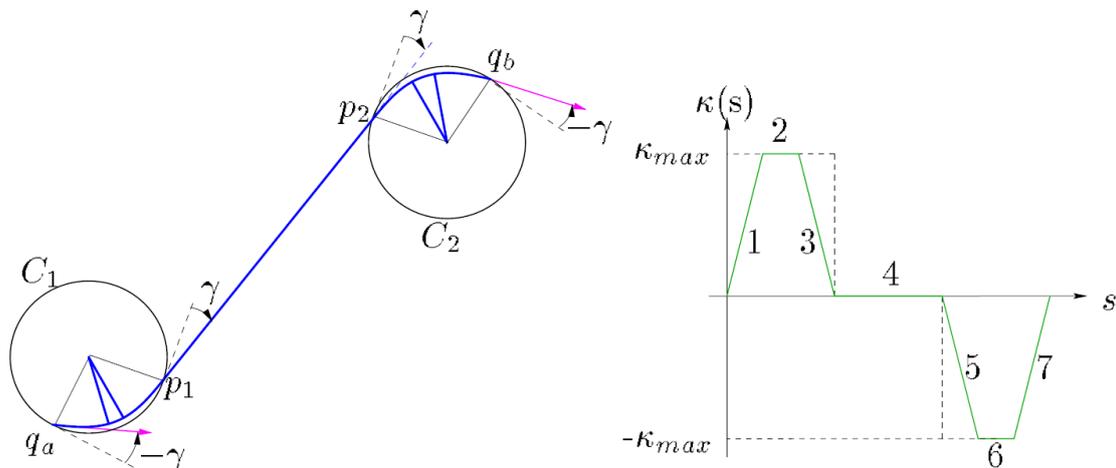


Figura 16. *SCC-path* (izquierda) y su perfil de curvatura (derecha).

Mediante el método presentado en este trabajo de investigación se consigue un planificador capaz de generar soluciones a partir de elementos más simples, cuya curvatura es continua y lineal con respecto a la longitud del arco. Se presenta un planificador local que utiliza los *SCC-paths* para generar una primera aproximación a la solución del problema y posteriormente se combina con un método de planificación de caminos probabilístico (*Probabilistic Path Planning*) para obtener un único planificador global del vehículo. Además de poder trabajar con un único vehículo se introducen las bases para trabajar con planificadores de caminos para múltiples vehículos. En la Figura 17 se observa un ejemplo sencillo de planificación de una ruta, en el que se comparan los resultados obtenidos con *Dubins' curves* y con *SCC-paths*.

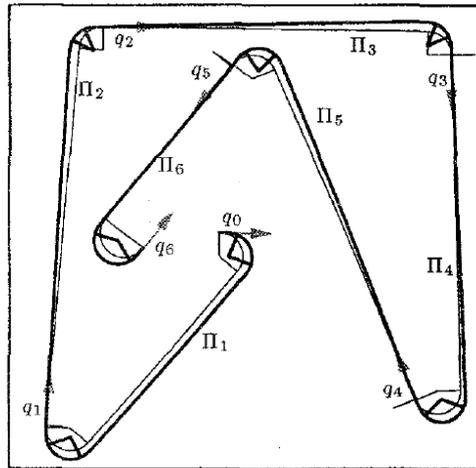


Figura 17. Trayectoria generada con *Dubins' curves* (línea fina) y con *SCC-paths* (línea gruesa).

En [16] también se describe de forma genérica un planificador global de trayectorias con curvatura continua para vehículos. Aunque la novedad es que se combinan los sistemas existentes de evitación de colisiones basados en líneas y círculos, introduciendo Clotoides y otras mejoras para crear un planificador completo (ver Figura 18).

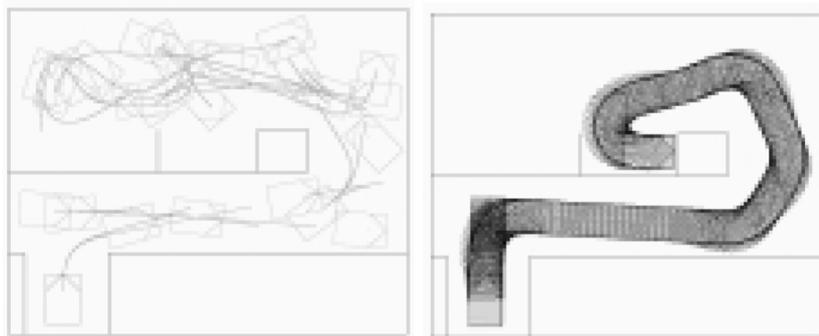


Figura 18. Ejemplo de trayectoria generada por el planificador propuesto en este artículo: *Landmarks* (izquierda) y traza del vehículo (derecha).

El concepto de postura se introduce en [17], que a diferencia de la pose, que sólo incluye posición y orientación del vehículo, también incluye la curvatura del mismo. Además se propone un método para el control del vehículo capaz de llegar a cualquier posición y seguir cualquier trayectoria de forma precisa.

No obstante, la idea fundamental del artículo es el hecho de utilizar polinomios de orden cúbico para la curvatura, los cuales, según explican, son ideales para usarse como primitivas en la generación de trayectorias y el control de vehículos móviles. A diferencia de las Clotoides, que son polinomios lineales en curvatura y están muy restringidas, las curvas cúbicas se pueden usar para determinar una trayectoria única desde cualquier postura arbitraria mediante una única primitiva. Otra propiedad de este tipo de curvas es que son las de menor orden que además garantizan un par o *torque* continuo aplicado a mecanismos de dirección, por eso generan trayectorias fáciles de seguir por un vehículo real. Igual que las Clotoides los polinomios cúbicos son difíciles de computar analíticamente, pero son fáciles de ejecutar. Finalmente, se describe un método numérico de tiempo real para el cómputo de este tipo de curvas.

Como ya se sabe la Clotoide tiene la propiedad de que su curvatura varía linealmente frente a la longitud del arco y también que ésta es una característica muy útil para el control de un vehículo, de forma que el radio de giro sea una función lineal con respecto a la distancia recorrida. Tanto en el diseño de carreteras o raíles como en la generación de rutas en robótica móvil se combinan tramos rectos, tramos circulares y tramos de transición para generar las trayectorias. La aportación fundamental de [18] es la extrapolación al mundo de la robótica móvil del uso de puntos de control de poli-líneas, característicos de aplicaciones de diseño CAD (*Computer Aided Design*). Se examina por tanto la utilidad del uso de poli-líneas de control para realizar el diseño de rutas formadas por rectas, arcos y Clotoides.

También [19] está relacionado con la generación de trayectorias suaves para el control basado en la inversión de la cinemática de robots móviles de ruedas. La principal aportación es la propuesta de una nueva primitiva para la generación de caminos, llamada  $\eta^3$ -*spline*. Se trata de un polinomio de séptimo orden de tipo *spline*, el cual permite interpolar entre una secuencia arbitraria de puntos con direcciones tangentes, curvatura y derivada de la curvatura también arbitrarias. Un *path* con continuidad geométrica de tercer orden tiene un vector tangente, una curvatura y una derivada de la curvatura continua a lo largo del arco recorrido. Adoptando este esquema de planificación y una técnica de planificación por inversión de la dinámica se garantizan velocidades y aceleraciones continuas. Con la nueva primitiva propuesta se pueden unificar otros tipos de curvas primitivas como por ejemplo arcos circulares, Clotoides, espirales, etc. Finalmente el artículo incluye resultados teóricos y algunos ejemplos de planificación de caminos.

En [20] utilizan el método de planificación de caminos *v-shape* y con pares de Clotoides simétricas consiguen resolver el problema de navegación de un vehículo articulado por una cantera. Utilizan líneas para unir la postura inicial con la final y calculan cuándo deben de empezar la Clotoide para cumplir la simetría necesaria definida en [13]. También especifican el método para calcular los parámetros de la Clotoide condicionados a la limitación de giro que tiene el vehículo articulado.

En [21] se continúa el trabajo de investigación realizado en [20]. Se presenta un método de generación de caminos utilizando, para la verificación del algoritmo, un vehículo autónomo de ruedas que trabaja como cargador en minas o lugares de construcción. Se utiliza el método *v-shape* para conectar la posición de carga y la de descarga, mediante la combinación de líneas rectas y tramos clotoidales (ver Figura 19). Hay que destacar que cada segmento se optimiza en longitud en el procedimiento de generación de la trayectoria.

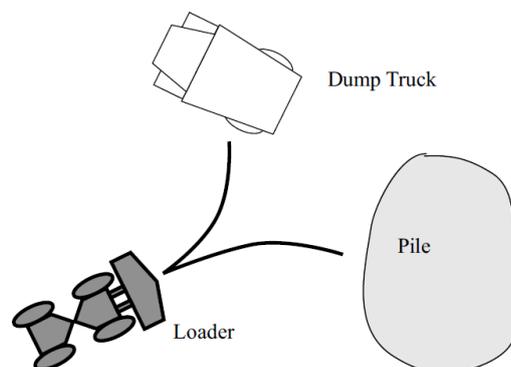
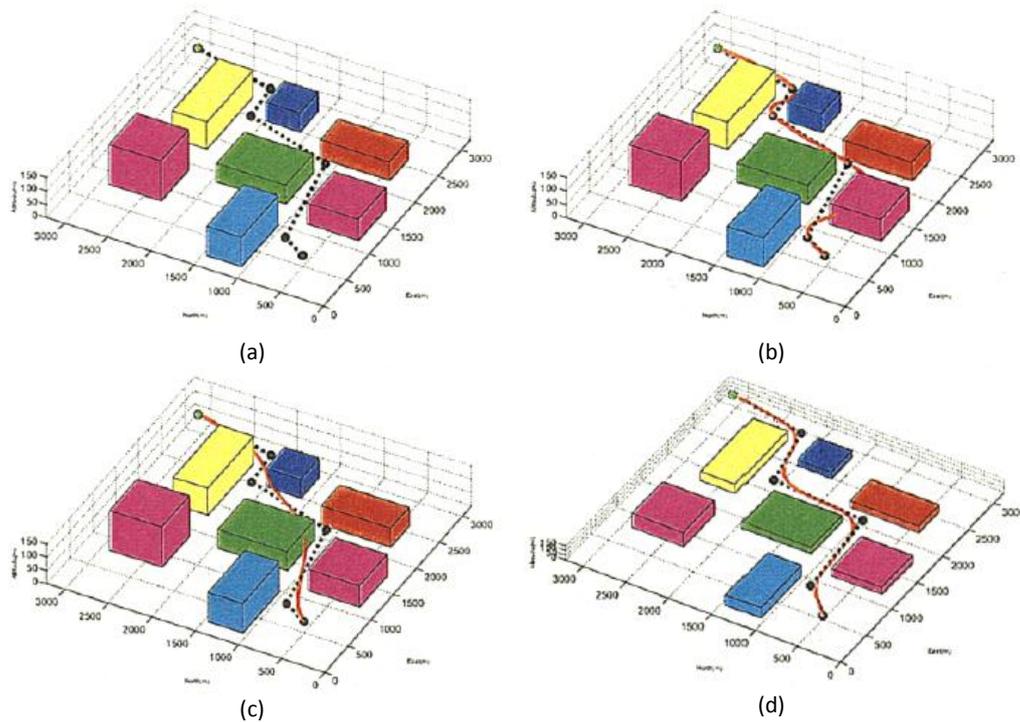


Figura 19. Ruta *v-shape*.

Además de la generación de trayectoria la aplicación desarrollada también utiliza un sistema de estéreo-visión para determinar la forma de la pila y determinar la posición y dirección óptimas de la pala para que ésta tenga el menor momento de torsión, complementándose con una estimación de la fuerza de resistencia realizada durante el movimiento de carga.

Los autores de [22] presentan un método para la generación en tiempo real de trayectorias de curvatura continua para UAV (*Unmanned Aerial Vehicles*) desplazándose en un entorno con obstáculos. Para ello utilizan primero un planificador para generar una ruta libre de colisiones y después unen los puntos de control obtenidos mediante espirales cúbicas de Bezier (Figura 20), las cuales demuestran que mejoran significativamente los resultados con respecto a los obtenidos con las curvas de Dubins y con las curvas continuas cúbicas de Bezier.



**Figura 20. Comparación de los métodos de generación de rutas suaves: (a) es el conjunto de puntos de referencia para obtener un camino suave, (b) es el camino suave obtenido mediante interpolación de una spline cúbica, (c) es el camino obtenido mediante una *B-spline* cúbica y (d) es el resultado obtenido mediante el uso de una Bezier cúbica.**

Al igual que el anterior artículo, [23] se centra en la planificación de trayectorias para un vehículo autónomo volador. El vehículo trabaja en un ambiente con un flujo de aire suave y uniforme y lo modelan como una partícula que viaja en un plano horizontal con una velocidad constante relativa al flujo de campo del ambiente. El vehículo puede girar en cualquier dirección, sujeto a una restricción de cambio de velocidad y aceleración. La principal contribución del artículo es la presentación de un método simple para generar caminos candidatos a ser los de menor tiempo para unos puntos inicial y final dados, tal y como se observa en la Figura 21.

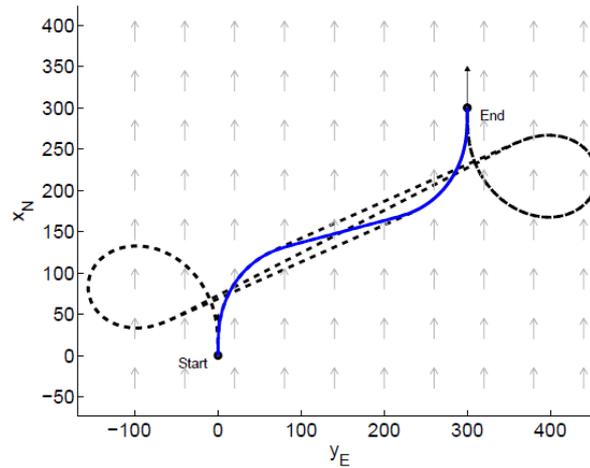


Figura 21. Los 4 caminos posibles. La candidata a ser la trayectoria de mínimo tiempo está coloreada de azul.

### 2.2.2. Trayectorias Suaves con Restricciones Cinemáticas y Dinámicas

La variación temporal de la aceleración se llama *jerk* y es importante en ciertas aplicaciones de mecánica y acústica. Concretamente en [24] se desarrollan las ecuaciones que definen el *jerk* para el movimiento sobre un plano. Éste se descompone en componentes tangencial y normal o radial y transversal. Además, se trabaja con un término llamado *aberrancy* que es invariante diferencial y afín, lo cual permite expresar la componente normal de forma que se reflejen mejor algunas propiedades del vector *jerk* en el movimiento sobre un plano.

En un plano se pueden definir los vectores de velocidad  $\mathbf{v}(t)$  y aceleración  $\mathbf{a}(t)$  como:

$$\left. \begin{aligned} \mathbf{v}(t) &= \dot{\mathbf{R}} = v \cdot \mathbf{t} \\ \mathbf{a}(t) &= \dot{\mathbf{v}} = \dot{v} \cdot \mathbf{t} + \frac{v^2}{\rho} \mathbf{n} \end{aligned} \right\} \quad (4)$$

Si se diferencia (4) utilizando  $\dot{\mathbf{t}} = (v/\rho) \cdot \mathbf{n}$  y  $\dot{\mathbf{n}} = -(v/\rho) \cdot \mathbf{t}$  se obtiene la variación temporal del vector aceleración, que se llama vector *jerk*:

$$\mathbf{j}(t) = \left( \ddot{v} - \frac{v^3}{\rho^2} \right) \cdot \mathbf{t} + \left( \frac{v}{\rho} \right) \left[ 3\dot{v} - \left( \frac{v}{\rho} \right) \dot{\rho} \right] \cdot \mathbf{n} \quad (5)$$

Si se compacta más (5) se obtiene:

$$\mathbf{j}(t) = \left( \ddot{v} - \frac{v^3}{\rho^2} \right) \cdot \mathbf{t} + \frac{1}{v} \left( \frac{d}{dt} \frac{v^3}{\rho} \right) \cdot \mathbf{n} \quad (6)$$

En [25] se desarrollan por primera vez rutas en un plano para unir dos configuraciones dadas (posición, orientación y curvatura)  $q_0$  y  $q_f$ , de forma que a lo largo de la trayectoria generada la derivada de la curvatura (con respecto a la longitud del arco) permanece acotada.

Según se argumenta en [26], la mayoría de estándares recomiendan el uso de Clotoides en el diseño de tramos de transición en carreteras. Concretamente, el estándar italiano siempre requiere el uso de Clotoides entre dos elementos de curvatura constante distinta (curvas tangentes o circulares), independientemente del radio de la curva o de la categoría de la

carretera. Esto se justifica por el hecho de que el uso de Clotoides garantiza un incremento gradual de la fuerza centrífuga, además de ser muy apropiado para arreglar los pasos elevados y dar una apariencia muy satisfactoria del acabado de la carretera.

El artículo se centra en demostrar cuales son los criterios óptimos para establecer un límite para el parámetro de las Clotoides utilizadas en los diseños de las carreteras. Se explica también que según el estándar vigente existe un rango demasiado elevado para la selección de la longitud de los tramos clotoidales, lo cual hace que en muchos casos el diseño no sea óptimo. En particular se expone que, a diferencia de otros estándares extranjeros, se permite utilizar curvas espirales excesivamente largas sin ninguna advertencia, pese a las posibles consecuencias negativas de su uso. De hecho, una Clotoide de gran longitud puede tener efectos potencialmente negativos sobre la percepción de la curva por parte del conductor y en consecuencia sobre la seguridad de la conducción. Con la propuesta realizada se busca establecer unos límites máximo y mínimo para la longitud de los tramos clotoidales, dependiendo del radio de curvatura que se calcula, considerando todos los aspectos relativos al comportamiento del conductor. Se garantiza un diseño óptimo si la selección de la longitud de las Clotoides está dentro de esos rangos, de modo que se eviten problemas de percepción de la curvatura y, en consecuencia, mejorando la seguridad de las carreteras.

Para establecer el criterio de confort se presenta una relación matemática que incluye la variación de la aceleración lateral como parámetro de limitación. Asumiendo que el camino de guía es una Clotoide y considerando el cambio a lo largo de la rampa  $q$ , el ratio de variación de la aceleración lateral  $c$  se define como

$$c = \frac{d\left(\frac{v^2}{r} - g \cdot q\right)}{dt} = \frac{d\left(\frac{v^2}{r}\right)}{dt} - \frac{d(g \cdot q)}{dt} \quad (7)$$

Dónde:

- $v$  es la velocidad (m/s).
- $r$  es el radio de curvatura (m).
- $g$  es la constante de gravedad (m/s<sup>2</sup>).
- $q$  es el ratio de superelevación.

Según el estándar italiano una vez calculado el parámetro  $c$  se puede comprobar en una tabla (ver Tabla 1) la correspondiente relación entre la velocidad máxima de trazado de la curva y los parámetros de diseño correspondientes a la Clotoide.

V <sub>p</sub>	40	50	60	70	80	90	100	110	120	130	140
R	45	76	118	178	252	339	437	544	667	806	964
L	27.2	26.6	30.5	36.0	39.7	42.5	48.1	56.3	66.1	74.5	87.2
A	35	45	60	80	100	120	145	175	210	245	290
$c_{max}$	1.54	1.36	1.21	1.06	0.94	0.84	0.75	0.68	0.60	0.54	0.48

**Tabla 1. Longitud  $L$  de la Clotoide correspondiente a  $c_{max}$  en  $s = 0$ .**

Otro criterio es la limitación del parámetro máximo de la Clotoide según la apariencia de la carretera. Para ello se debe consultar otra tabla (ver Tabla 2).

V <sub>p</sub>	40	50	60	70	80	90	100	110	120	130	140
R	45	76	118	178	252	339	437	544	667	806	964
A <sub>max</sub>	40	55	80	105	140	175	210	250	290	335	380

Tabla 2. Parámetro de la Clotoide correspondiente a un *offset* tangente de 1m.

El tercer criterio de selección relaciona el gradiente entre el radio y la longitud de la Clotoide, determinando una gráfica (ver Figura 22) y el método de obtención de los límites según las velocidades.

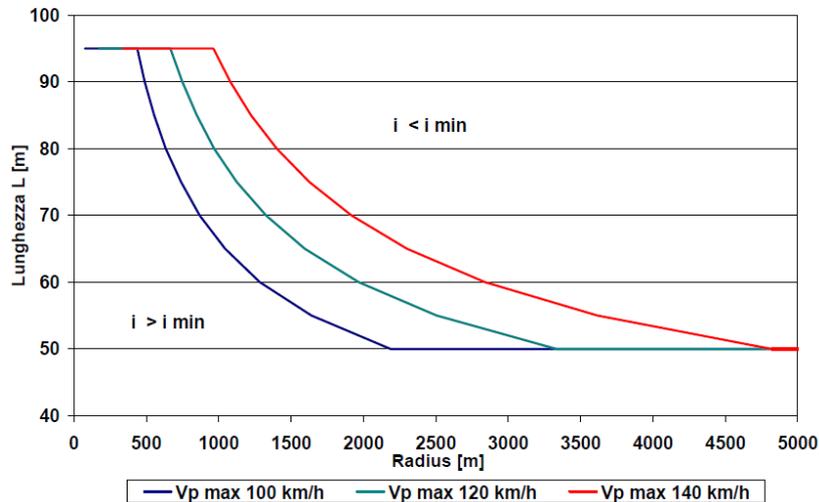


Figura 22. Longitud *L* correspondiente al gradiente mínimo relativo frente al radio de curvatura *R*.

En definitiva, lo que este trabajo de investigación propone es simplificar el criterio de diseño mediante la unificación de la selección de los parámetros de las Clotoides en un único método. Para ello se presenta la gráfica de la Figura 23, que relaciona directamente el parámetro *A* de la Clotoide con el radio de curvatura *R*.

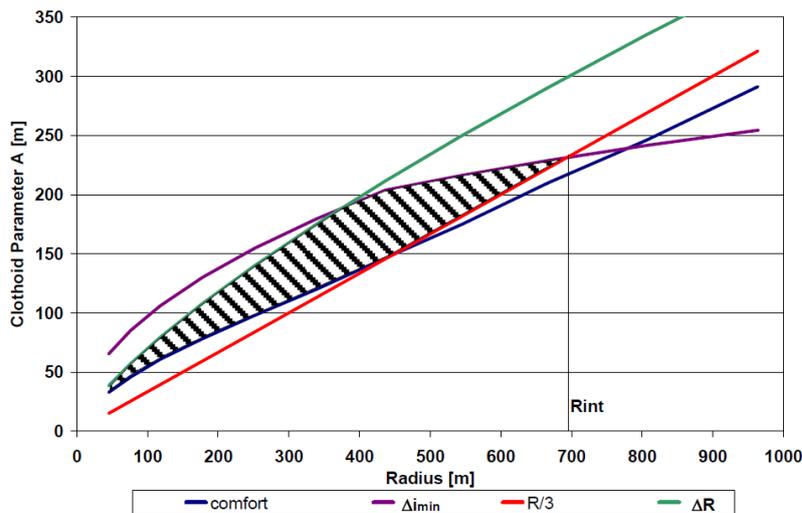


Figura 23. Gráfica de relación entre el parámetro de la Clotoide *A* y el radio de curvatura *R*.

Finalmente se llega a una tabla (ver Tabla 3) que permite obtener rápidamente los límites de diseño en función del radio deseado.

R	45	76	118	178	252	339	437	544	667	806	964
$A_{\min}$	35	45	60	80	100	120	145	180	220	270	320
$A_{\max}$	40	55	80	105	140	175	205	215	230	270	320

**Tabla 3. Tabla de relación entre el parámetro de la Clotoide  $A$  y el radio de curvatura  $R$ .**

En [27] se remarca el creciente interés que tienen los vehículos eléctricos autónomos para el futuro transporte público en ciudades, ya que pueden ofrecer la misma flexibilidad pero con muchas más ventajas. Para la planificación de operaciones en este tipo de vehículos se deben de generar trayectorias suaves, con aceleraciones y *jerks* bajos, para garantizar la confortabilidad de los pasajeros.

El artículo se centra en introducir una etapa de planificación de velocidad para generar una secuencia temporal adecuada para el uso en la interpolación de curvas del planificador de movimientos. Además, el perfil de velocidad generado se puede separar en una trayectoria (con componente temporal) para su uso en el seguimiento de trayectorias o se puede usar independientemente (como una curva 2D sin componente temporal) para el seguimiento de caminos.

Una aportación muy interesante es el hecho de que permite cuantificar las aceleraciones y *jerks* máximos, ya que analiza los efectos de las aceleraciones sobre el cuerpo humano a partir de sus definiciones matemáticas [24] y la normativa ISO correspondiente. Según se explica, cuando un vehículo sigue una trayectoria con cierta velocidad  $v$  se inducen unas aceleraciones sobre los pasajeros, que pueden ser expresadas como:

$$\mathbf{a} = \frac{dv}{dt} \mathbf{e}_T + v \frac{d\theta}{dt} \mathbf{e}_N \quad (8)$$

Donde  $v$  denota la velocidad longitudinal (tangente a la trayectoria),  $\theta$  es la orientación del vehículo, y  $\mathbf{e}_T$  y  $\mathbf{e}_N$  son los vectores unitarios en las direcciones tangente y normal a la trayectoria respectivamente (equivalentes a los vectores  $\mathbf{t}$  y  $\mathbf{n}$  de las ecuaciones (4-6)).

Además:

$$\frac{d\theta}{dt} = \frac{1}{\rho} v \quad (9)$$

Donde  $\rho$  es el radio de curvatura.

De (8) y (9) se obtiene la aceleración longitudinal (componente tangencial) se ve afectada por las variaciones en velocidad:

$$a_T = \frac{dv}{dt} \quad (10)$$

También se deduce que la aceleración lateral (componente normal) se origina por cambios en la orientación del vehículo, cuyos valores a su vez se ven afectados por la velocidad del vehículo:

$$a_N = \frac{d\theta}{dt} v = \frac{1}{\rho} v^2 \quad (11)$$

La aceleración lateral es función de la curvatura y de la velocidad de la trayectoria. Asumiendo velocidad constante, cuanto menor es la curvatura menor es la aceleración lateral inducida y por tanto se producirán menos efectos perjudiciales sobre los pasajeros. El estándar ISO 2631-1 (Tabla 4) muestra la relación entre el confort del cuerpo humano y la aceleración eficaz (r.m.s.) máxima, definida como:

$$a_w = \sqrt{k_x^2 a_{wx}^2 + k_y^2 a_{wy}^2 + k_z^2 a_{wz}^2} \quad (12)$$

Donde  $a_{wx}$ ,  $a_{wy}$ ,  $a_{wz}$  son las componentes eficaces de la aceleración en los ejes  $x$ ,  $y$ ,  $z$ ; mientras que  $k_x$ ,  $k_y$ ,  $k_z$  son factores de multiplicidad.

Por ejemplo, para una persona sentada  $k_x = k_y = 1.4$  y  $k_z = 1$ , mientras que para el movimiento en el plano  $xy$ , se tiene  $a_{wz} = 0$ . Además el sistema de coordenadas local se pone de tal forma que el eje  $x$  se alinee con el eje longitudinal del vehículo y el eje  $y$  defina la dirección lateral de la trayectoria.

Overall acceleration	Consequence
$a_w < 0.315 \text{ m/s}^2$	Not uncomfortable
$0.315 < a_w < 0.63 \text{ m/s}^2$	A little uncomfortable
$0.5 < a_w < 1 \text{ m/s}^2$	Fairy uncomfortable
$0.8 < a_w < 1.6 \text{ m/s}^2$	Uncomfortable
$1.25 < a_w < 2.5 \text{ m/s}^2$	Very uncomfortable
$a_w > 2.5 \text{ m/s}^2$	Extremely uncomfortable

Tabla 4. Límite de aceleraciones según el estándar ISO 2631-1.

Como es lógico, cualquier planificador de trayectorias para vehículos de transporte de pasajeros debe generar trayectorias suaves con bajas aceleraciones y *jerks* asociados. Por lo tanto, el planificador de trayectorias no solo debe generar curvas suaves (dimensión espacial) sino que también su perfil de velocidad asociado (dimensión temporal). Finalmente se analizan tres métodos de planificación de trayectorias desde el punto de vista del confort de los pasajeros, la facilidad de implementación y el seguimiento de trayectoria.

Como se puede ver en la Figura 24 los autores de este artículo son capaces de resolver problemas bastante complejos dentro de un entorno urbano.

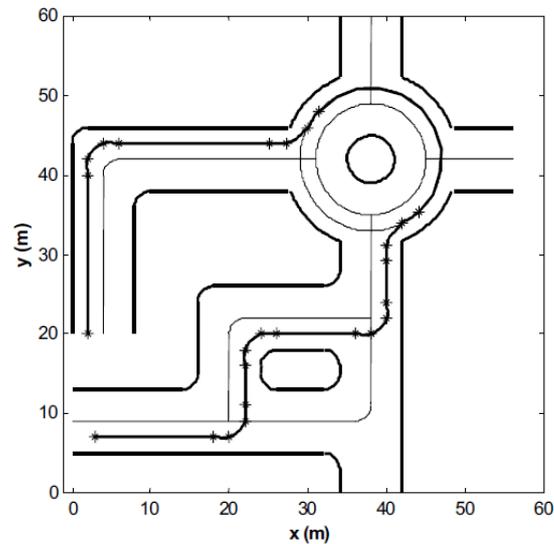


Figura 24. Ejemplo de problema complejo de planificación resuelto por el método propuesto.

En [28] se mejora el planificador que incluye los denominados *SCC-paths* [16], ya que además añaden las restricciones de curvatura continua y derivada de la curvatura, de modo que estén acotadas y no superen cierto valor límite. Se analiza la complejidad computacional de ambos métodos, el *RS-path* y el *SCC-path*, haciendo un análisis temporal. También se compara la longitud de los caminos generados por ambos métodos, así como la calidad del seguimiento de las rutas generadas por parte de un vehículo. Se demuestra que el uso de los denominados *SCC-paths* mejora las soluciones obtenidas mediante las *Dubins' curves*, ya que éstas sólo usan tramos rectos y arcos circulares, lo cual, como ya se ha comentado anteriormente genera una discontinuidad de la curvatura de la trayectoria.

En [29] utilizan un robot no-holonómico para diseñar un planificador genérico mediante la combinación de Clotoides y anti-Clotoides. Al tener restricciones holonómicas se incluye por primera vez en un planificador junto al generador de trayectorias continuas el concepto de tiempo. Se consideran por lo tanto los perfiles temporales no solo de posición sino también de velocidad y aceleración. Se demuestra que la solución sub-óptima obtenida en un planificador de trayectorias continuas para vehículos con limitación de su ángulo de giro es óptima para robots totalmente maniobrables, es decir, aquellos que no tienen dicha limitación cinemática.

La aportación más significativa para la comunidad robótica sobre generación de trayectorias continuas con restricciones cinemáticas y dinámicas es la de [30]. En trabajos de investigación anteriores a este [13], [14], [15], [16], los autores derivaron los *RS-paths* (*Reeds and Shepp's paths*) introducidos en [6], que son el camino más corto entre dos configuraciones dadas generado mediante segmentos de recta y arcos circulares (de radio mínimo). A partir de ellos crearon los *SCC-paths* (*Simple Continuous-Curvature paths*), que son similares pero que además aseguran la continuidad, ya que sustituyen los arcos circulares por lo que denominan *CC-turns*.

En la Figura 25 se muestra un ejemplo de perfil de curvatura de los *RS-paths* y de los *SCC-paths*.

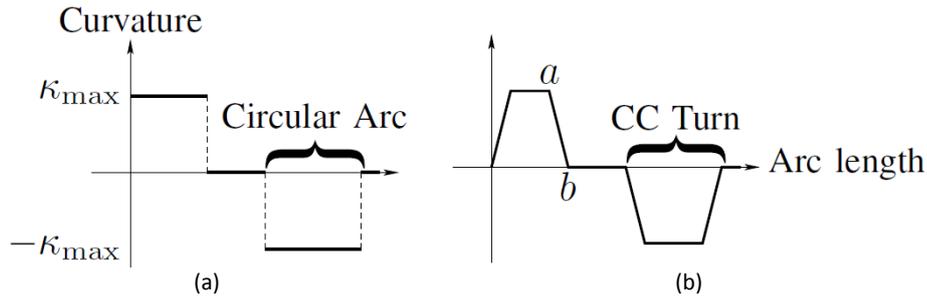


Figura 25. (a) Perfil de curvatura discontinuo de un *RS-path*. (b) Perfil de curvatura continuo de un *CC-path*. El tramo entre  $a$  y  $b$  es un arco de Clotoide.

La principal aportación de este artículo es el hecho de obtener trayectorias limitadas tanto cinemática como dinámicamente. En primer lugar establece el modelo del vehículo, el cual, al tener una configuración como un coche y no como un robot, tiene limitaciones cinemáticas en el ángulo de giro de las ruedas orientables.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\kappa} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \kappa \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \sigma \quad (13)$$

Este modelo extiende el introducido en [3], ya que incorpora un cuarto estado del sistema, la curvatura. Tal y como se observa, este modelo dinámico tiene dos entradas de control:

- $v$  es la velocidad de conducción de las ruedas de tracción (traseras).
- $\sigma$  es la aceleración angular que está relacionada con  $\dot{\phi}$ , la velocidad de giro de las ruedas orientables (delanteras). Es la derivada de la curvatura  $\kappa$ .

En [25] el objetivo es obtener la planificación del camino más corto sujeto a restricciones. Se asume una velocidad de conducción unitaria con un límite superior de la aceleración angular (suponiendo que la velocidad actual de la dirección del vehículo está limitada):

$$v = 1 \rightarrow |\sigma| \leq \sigma_{max} \quad (14)$$

Bajo esas suposiciones, el trabajo realizado en [30] determina que la planificación de la trayectoria entre dos configuraciones es equivalente a obtener una curva plana de curvatura continua que permita unir dos puntos en el plano con una orientación y una curvatura determinadas. Como ya se ha comentado, los *CC-paths* combinan *CC-turns* y tramos rectos para planificar rutas de curvatura continua. Pero existen problemas a la hora de implementarse puesto que en dichas soluciones no se tiene en cuenta la controlabilidad del sistema en periodos de tiempo pequeños, es decir, que para ir de una configuración a otra pueden requerirse acciones de control elevadas en periodos de tiempo cortos, lo cual puede no ser realizable físicamente por el vehículo debido a sus limitaciones. Para evitar ese problema se presenta un nuevo tipo de *CC-path* en el que existe limitación de *sharpness*  $|\sigma| \leq \sigma_{max}$  y limitación de la curvatura máxima o en consecuencia del radio mínimo  $R_{min} = \kappa_{max}^{-1}$ .

El caso general de un *CC-turn* permite dividir su geometría en tres partes:

- Un arco de Clotoide con *sharpness*  $\sigma = \pm\sigma_{max}$ , con curvatura desde 0 hasta  $\pm\kappa_{max}$ .
- Un tramo circular con radio  $\pm\kappa_{max}^{-1}$ .
- Un arco de Clotoide con *sharpness*  $\sigma = \mp\sigma_{max}$ , con curvatura desde  $\mp\kappa_{max}$  hasta 0.

Por lo tanto, como conclusión hay que destacar que la principal aportación de este artículo es el hecho de que basándose en los conocidos *CC-paths* es capaz de resolver problemas de planificación con restricciones cinemáticas (limitaciones en los ángulos de giro del vehículo controlado, que se representa por  $\delta_{min}$ ) y dinámicas (limitaciones en el *sharpness* o brusquedad y en consecuencia en el *jerk* sufrido por el vehículo en su movimiento, que se representa por  $\sigma_{max}$ ).

### 2.2.3. Trayectorias Suaves en Aplicaciones de Conducción

Además de los estudios y desarrollos teóricos realizados en relación a la generación de trayectorias de curvatura continua, los cuales sirven como base matemática para su uso en distintas aplicaciones de planificación de movimientos en robótica móvil, también se han realizado trabajos orientados a las aplicaciones concretas de cambio de carril, adelantamiento de vehículos y maniobras de aparcamiento. Este tipo de problemas se han intentado resolver de diversas formas, siendo la resolución geométrica el caso más sencillo [31], [32], [33]. No obstante, la inclusión en el planificador de movimientos de trayectorias de curvatura continua hace que las soluciones sean óptimas para resolver este tipo de situaciones, puesto que se corresponde con el control manual que los humanos aplican sobre un vehículo cuando realizan dichas maniobras. A continuación se comentarán algunos trabajos de investigación cuyo objetivo es incluir trayectorias suaves para resolver los problemas de conducción planteados.

[34] se centra en el problema de planificación de movimientos de robots con configuración coche (robots móviles con restricciones no holonómicas, cuyo radio de giro está limitado). Se presenta un planificador exacto y rápido para el modelo de robot que proponen. Dicho planificador se basa en un algoritmo recursivo de subdivisión que genera un camino libre de colisión mediante un planificador geométrico de más bajo nivel que ignora las restricciones de movimiento del vehículo. Tras la primera fase se optimiza la trayectoria dada para obtener una ruta de longitud mínima que además cumpla las restricciones cinemáticas del vehículo. Todos los desarrollos teóricos se complementan con implementaciones que demuestran la alta velocidad del algoritmo propuesto. Entre los ejemplos de demostración destaca el problema del aparcamiento, en el que se ven claramente diferenciadas las tres fases (ver Figura 26).

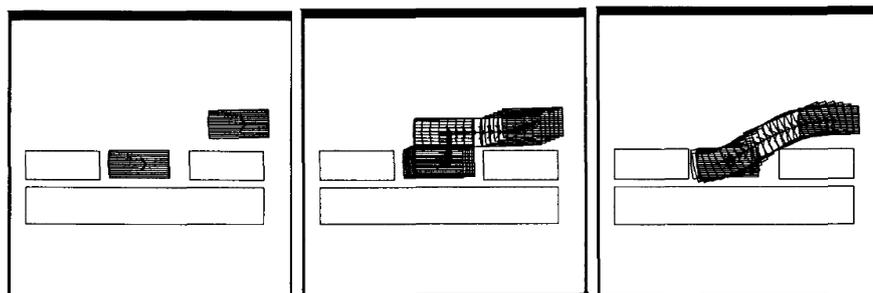


Figura 26. Posiciones inicial y final (izquierda). Solución del planificador sin restricciones holonómicas (centro). Solución final optimizada con restricciones holonómicas (derecha).

Los autores de [35] presentan un algoritmo para resolver el problema de cambio de carril. El objetivo es que se pueda aplicar sobre sistemas de vehículos inteligentes en carreteras. El algoritmo se basa en el uso de polinomios para la planificación de trayectorias y el uso de s-topos simplificados para la representación de los obstáculos. Las restricciones dinámicas también se tienen en cuenta. El criterio resultante de selección de la trayectoria deseada y factible, involucra la búsqueda de un único coeficiente. Las Figuras 54, 55 y 56 muestran diversos ejemplos para ilustrar las principales ventajas del algoritmo, especialmente en el caso del cambio de carril con múltiples obstáculos.

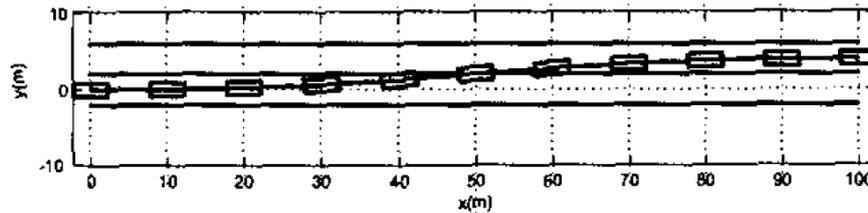


Figura 27. Trayectoria calculada sin obstáculos.

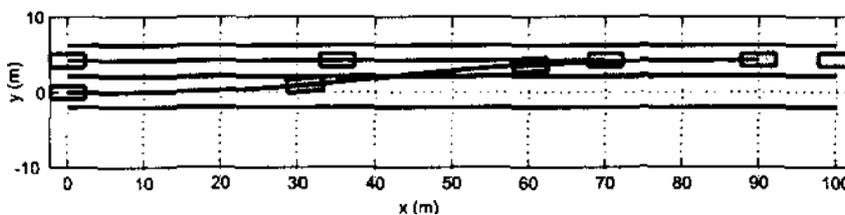


Figura 28. Trayectoria libre de colisiones.

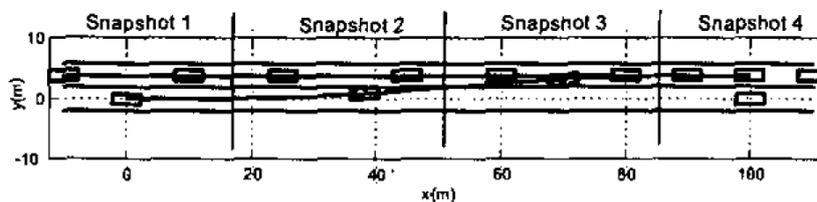


Figura 29. Instantáneas de la trayectoria libre de colisiones.

En cambio [36] aporta una ligera variación con respecto al anterior, puesto que la aproximación de las Clotoides la realiza a través de las conocidas s-series [58]. Mediante dichas aproximaciones se generan *Elementary paths* que se introducen en el planificador y mediante cálculos trigonométricos se calcula el parámetro característico. El problema del cambio de carril se resuelve particularizando un *BiElementary path*, como se observa en las Figuras 30 y 31.

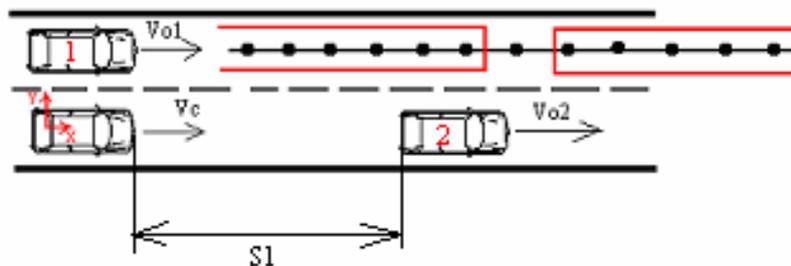


Figura 30. Problema de cambio de carril con obstáculos móviles.

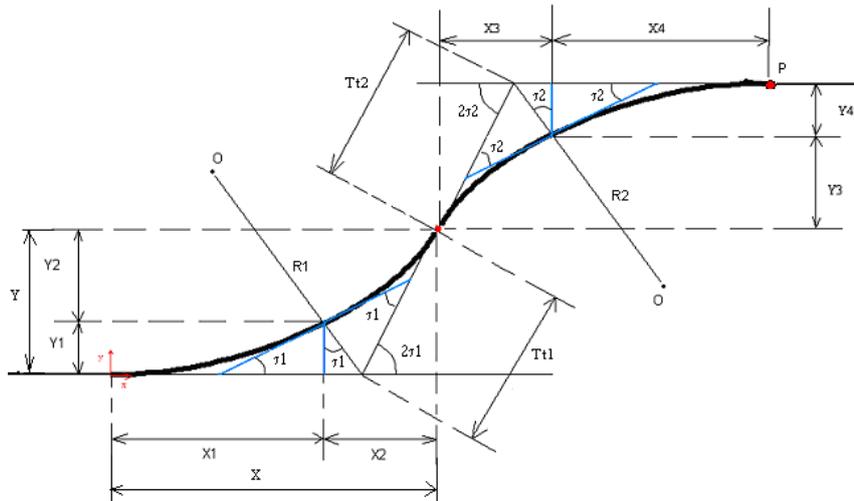


Figura 31. BiElementary path aplicado al problema de cambio de carril.

Después del proceso de planificación en el que se tiene en cuenta el movimiento relativo de todos los obstáculos y del vehículo propio se obtienen una serie de soluciones factibles, tal y como se observa en la Figura 32.

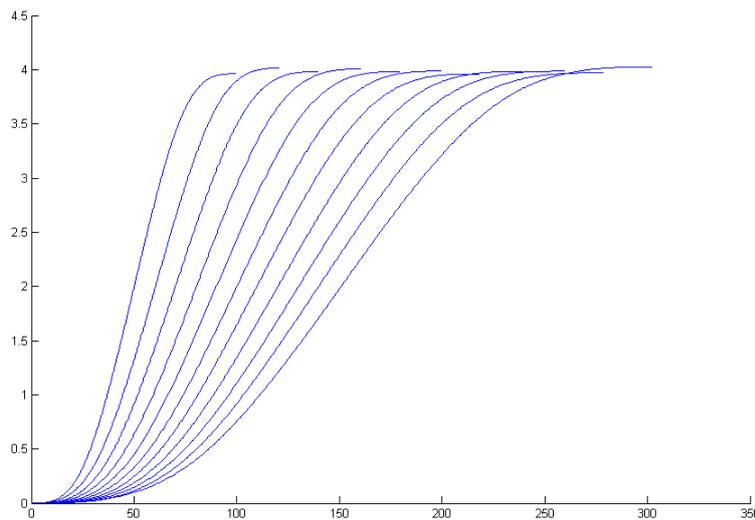


Figura 32. Trayectorias factibles obtenidas por el planificador.

Debido al hecho de que la generación de trayectorias clotoidales requiere de una computación complicada de las integrales de Fresnel, en [37] se presenta un generador *on-line* de trayectorias clotoidales mediante el uso de una curva paramétrica genérica. Para ello previamente se realiza una aproximación *off-line* de las integrales de Fresnel mediante Curvas Rational Bezier (RBC), las cuales posteriormente se utilizan para la generación de trayectorias *on-line* a partir del escalado y la rotación de las curvas unitarias. Esta aproximación garantiza que la curva resultante tiene el mismo comportamiento que la Clotoide. El método se particulariza para maniobras de adelantamiento de vehículos, de modo que se utiliza una composición de dos *Elementary paths* iguales cada uno construido mediante la unión de dos tramos clotoidales idénticos y simétricos.

En la Figura 33 se muestra un ejemplo de adelantamiento mediante el simulador que los autores han creado para las demostraciones prácticas.



Figura 33. Maniobra de cambio de carril con evitación de obstáculo.

Pese a tener un carácter genérico sobre planificación de trayectorias de curvatura continua para robótica móvil, [38] se centra en el problema de maniobras de aparcamiento (ver Figura 34), con el objetivo de generar trayectorias con límites superiores de curvatura y su derivada.

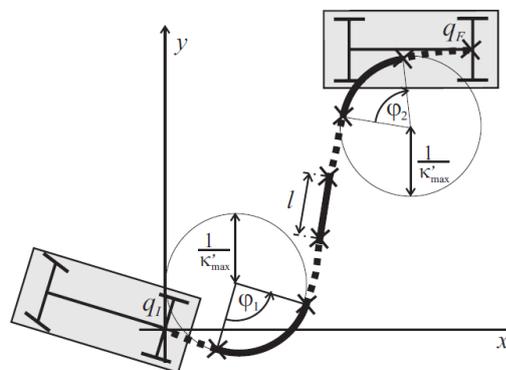


Figura 34. Ejemplo de aparcamiento en paralelo.

Para resolver el problema se utilizan las tres primitivas clásicas: segmentos de línea recta, segmentos circulares y *continuous-curvature turns* o *CC-turns*. No obstante, una de las aportaciones más interesantes es que realizan una clasificación de los *CC-turns* y describen el movimiento a lo largo de los distintos tipos de *CC-turn*, tal y como se ve en la Figura 35.

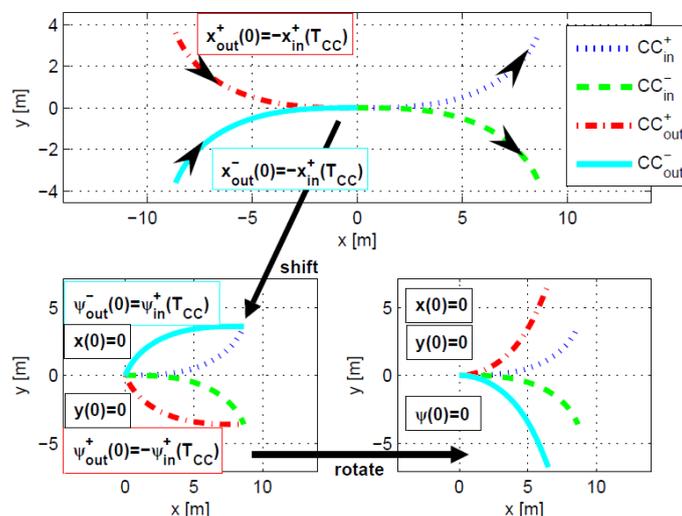


Figura 35. Cuatro *CC-turns* completos y diferentes en el plano *XY*.

En [39] se presenta un nuevo método para computar de forma simple y rápida el *sharpness* y la curvatura de segmentos de Clotoide en trayectorias continuas. Como ya se sabe los segmentos clotoidales son necesarios en la generación de caminos como curvas de transición para unir los segmentos rectos con los arcos circulares. Este algoritmo forma parte de un generador de trayectorias y funciona como planificador de movimientos que genera trayectorias suaves, naturales y fáciles de seguir, utilizando el mínimo número de primitivas para llegar a la configuración final deseada, de forma similar a como actúan los humanos cuando conducen.

Entre otras aplicaciones destaca la maniobra de cambio de carril, la cual se usa como ejemplo para demostrar el correcto funcionamiento del algoritmo de computación propuesto en este artículo, tal y como se puede ver en la Figura 36.

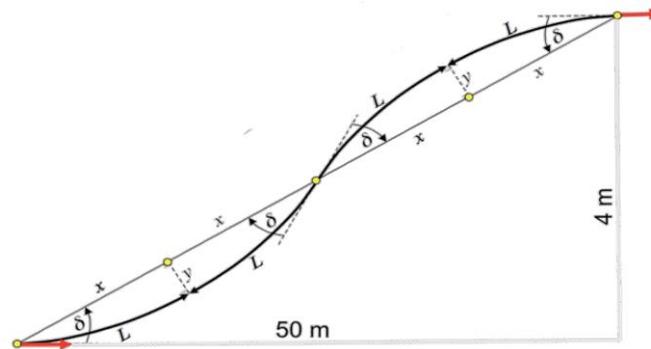


Figura 36. Ejemplo de trayectoria formada por 4 Clotoides simétricas para resolver el problema de cambio de carril.

## 2.2.4. Trayectorias Suaves en Otras Aplicaciones

Como ya se ha comentado en apartados anteriores, la Clotoide o espiral de Cornu posee una propiedad geométrica interesante: su curvatura es proporcional a la longitud del arco medida desde el origen de coordenadas. La ecuación que rige esta curva es:

$$R \cdot l = A^2 \quad (15)$$

Esta propiedad ha hecho que este tipo de curvas sean muy utilizadas por los ingenieros de caminos y topógrafos para obtener los trazados de las carreteras sin discontinuidades, es decir, con una variación de la curvatura constante que garantice una total comodidad de los ocupantes del vehículo. Por lo tanto, no solo los investigadores utilizan las Clotoides para generación de trayectorias y navegación en robótica móvil, sino también profesionales como los ingenieros de caminos o los topógrafos utilizan estas curvas para el diseño de las carreteras y/o vías de tren [40] y en la legislación de muchos países así se contempla [41], [42], [43], [44].

En [40] podemos encontrar la forma que tienen de calcular el parámetro característico de una Clotoide para el diseño de un tramo de carretera:

$$A^2 = \frac{v^3}{J} \quad (16)$$

Donde  $v$  es la velocidad del vehículo (m/s),  $J$  es el momento de inercia y  $A$  es el parámetro que define la Clotoide.

También se utiliza una ecuación parecida para condicionar el radio del tramo circular:

$$V^2 = 127 \cdot R \cdot f \quad (17)$$

Donde  $R$  es el radio del arco circular y  $f$  es el coeficiente de rozamiento transversal.

Finalmente hay que destacar que se han utilizado Clotoides para el diseño y análisis de montañas rusas, ya que hay trabajos de investigación relacionados [45].

### 2.2.5. Métodos de Aproximación de Clotoides

Dadas las características de las Clotoides como trayectorias suaves, también se han utilizado para realizar mejoras en aplicaciones de diseño y modelado [46], [47]. Por eso, a la larga se han incorporado en muchos programas de diseño CAD. En dichos programas no hay limitaciones de capacidad de procesamiento ni problemas temporales, pero en robótica móvil sí que hay requerimientos de tiempo real y el problema de las Clotoides es que no tienen solución analítica cerrada. La evaluación numérica de las integrales de Fresnel tiene mucho coste computacional y además su precisión depende del número de puntos y del tipo de discretización utilizado, por lo que es inviable. Por ese motivo en los últimos años los investigadores han intentado obtener aproximaciones precisas de las Clotoides con un coste computacional lo más bajo posible.

Primero en [48] y después en [49] se desarrollan los primeros trabajos relacionados expresamente con la aproximación numérica de las integrales de Fresnel. Se obtienen dos expresiones distintas, la primera para valores de  $x \leq 4$  y la segunda para valores de  $x \geq 4$ . Dichas aproximaciones se obtienen a través del  $\tau$ -method of Lanczos, que define las integrales como:

$$f(x) = \int_0^x \frac{e^{-it}}{\sqrt{2\pi t}} \cdot dt = C(x) - iS(x) \quad (18)$$

Las aproximaciones obtenidas son las siguientes:

$$\text{Para } 0 \leq x \leq 4 \rightarrow f(x) = e^{-ix} \sqrt{\frac{x}{4}} \cdot \sum_{n=0}^{11} (a_n + ib_n) \left(\frac{x}{4}\right)^n \quad (19)$$

$$\text{Para } x \geq 4 \rightarrow f(x) = \frac{1-i}{2} + e^{-ix} \sqrt{\frac{4}{x}} \cdot \sum_{n=0}^{11} (a_n + ib_n) \left(\frac{4}{x}\right)^n \quad (20)$$

Hay que destacar que el error máximo de la primera aproximación es de  $1.6 \cdot 10^{-9}$  y el de la segunda de  $0.5 \cdot 10^{-9}$ , lo cual demuestra que la aproximación es bastante buena.

Los valores numéricos de los coeficientes  $a_n$ ,  $b_n$ ,  $c_n$  y  $d_n$  que aparecen en las ecuaciones (19) y (20) son los que se muestran en la Tabla 5.

$a_0 = +1.595769140$	$b_0 = -0.000000033$	$c_0 = 0$	$d_0 = +0.199471140$
$a_1 = -0.000001702$	$b_1 = +4.255387524$	$c_1 = -0.024933975$	$d_1 = +0.000000023$
$a_2 = -6.808568854$	$b_2 = -0.000092810$	$c_2 = +0.000003936$	$d_2 = -0.009351341$
$a_3 = -0.000576361$	$b_3 = -7.780020400$	$c_3 = +0.005770956$	$d_3 = +0.000023006$
$a_4 = +6.920691902$	$b_4 = -0.009520895$	$c_4 = +0.000689892$	$d_4 = +0.004851466$
$a_5 = -0.016898657$	$b_5 = +5.075161298$	$c_5 = -0.009497136$	$d_5 = +0.001903218$
$a_6 = -3.050485660$	$b_6 = -0.138341947$	$c_6 = +0.011948809$	$d_6 = -0.017122914$
$a_7 = -0.075752419$	$b_7 = -1.363729124$	$c_7 = -0.006748873$	$d_7 = +0.029064067$
$a_8 = +0.850663781$	$b_8 = -0.403349276$	$c_8 = +0.000246420$	$d_8 = -0.027928955$
$a_9 = -0.025639041$	$b_9 = +0.702222016$	$c_9 = +0.002102967$	$d_9 = +0.016497308$
$a_{10} = -0.150230960$	$b_{10} = -0.216195929$	$c_{10} = -0.001217930$	$d_{10} = -0.005598515$
$a_{11} = +0.034404779$	$b_{11} = +0.019547031$	$c_{11} = +0.000233939$	$d_{11} = +0.000838386$

Tabla 5 Valores numéricos de los coeficientes  $a_n$ ,  $b_n$ ,  $c_n$  y  $d_n$ .

Posteriormente, en [50] se realiza otra aproximación numérica, pero esta vez basándose en la expansión de Maclaurin de las integrales de Fresnel (21) y (22). Durante todo el artículo el autor intenta demostrar las ventajas de su aportación.

$$C(x, \alpha) = x^{(1-\alpha)} \sum_{v=0}^{\infty} \frac{(-1)^v x^{2v}}{(2v+1-\alpha)(2v)!} \quad (21)$$

$$S(x, \alpha) = x^{(2-\alpha)} \sum_{v=0}^{\infty} \frac{(-1)^v x^{2v}}{(2v+2-\alpha)(2v+1)!} \quad (22)$$

En [51] se hace una aproximación de las integrales de Fresnel mediante un modelo de Chebyshev, partiendo de una aproximación previa que utiliza la expansión en serie de Taylor.

$$C(x) = \frac{1}{2} + f(x) \sin\left(\frac{\pi x^2}{2}\right) - g(x) \cos\left(\frac{\pi x^2}{2}\right) \quad (23)$$

$$S(x) = \frac{1}{2} - f(x) \cos\left(\frac{\pi x^2}{2}\right) - g(x) \sin\left(\frac{\pi x^2}{2}\right) \quad (24)$$

El objetivo principal del artículo es presentar una aproximación racional eficiente para  $S(x)$  y  $C(x)$  cuando  $|x|$  es pequeño y para  $f(x)$  y  $g(x)$  en el resto de casos. Hay que remarcar que se obtienen errores relativos menores de  $10^{-19}$  en la mayoría de casos.

$$\text{Para } |x| \leq 1.2 \text{ y } 1.2 \leq |x| \leq 1.6 \rightarrow \begin{cases} C_{lm}(x) = x R_{lm}(x^4) \\ S_{lm}(x) = x^3 R_{lm}(x^4) \end{cases} \quad (25)$$

$$\text{Para } 1.6 \leq |x| \leq 1.9 \text{ y } 1.6 \leq |x| \leq 1.9 \rightarrow \begin{cases} f_{lm} = x^{-1} R_{lm}(x^{-4}) \\ g_{lm} = x^{-3} R_{lm}(x^{-4}) \end{cases} \quad (26)$$

$$\text{Para } 2.4 \leq |x| \rightarrow \begin{cases} f_{lm} = x^{-1} \left\{ \frac{1}{\pi} + x^{-4} R_{lm}(x^{-4}) \right\} \\ g_{lm} = x^{-3} \left\{ \frac{1}{\pi^2} + x^{-4} R_{lm}(x^{-4}) \right\} \end{cases} \quad (27)$$

Siendo  $R_{lm}(\cdot)$  una función racional.

En [52] se hace una aproximación racional a las integrales de Fresnel, también mediante la expansión en serie de Taylor. No obstante la diferencia es que se utiliza la forma polar.

$$C(x) = \frac{1}{2} - R_{lm}(x) \sin \left[ \frac{\pi}{2} \cdot (A_{jk}(x) - x^2) \right] \quad (28)$$

$$S(x) = \frac{1}{2} + R_{lm}(x) \cos \left[ \frac{\pi}{2} \cdot (A_{jk}(x) - x^2) \right] \quad (29)$$

Donde las funciones  $R_{lm}$  y  $A_{jk}$  son aproximaciones racionales de la forma:

$$R_{lm} = \frac{\sum_{i=0}^l c_i x^i}{\sum_{i=0}^m d_i x^i} \quad (30)$$

$$A_{jk} = \frac{\sum_{i=0}^j a_i x^i}{\sum_{i=0}^k b_i x^i} \quad (31)$$

Los coeficientes se calculan de la siguiente forma:

$$f(x) = \frac{1+0.882x}{2+1.772x+3.017x^2} \quad (32)$$

$$g(x) = \frac{1}{2+4.167x+3.274x^2+6.890x^3} \quad (33)$$

En [53] y posteriormente en [54] se hace una aproximación de las integrales de Fresnel mediante un método recursivo con un error  $|e| < 1 \cdot 10^{-9}$ . Se asume que la aproximación continua de las componentes de la Clotoide es la misma que en [51] (ver ecuaciones (23) y (24)). Se consigue una expresión genérica para aproximar las integrales de Fresnel y compensar los errores mediante una ligera modificación del método utilizado en [52]. Los parámetros  $f(x)$  y  $g(x)$  que aparecen en las ecuaciones (32) y (33) se modifican ligeramente:

$$f(x) \approx f_0(x) = \frac{1+0.926x}{2+1.792x+3.104x^2} + \varepsilon(x) \quad (34)$$

$$g(x) \approx g_0(x) = \frac{1}{2+4.142x+3.492x^2+6.67x^3} + \varepsilon(x) \quad (35)$$

$$|\varepsilon(x)| \leq 2 \cdot 10^{-3}, \quad 0 < x < \infty \quad (36)$$

Además se propone una forma de calcularlos de forma recursiva para cualquier orden.

$$f(x) = \sum_{n=0}^{\infty} f_n \cdot x^{-2n-1} \quad (37)$$

$$g(x) = \sum_{n=0}^{\infty} g_n \cdot x^{-2n-1} \quad (38)$$

Concretamente se particulariza para el caso de orden 11, de modo que aparece una tabla (Tabla 6) con los valores que deben tener los parámetros para un ajuste de dicho orden. Así no es necesario recalcular cada vez la ecuación, sino que mediante la sustitución de los valores se consigue una expresión más sencilla y por consiguiente mucho más rápida de computar.

$n$	$f_n$	$g_n$
0	0.318309844	0
1	9.34626E-08	0.101321519
2	-0.09676631	-4.07292E-05
3	0.000606222	-0.152068115
4	0.325539361	-0.046292605
5	0.325206461	1.622793598
6	-7.450551455	-5.199186089
7	32.20380908	7.477942354
8	-78.8035274	-0.695291507
9	118.5343352	-15.10996796
10	-102.4339798	22.28401942
11	39.06207702	-10.89968491

**Tabla 6. Valores de los parámetros para un ajuste de orden 11.**

En [55] aproximan la Clotoide en el tramo  $[0 - \pi/2]$  mediante curvas Bezier y *B-splines*. Además, se demuestra que el error de aproximación tiende a cero a medida que se incrementa el número de puntos utilizado.

El procedimiento para la aproximación es utilizar un único desarrollo de Taylor y posteriormente mejorar esta aproximación mediante una *B-spline* de grado reducido, lo cual aporta ciertas ventajas haciendo posible la incorporación a programas de CAD/CAM. No obstante, esta estrategia resulta bastante costosa de implementar porque requiere fórmulas de complejidad elevada.

Existen aproximaciones de la Clotoide basadas en la polinomiales estándar, la más básica es la espiral cúbica, cuya ecuación genérica es:

$$y = \left(\frac{1}{A^2}\right) \cdot x^3 \tag{39}$$

Después se mejoró y se obtuvo el modelo utilizado en [56] y [57], cuya ecuación caracterizada para la Clotoide es la siguiente:

$$y = \left(\frac{1}{R_{ini}}\right) \cdot \frac{x^2}{2} + \left(\frac{1}{A^2}\right) \cdot \frac{x^3}{6} \tag{40}$$

El problema es que estos modelos son muy poco precisos y el error aumenta conforme el radio final es más pequeño. Por eso, en [56] se describe un nuevo modelo y un método para la estimación precisa de la geometría del camino a seguir por un automóvil. Para ello se utiliza un modelo polinomial de tercer orden. En primer lugar se define la variación de la curvatura en la Clotoide como:

$$C(x) = c_0 + c_1 \cdot x \tag{41}$$

Donde  $c_0$  y  $c_1$  son los coeficientes de la Clotoide.

También definen la aproximación polinomial de la Clotoide como:

$$y = c_0 \frac{x^2}{2} + c_1 \frac{x^3}{6} \tag{42}$$

El único inconveniente es que los parámetros anteriores son variables en el tiempo y dependen directamente de la velocidad del vehículo.

Por otro lado, la geometría de la carretera se divide en dos segmentos clotoiales distintos pero contiguos (ver Figura 37), de modo que cada segmento se modela como una curva polinomial simple.

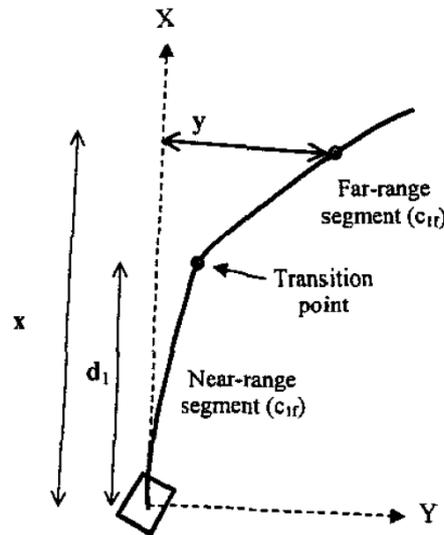


Figura 37. Modelo de 2 Clotoides contiguas.

Se obtiene una expresión paramétrica cerrada del mismo orden polinómico que los modelos anteriores. El modelo polinomial que utilizan para estimar la curva es el siguiente:

$$\text{Para } 0 < x < d_1 \rightarrow y = c_0 \frac{x^2}{2} + c_{1n} \frac{x^3}{6} \tag{43}$$

$$\text{Para } d_1 < x \leq d_2 \rightarrow y = c_0 \frac{x^2}{2} + c_{1n} \left( \frac{d_1^3}{6} + \frac{x(x-d_1)d_1}{2} \right) + c_{1f} \frac{(x-d_1)^3}{6} \tag{44}$$

También se describe un método basado en estimación recursiva del modelo de la carretera. La estimación de los parámetros  $c_0$ ,  $c_{1n}$ ,  $c_{1f}$  se realiza mediante la utilización del filtro de Kalman localizando marcas en la curva. La simulación de este método se puede ver en la Figura 38.

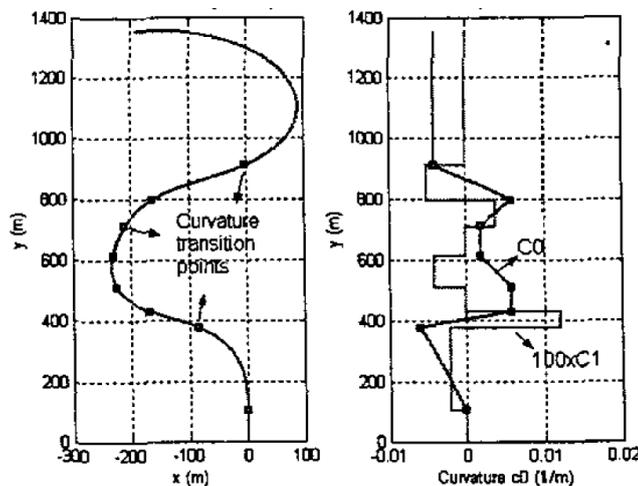


Figura 38. Simulación del modelo de Clotoides contiguas. Geometría de carretera (izquierda) y curvatura (derecha).

Finalmente se compara el método propuesto con las aproximaciones anteriores para demostrar la viabilidad y mayor exactitud del mismo. Hay que destacar que esta mayor precisión no implica un aumento en el coste computacional y/o la sensibilidad frente al ruido. Lo que sí que se remarca en el artículo es que la alta precisión de la estimación de la geometría de la ruta viaria será útil de forma directa en aplicaciones que se basan en la detección de objetivos en la trayectoria de avance del vehículo, como por ejemplo en el control de velocidad de crucero o en el sistema de aviso de colisión que incorporan algunos automóviles.

La aproximación de la Clotoide a una polinomial cúbica se usa ampliamente para el modelado de carriles de carretera en aplicaciones de vehículos con un control adaptativo de velocidad, alarmas de colisión y de salida de carril, navegación autónoma, etc. Pese a su uso extendido, en [57] se demuestra que la estimación de curvas de carreteras mediante una aproximación de la polinomial de tercer orden es un método muy pobre, ya que es muy sensible al ruido de los sensores o al offset lateral del vehículo en la carretera.

En este artículo se demuestra que errores sistemáticos pequeños o aleatorios (ruido) pueden hacer que la geometría de la carretera sea sustancialmente similar, pero en cambio provocan diferencias significativas en los parámetros de las Clotoides.

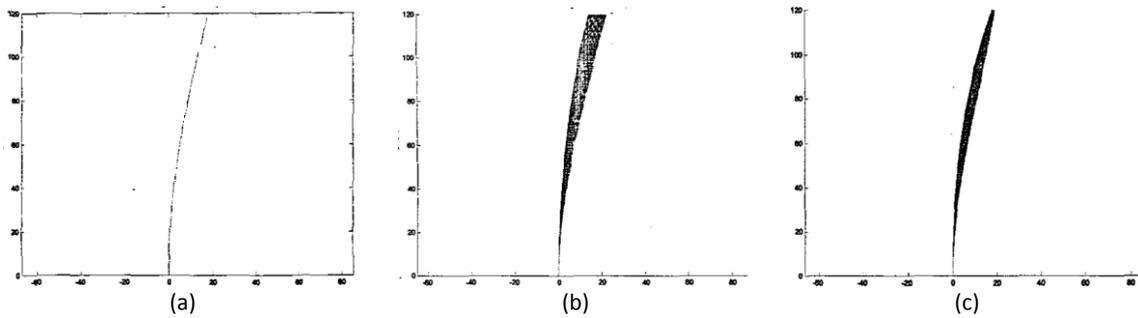


Figura 39. (a) Clotoide modelo, (b) Clotoide con errores angulares y (c) Clotoide con errores de offset lateral.

Los autores de este artículo utilizan fusión de datos sensorial para estimar el modelo de la carretera y además proponen un modelo más detallado y completo para la ecuación de tercer orden.

$$y = c_1 \frac{x^3}{6} + c_0 \frac{x^2}{2} + c_2 x + c_3 \quad (45)$$

Los términos  $c_2$  y  $c_3$  los utilizan para modelar el offset lateral.

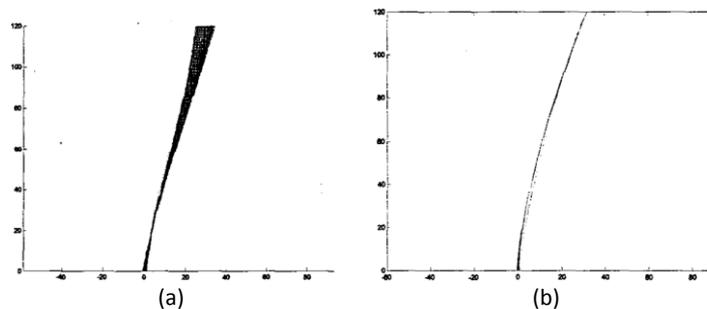


Figura 40. (a) Estimación del modelo de Clotoide tras la fusión sensorial del modelo de parámetros clásico de la Clotoide. (b) Estimación del modelo de Clotoide transformado al espacio rango-offset.

[58] describe el método para realizar una aproximación polinomial de las Clotoides mediante *s-power series*. Dichas series se basan en las expansiones de Hermite que utilizan dos puntos a diferencia de los desarrollos de Taylor que sólo usan uno. La ventaja de esta aproximación es que garantiza el contacto en los extremos, consiguiendo un error muy pequeño con un bajo grado del polinomio. De hecho, tal y como se demuestra en el artículo, con una polinomial de orden 3 la diferencia entre la Clotoide y su aproximación es inapreciable. La principal ventaja de este método es el hecho de que subdividir los tramos para conseguir un orden menor no deja huecos y discontinuidades como cualquiera de las anteriores técnicas (ver Figura 41).

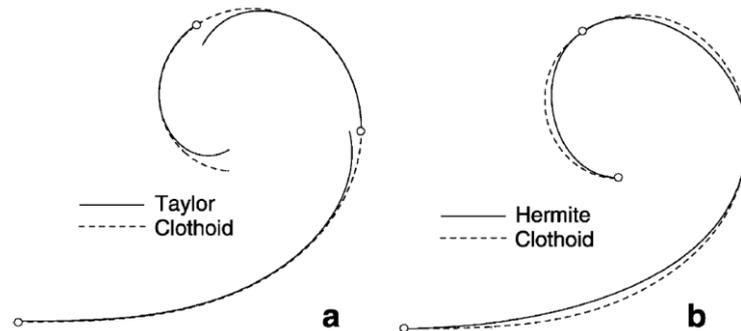


Figura 41. Comparativa entre aproximaciones: (a) Taylor y (b) Hermite.

En [59] la aproximación se hace mediante un arco de *spline*. La principal ventaja es que así es muy sencillo de modelar el *lay out* y el *offset* de la Clotoide. Se muestran ejemplos de aproximación para verificar la gran precisión de este método. También se demuestra que si el arco de *spline* se divide en  $n$  arcos, el error de la aproximación es de orden  $O(1/n^2)$

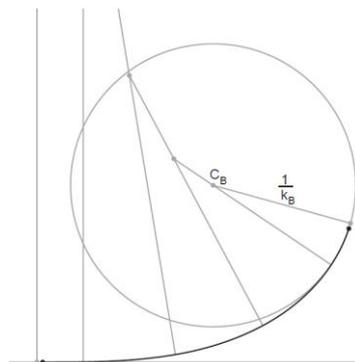


Figura 42. Ejemplo de unión recta y círculo mediante una Clotoide modelada como un arco de *spline*.

La ecuación o curva de Bezier fue originalmente desarrollada por Pierre Bezier en 1970 para operaciones de CAD/CAM. Este ingeniero de la empresa Renault, utilizó estas curvas para representar de forma eficiente las curvas de los vehículos en los programas de diseño que utilizaba.

La representación mediante puntos de control de Bezier es una de las técnicas más extendidas, más intuitivas y con un menor coste computacional. Por ese motivo existe una fuerte labor investigadora en intentar acabar traduciendo cualquier nueva representación polinomial a ecuación de Bezier y a su variante racional la R-Bezier o RBC (*Rational Bezier Curve*).

Aunque ya se han presentado y explicado ciertas aportaciones que utilizan las curvas Bezier como aproximación de trayectorias clotoidales, más recientemente, en [60] se estudian las ventajas de aproximar la Clotoide mediante curvas RBC. Además, a nivel de aplicación se presenta una técnica novedosa para la generación en tiempo real de caminos de curvatura continua para robots móviles.

Como primer paso, ajustando una RBC se obtiene una aproximación *off-line* de las integrales de Fresnel. La ecuación de una curva RBC es como sigue.

$$P(u) = \frac{\sum_{k=0}^N w_k \cdot C_k \cdot \frac{N!}{k!(N-k)!} u^k \cdot (1-u)^{N-k}}{\sum_{k=0}^N w_k \cdot \frac{N!}{k!(N-k)!} u^k \cdot (1-u)^{N-k}} \quad (46)$$

Dónde:

- $C_k$ : Puntos de control de Bezier.
- $W$ : Pesos de los puntos de control de Bezier.
- $N$ : Orden de la ecuación de Bezier (número de puntos de control).
- $u$ : Parámetro intrínseco  $[0 - 1]$ .

Es conocido que la aproximación mediante una curva Bezier no pasa a través de los puntos de control inicial y final. En cambio, cuando la aproximación se realiza mediante RBC el ajuste pasa tanto por el primero como por el último de los puntos que se desea ajustar (ver Figura 43). Esto es muy deseable en robótica móvil, ya que el objetivo es unir dos configuraciones dadas y por las cuales es obligatorio pasar para un instante determinado.

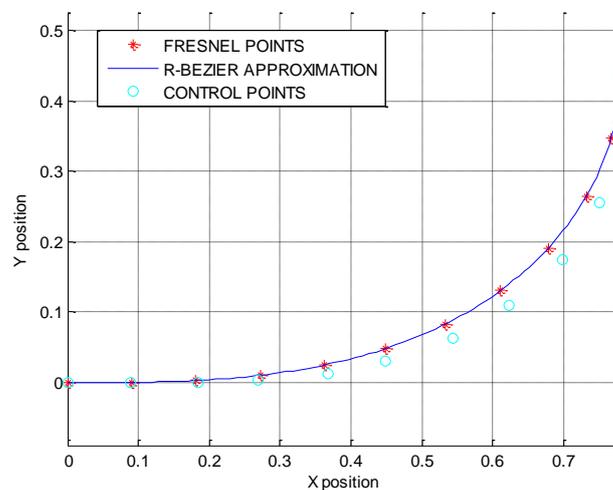


Figura 43. Aproximación de los puntos de Fresnel con una curva R-Bezier de orden  $N=11$ .

Una vez se ha calculado y fijado como base la aproximación de la Clotoide unitaria, *on-line* sólo es necesario calcular la rotación y la translación correspondiente para generar la trayectoria deseada. En dicho proceso, los coeficientes, los pesos y los puntos de control se mantienen invariables. Los autores presentan el método genérico para la obtención de los pesos y los puntos de control, aunque caracterizan su aproximación para orden  $N = 11$ , de modo que presentan la tabla correspondiente de pesos y puntos (ver Tabla 7), para las coordenadas  $X$  e  $Y$ .

$i$	C		S	
	$C_i$	$w_i$	$C_i$	$w_i$
0	0	1	0	1
1	0.0909	$1-77 \cdot 10^{-7}$	0	1
2	0.1818	$1+77 \cdot 10^{-8}$	0	1
3	0.2727	$1-17 \cdot 10^{-7}$	0.003	$1-27 \cdot 10^{-6}$
4	0.3636	$1+15 \cdot 10^{-7}$	0.012	$1+92 \cdot 10^{-7}$
5	0.4540	$1-18 \cdot 10^{-7}$	0.031	$1+14 \cdot 10^{-7}$
6	0.5422	$1-82 \cdot 10^{-8}$	0.0634	$1-47 \cdot 10^{-7}$
7	0.6251	$1-68 \cdot 10^{-8}$	0.1107	$1+36 \cdot 10^{-7}$
8	0.6973	$1-22 \cdot 10^{-8}$	0.1755	$1-19 \cdot 10^{-7}$
9	0.7513	$1-40 \cdot 10^{-9}$	0.2564	$1+66 \cdot 10^{-8}$
10	0.7797	$1-47 \cdot 10^{-8}$	0.3473	$1-27 \cdot 10^{-9}$
11	0.7798	1	0.4382	1

Tabla 7. Pesos y puntos de control para las coordenadas X e Y (C y S).

Además, se garantiza que la aproximación mediante RBC de orden bajo tiene el mismo comportamiento que la Clotoide original. Por lo tanto, en este trabajo de investigación destaca como principal aportación un planificador en tiempo real que utiliza trayectorias clotoidales y que es capaz de unir en un plano cualquier configuración inicial y final mediante el camino más corto.

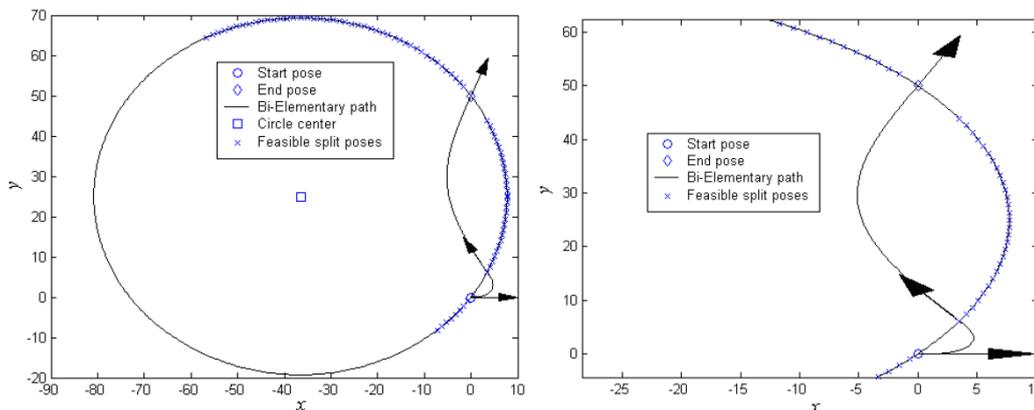


Figura 44. Poses Inicial (0, 0, 0) y Final (0, 50, 1. 2) y ruta generada.

### 2.2.6. Conclusión

Además de todos los trabajos expuestos que se han publicado en congresos y revistas internacionales, también se han realizado numerosas tesis y trabajos extensos de investigación relacionados con la generación de trayectorias de curvatura continua. Debido a la limitación en cuanto a extensión de la memoria de la presente Tesina no se comentan los desarrollos realizados en algunas de las tesis consultadas. No obstante sí se van a mencionar a nivel de referencia bibliográfica, con tal de destacar que se trata de consultas obligatorias o al menos recomendadas en caso de tener que trabajar en este campo de la robótica. Entre los trabajos de investigación y tesis más importantes destacan los siguientes [61], [62], [63], [64], [65], [66], [67], [68], [69].

Tras el estudio realizado se llega a la conclusión de que pese a que se ha trabajado con Clotoides para modelar las carreteras y/o generar las trayectorias mediante un planificador de movimiento, no se han desarrollado investigaciones en relación a control de robots para seguimiento de líneas utilizando Clotoides.

## **2.3. El problema del seguimiento de línea**

Las Clotoides también se utilizan en otros campos, como por ejemplo para intentar estimar y modelar las curvas de las carreteras. Algunos métodos de identificación de la curva se basan en la localización de unas determinadas balizas/marcas colocadas estratégicamente en la curva intentando localizarlas mediante visión artificial, GPS/MAP o usando un escáner láser. Otros métodos en cambio utilizan la identificación de los bordes del camino o carretera mediante sistemas de visión, para posteriormente generar un modelo de la curva con una ecuación polinomial o cualquier otro modelo matemático.

Por otro lado, uno de los principales objetivos de los sistemas inteligentes en vehículos de carretera o IVHS (*Intelligent Vehicle Highway Systems*) es el control lateral del movimiento del vehículo, ya que éste se usa en maniobras de adelantamiento o cambio de carril, de modo que se incremente la seguridad del sistema en caso de conducción manual-asistida, o bien se consiga un control estable en caso de conducción autónoma. El objetivo fundamental en este tipo de problemas de navegación es seguir la ruta marcada por el centro del carril o mediante cualquier otro tipo de guía. Además, se busca que la trayectoria generada sea suave bajo ciertas condiciones de operación como la velocidad del vehículo, las condiciones de la carretera o las perturbaciones externas, como por ejemplo el viento.

Es por todos los motivos anteriormente expuestos por los que en la comunidad robótica y en automoción se han realizado numerosos estudios e investigaciones relacionadas con el seguimiento de líneas y/o caminos, así como la detección y modelado de carreteras, mayoritariamente mediante sistemas de visión, aunque no siempre únicamente con visión.

### **2.3.1. Detección y Modelado del Camino**

En [70], para resolver el problema de seguimiento se utilizan unos marcadores magnéticos sobre la carretera y magnetómetros en el vehículo, a modo de sistema de referencia para la sensorización lateral. Para la etapa de control lateral del vehículo se diseña e implementa un regulador de lógica borrosa o FLC (*Fuzzy Logic Controller*) sobre un automóvil real. Los parámetros del regulador se fijan de forma manual a través de medidas empíricas obtenidas mediante un estudio de las operaciones reales que los humanos realizan al conducir.

En [71] también se introduce un algoritmo para el control de la posición lateral del vehículo en el problema del seguimiento de carriles, pero como sensor se utiliza por primera vez un sistema de visión. Dicho algoritmo de guiado del vehículo se descompone en tres pasos: el muestreo y análisis de la imagen, la determinación de la curvatura de la carretera y la determinación del desplazamiento lateral del vehículo respecto a la línea central del carril. La salida de control del sistema puede aplicarse como sistema de alarma para la asistencia a la conducción o bien directamente como acción de control al motor de direccionamiento del vehículo en caso de conducción autónoma.

En primer lugar hay que destacar que el algoritmo se basa en el efecto producido al deshacer la perspectiva de la cámara. La técnica utilizada es la *Inverse Perspective Mapping* (IPM) que se basa en el hecho de re-mapear cada píxel mediante la transformación homogénea inversa

para deshacer la perspectiva y llevarlo a su posición correspondiente en el plano de movimiento del vehículo. Esto ofrece una mejor visión del entorno mediante el conocido ojo de pájaro (*bird's eye*), lo cual facilita la detección y el modelado de la carretera (ver Figura 45).

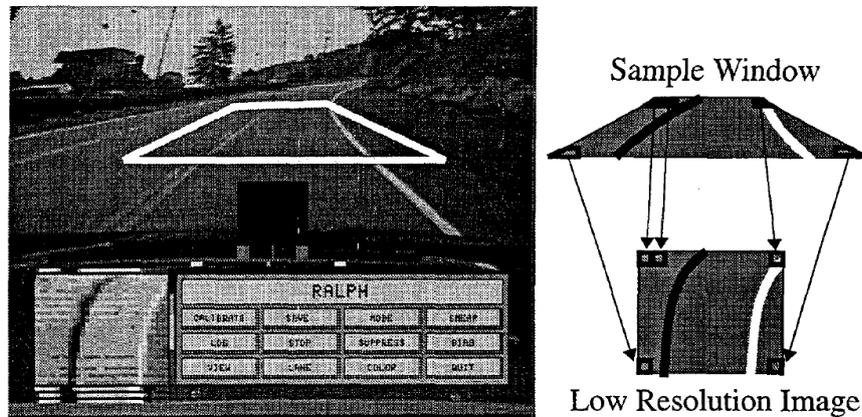


Figura 45. Proceso de detección de carretera con el sistema RALPH

Una vez la imagen está mapeada sobre el plano de movimiento del vehículo se determina la curvatura de la misma. La imagen se compara con una base de datos con imágenes modelo y de ahí se obtiene el modelo de la carretera, tal y como se observa en la Figura 46.

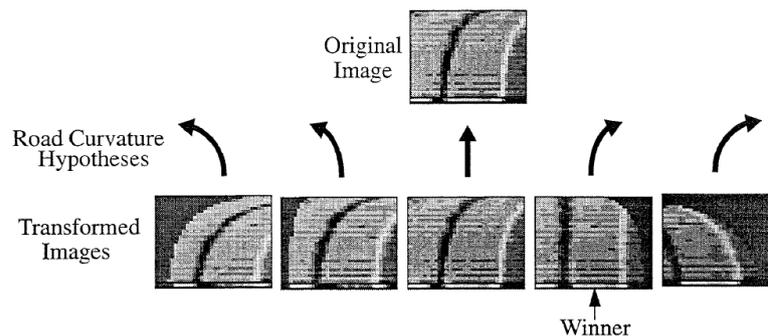


Figura 46. Proceso de estimación de la curvatura mediante comparación con otras imágenes transformadas.

Finalmente, mediante el análisis de la imagen se estima el offset lateral del vehículo. Dicho proceso es muy sencillo ya que en la imagen binarizada se tiene una línea del borde de la carretera en negro y otra en blanco, de modo que basta con detectar el flanco de bajada y el de subida para cada columna de la imagen (ver Figura 47).

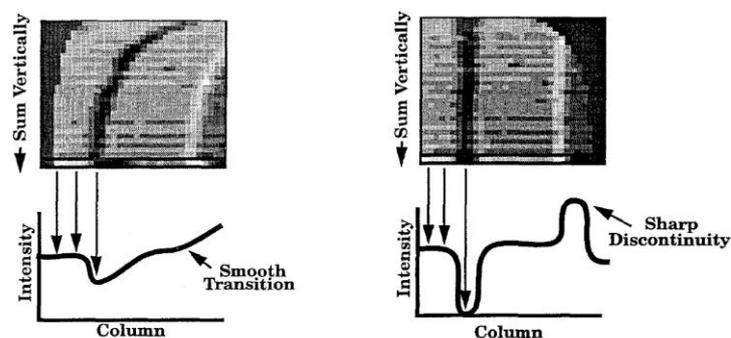


Figura 47. 2 ejemplos de detección de bordes con el método propuesto. En la parte inferior se observan los perfiles de nivel de gris de ambas imágenes.

Al igual que otros trabajos previos, [72] aborda el problema de detección y seguimiento de caminos. No obstante, aquí se presenta una aproximación mediante múltiples sensores, que se basa en el filtro EKF para estimar los parámetros de los segmentos que modelan la carretera. La diferencia fundamental respecto a otros trabajos es que además de la imagen en nivel de gris de la cámara también se utiliza la información de un mapa digital combinado con un GPS, así como las medidas de orientación y velocidad del vehículo (ver Figura 48).

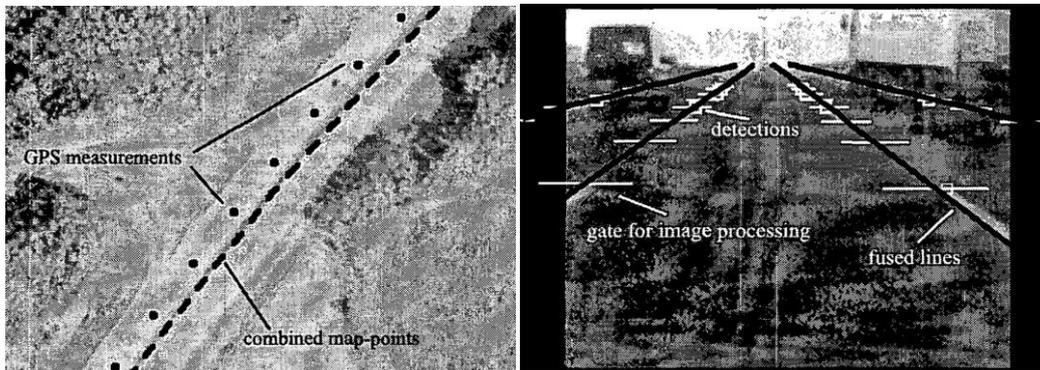


Figura 48. Datos tomados por los sensores combinados mediante fusión sensorial.

La solución combina tareas de localización con la tarea de fusión entre los datos de las líneas detectadas y los mapas digitales. Por lo tanto, el espacio de estados contiene parámetros de movimiento, así como parámetros de los tramos curvos que modelan la ruta seguida.

Se realizan simulaciones para demostrar que el algoritmo funciona satisfactoriamente en situaciones en las que el vehículo realiza cambios de carril rápidos o cuando las líneas pintadas en la carretera desaparecen momentáneamente por taparlas algún objeto del entorno.

En [73] se propone un innovador sistema de asistencia a la conducción mediante la indicación nocturna de los contornos de la carretera. Dichos indicadores se estiman a partir de la obtención de la posición 3D de los delineadores situados en los bordes de la carretera y que se extraen mediante un filtro circular. La Figura 49 muestra algunos ejemplos de delineadores reflectantes de carreteras.



Figura 49. Ejemplos de indicadores reflectantes.

Entre dos *frames* consecutivos hay que hacer un seguimiento de las marcas. En caso de utilizar dos cámaras la estimación se realiza mediante técnicas de estereovisión. Pero si sólo se dispone de una única cámara se utilizan técnicas de correspondencia de puntos entre medidas, considerando tanto las medidas obtenidas por visión como los datos conocidos relativos al desplazamiento del vehículo (velocidad y orientación).

Además, como se observa en la Figura 50, se demuestra que para la identificación de la curvatura de la carretera basta con los indicadores reflectantes, los cuales pueden ser muy útiles para este tipo de sistemas y aplicaciones, pese a que no se utilizan normalmente en los procesos de detección de los bordes de caminos.



Figura 50. Ejemplo de detección de delineadores.

Finalmente, el modelado 3D de los contornos de la carretera se realiza mediante el ajuste de los puntos obtenidos a una curva de tipo Clotoide y con una función cúbica se modela la altura de los mismos (ver Figura 51).

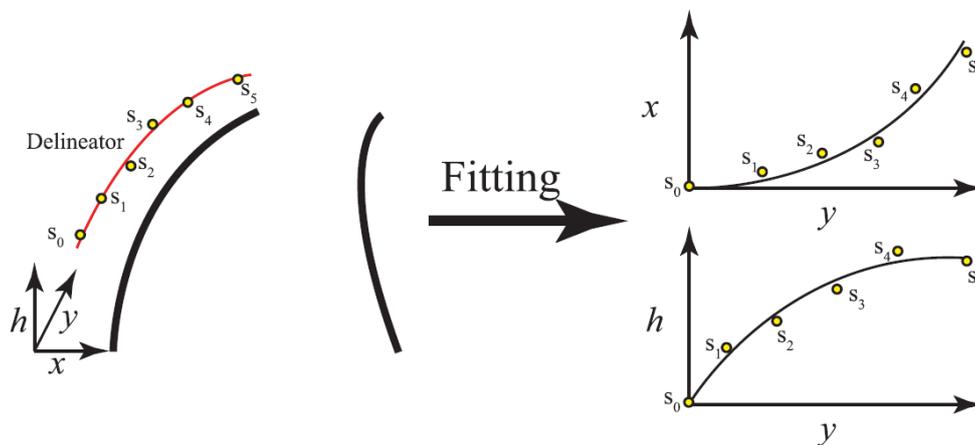


Figura 51. Clotoide y función cúbica aplicadas para estimar la posición 3D de los delineadores.

[74] presenta una primera aproximación a la identificación visual de carreteras para vehículos autónomos mediante el uso de una red neuronal de tipo *Concurrent Self-Organizing Maps* (CSOM) como clasificador. Se presentan resultados experimentales mediante la simulación por computador del modelo propuesto por los autores, aunque también se realiza una implementación sobre un robot móvil, entrenando primero el seguidor de caminos neuronal de tiempo real y testeando después su funcionamiento.

Para el entrenamiento y el testeo se utiliza un modelo neuronal, que permite capturar y etiquetar distintas imágenes de carreteras dentro de un *data set*. La ruta a identificar se cuantifica en cinco posibles salidas, tal y como se puede ver en la Figura 52.

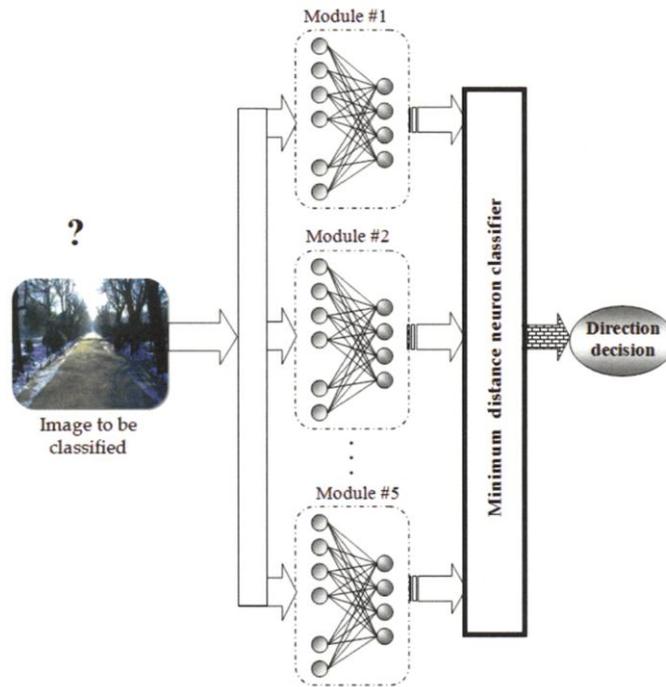


Figura 52. Etapa de reconocimiento del sistema neuronal seguidor de caminos.

En [75] se describe un preciso modelo de carretera para la localización de vehículos y la correspondencia de mapas, con el propósito de asistir en la conducción. Para la estimación de la curvatura el modelo utiliza un mapa de carreteras digital, así como puntos especiales de la carretera (puntos IP) que sirven como *way points*, tales como: puntos de intersección, puntos con gran variación de la orientación, etc. La geometría de la carretera se modela mediante reconstrucción de la misma utilizando líneas, círculos y curvas clotoidales (ver Figura 53). Finalmente, con tal de verificar el rendimiento del modelo en una situación práctica, se utiliza el sistema VICNAS con un nuevo sistema híbrido de localización del vehículo, en el que se combina un GPS comercial con sensores inerciales.



Figura 53. Sistema de visión para navegación por carreteras (Vision-based Road Navigation System: VICNAS).

Como se puede observar, han sido muchos y muy variados los métodos utilizados por los investigadores para resolver el problema de detección del camino a seguir, aunque entre todos los sensores destacan los sistemas de visión. Este tipo de elementos se han utilizado ampliamente en la bibliografía, solos o junto a otros sensores que refuercen sus medidas mediante fusión sensorial.

### 2.3.2. El Control en el Problema de Seguimiento de Caminos

Al tratarse de un sistema dinámico, cualquier robot móvil se puede controlar desde un punto de vista genérico de la teoría de control. Es por ello que se pueden utilizar reguladores PID, Fuzzy, Adaptativos, etc. También es muy común en este tipo de técnicas el uso de filtros y otras herramientas que garanticen robustez en la estimación y el control, entre las que destaca el filtro de Kalman. Es por ello que la mayoría de las técnicas propuestas para resolver el problema de seguimiento de caminos utilizan Filtros de Kalman Extendidos (EKF) en los que además de la información de los sensores se incluye también la propia cinemática y/o dinámica del vehículo autónomo.

No obstante, cuando se analiza el modelo físico del vehículo se obtienen las ecuaciones que rigen su cinemática y por tanto se pueden utilizar métodos característicos de este tipo de sistemas robóticos. Por ese motivo, en los trabajos de investigación se utilizan normalmente técnicas de control cinemático convencional, específicos de robótica móvil, aunque como ya se ha mencionado no siempre es así. A continuación se van a comentar algunas de las aportaciones más relevantes en la etapa de control para el problema de seguimiento de caminos por robots o vehículos móviles autónomos.

En [70] se dan los primeros pasos para resolver el problema de control lateral del movimiento de un vehículo. Se diseña e implementa un regulador de lógica borrosa (*Fuzzy Logic Controller* o FLC) sobre un automóvil real (Toyota Celica). Los parámetros del regulador se fijan de forma manual a través de medidas empíricas obtenidas mediante un estudio de las operaciones reales que los humanos realizan al conducir.

El regulador se estructura de forma modular mediante tres etapas bien diferenciadas (ver Figura 54): una etapa de realimentación en la que se implementan varias leyes de control según distintas variables, una etapa de pre-visualización mediante la que se estima el radio de curvatura y una etapa de planificación de ganancia para seleccionar el regulador adecuado según la situación detectada y la velocidad del vehículo.

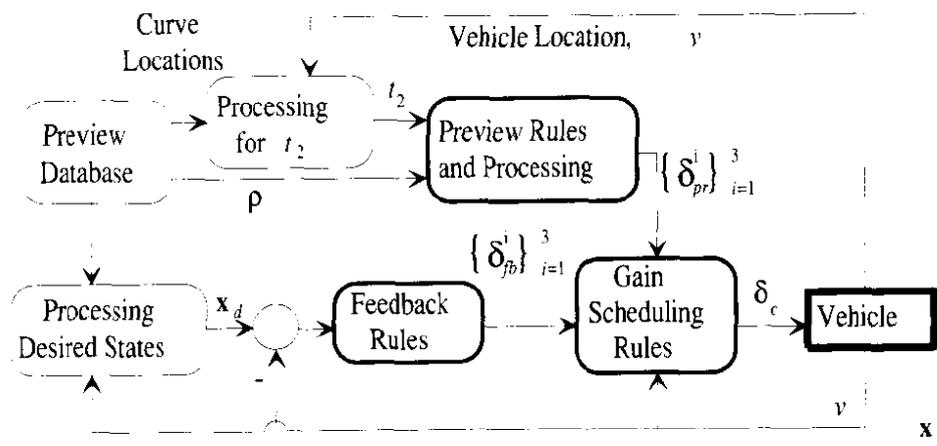


Figura 54. El regulador se estructura en tres etapas: realimentación, pre-visualización y planificación de ganancia.

Además, se realizan experimentos con el modelo implícito del vehículo, con tal de verificar el controlador diseñado. Finalmente se compara con otras técnicas como los reguladores clásicos PID (*Proportional-Integral-Derivative*) o los FSLQ (*Frequency Shaped Linear Quadratic*), los cuales utilizan el modelo explícito del vehículo.

Los autores de [76] destacan la importancia que ha tomado el problema de *Visual Servoing* (control con realimentación mediante un sensor de visión) dentro de la comunidad de visión artificial, la de control y especialmente la de robótica, en la cual se han desarrollado numerosas aplicaciones utilizando visión para cerrar el lazo de control. No obstante, a finales de los años 90 no había excesivos trabajos a nivel de desarrollo teórico, por ese motivo el presente artículo se centra en desarrollar la matemática relacionada con el problema de estimación a partir de las medidas de visión. Para ello se centran en la navegación de un robot terrestre no-holonómico, cuyo objetivo es seguir una curva continua situada en el plano de tierra. En la Figura 55 se observa el ejemplo de seguimiento de línea recta.

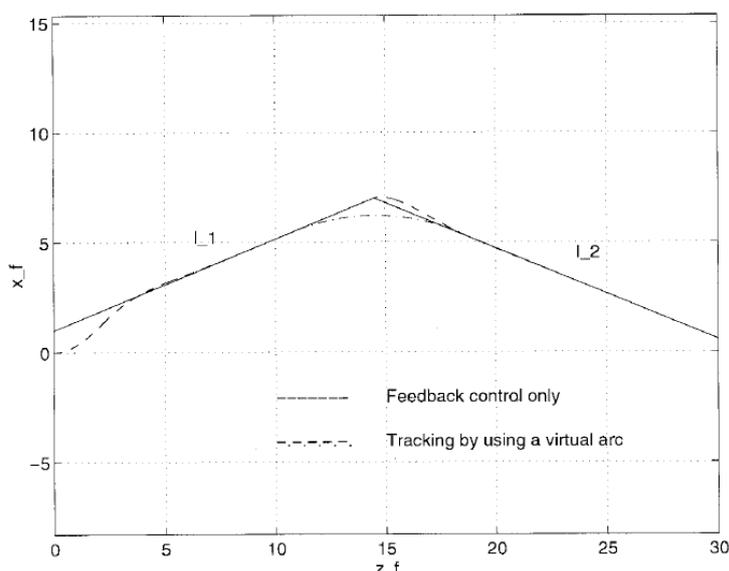


Figura 55. Comparación entre dos métodos de seguimiento de curvas de tipo recta.

En este artículo se presentan leyes de control que incluyen, por primera vez para este tipo de problemas, una etapa de estimación mediante EKF (Filtro de Kalman Extendido) para hacer el seguimiento de la curva mucho más robusto.

[77] desarrolla un sistema de seguimiento de carril basado en visión para su uso en vehículos inteligentes. El algoritmo estima la geometría y la anchura de la carretera, así como la posición y orientación del vehículo respecto a la línea central de la misma. Se utilizan técnicas básicas de visión para extraer el conjunto de marcadores candidatos a representar mejor la línea buscada en la imagen. Al igual que en trabajos anteriores, la geometría de la carretera se modela mediante curvas Clotoides, de modo que sea sencillo determinar el *offset* y ángulo de orientación del vehículo respecto de la línea central. La novedad es que para filtrar los posibles datos espurios se utiliza un método de optimización dinámica.

Además, con tal de dar mayor robustez al algoritmo también se incorpora un filtro de Kalman para estimar los parámetros y preservar la suavidad de la trayectoria, así como para predecir los parámetros de la línea detectada en la siguiente imagen. También se realiza fusión sensorial en un algoritmo de más alto nivel. El esquema completo de la arquitectura del sistema propuesto aparece en la Figura 56.

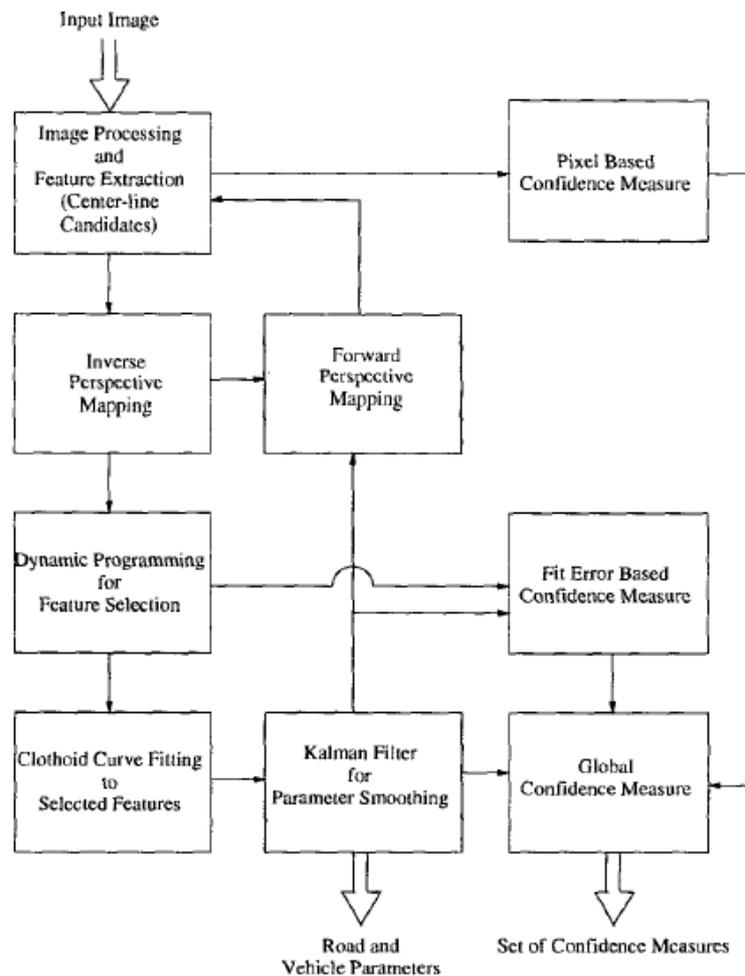


Figura 56. Esquema completo del sistema de identificación y seguimiento de líneas.

En [78] también se aborda el problema del seguimiento de líneas con robots móviles utilizando sistemas de visión artificial, pero desde el punto de vista de una estructura multi-frecuencia. Se implementan y comparan dos técnicas de control distintas: la primera estima la postura del robot a partir de la odometría (velocidades angulares de las ruedas) utilizando la información de la cámara cuando ésta está disponible (ver Figura 57); y la segunda utiliza un EKF-MR (*Multi-Rate Extended Kalman Filter*) para la fusión sensorial de datos de la imagen y los odómetros (ver Figura 58). Finalmente los resultados comparativos se pueden ver en la Figura 59.

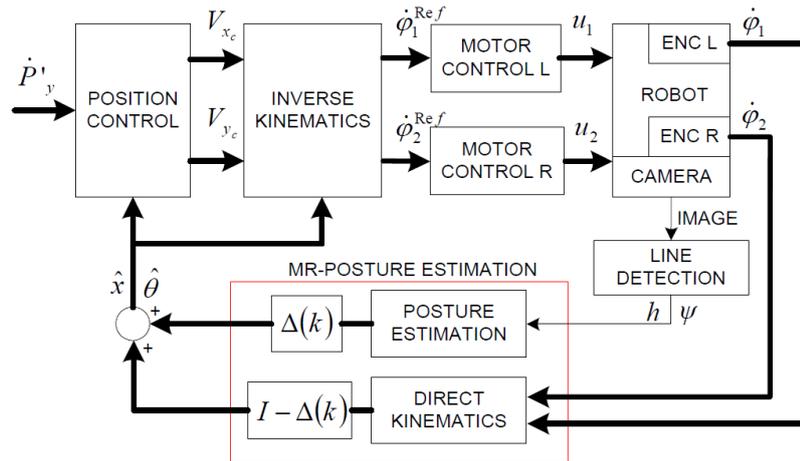


Figura 57. Arquitectura de control con Estimación Multi-Frecuencia de la postura.

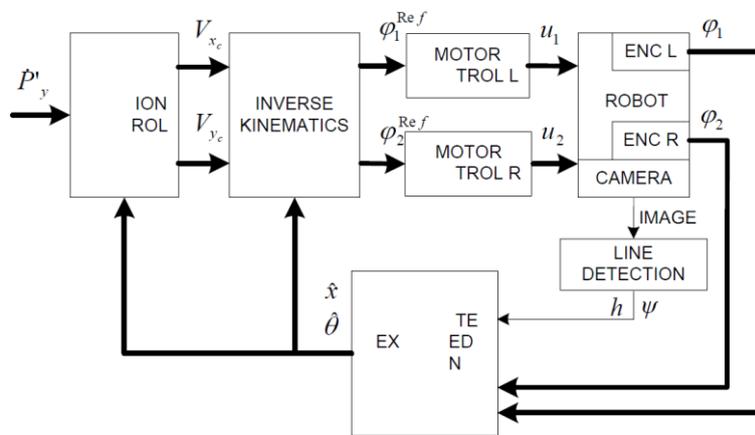


Figura 58. Arquitectura de control con Filtro de Kalman Extendido Multi-Frecuencia (MR-EKF).

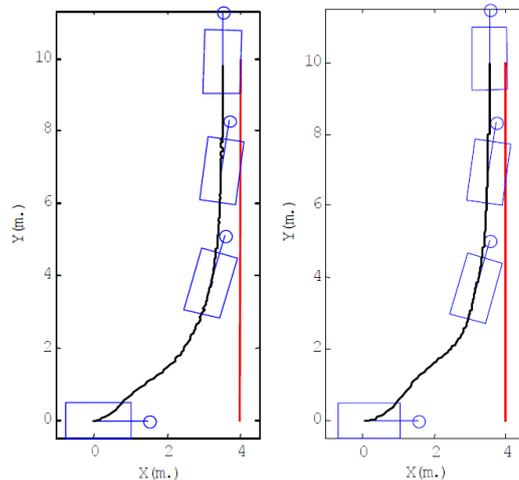


Figura 59. Comparativa de ambas técnicas con perturbación de entrada constante.

En el caso de reconocimiento de carreteras de elevada curvatura (Figura 60 y Figura 61), la aproximación basada en el modelo clotoidal tiene ciertas limitaciones, por ese motivo en [79] se propone un modelo extendido en el que se puedan tener dos segmentos de Clotoide en el campo de visión de la cámara. Se presenta un algoritmo de estimación de los parámetros de las Clotoides y del punto de transición.

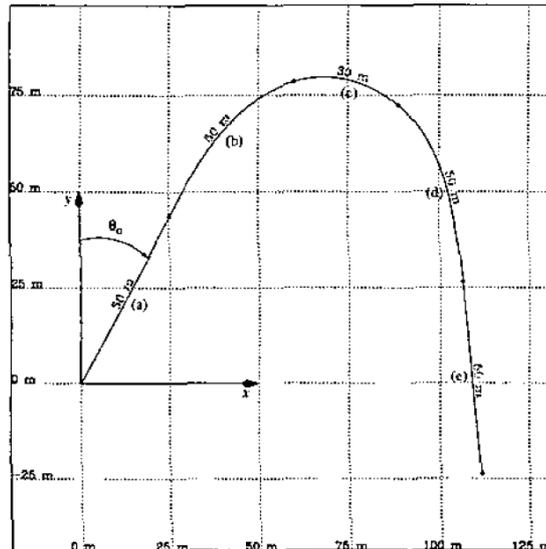


Figura 60. Ejemplo de curva de una carretera.

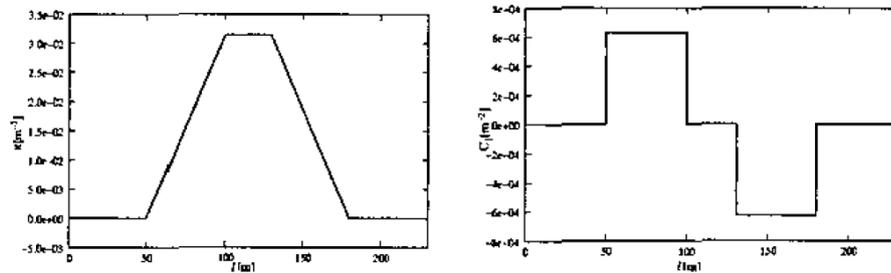


Figura 61. Perfiles de curvatura (izquierda) y sharpness (derecha) de la trayectoria presentada en la Figura 60.

La estructura del algoritmo de estimación de los parámetros del camino se puede ver en la Figura 62. Para eliminar el error se consideran tanto el módulo de estimación del movimiento propio del vehículo (*Ego-motion*) como el de procesamiento de imagen (IP).

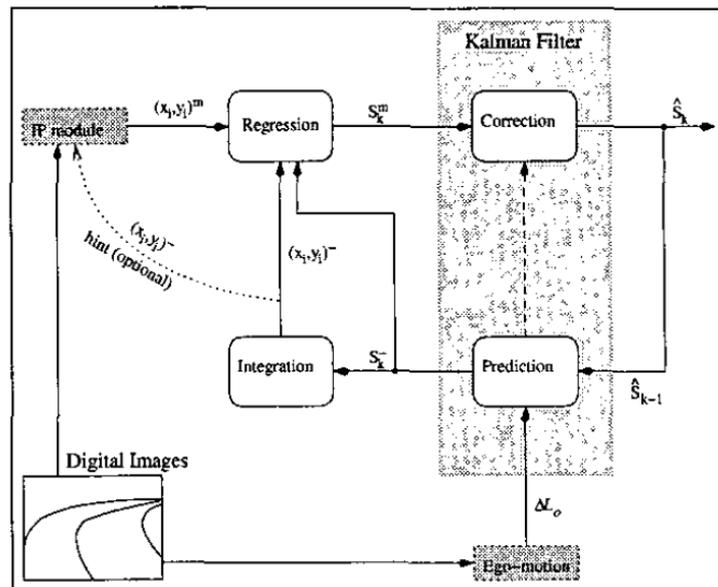


Figura 62. Estructura del algoritmo de estimación de los parámetros del camino.

Hay que tener en cuenta que  $\kappa$  es el índice discreto temporal,  $S$  es el modelo de estado de la carretera y  $(x_i, y_i)$  es el centro de la carretera.

En la Figura 63 aparecen los resultados del proceso de estimación, en los que se observan diferencias entre la Clotoide medida y la estimada.

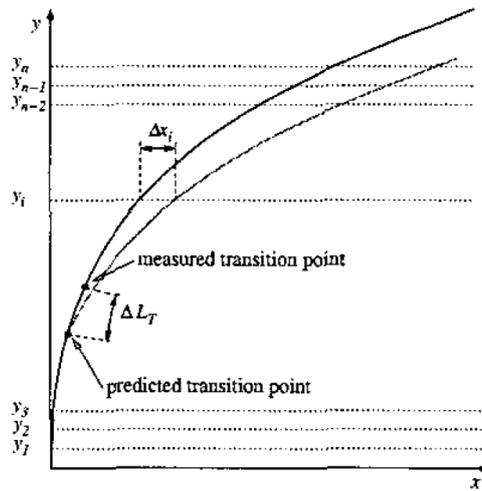


Figura 63. Clotoide estimada y medida.

En [80] se realiza una revisión y comparación de algunas técnicas utilizadas para realizar el control lateral de los vehículos cuando se desplazan por una carretera. Dichos reguladores son:  $H_\infty$ , Adaptivo, PID y Fuzzy. Se utiliza un circuito de test para compararlos mediante simulaciones en las que se incluyen variables como la curvatura, las perturbaciones del viento, las variaciones de la velocidad, el coeficiente de adherencia del vehículo, etc. Los resultados de las simulaciones se observan en la Figura 64.

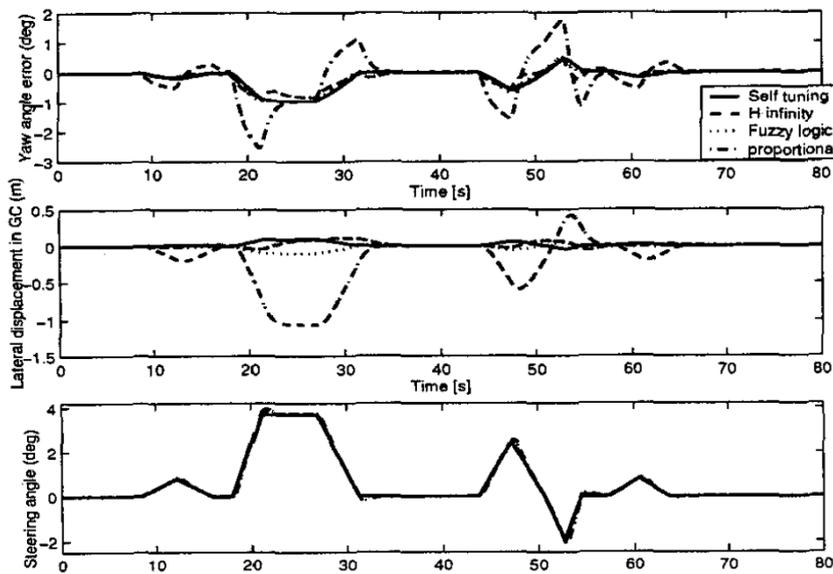


Figura 64. Resultados de los distintos reguladores utilizados.

Una de las novedades que aporta [81] es el hecho de que se centra en la generación del movimiento de un vehículo en el seguimiento de caminos, teniendo en cuenta restricciones cinemáticas y dinámicas, así como limitaciones de los motores. Como ya se sabe, la dinámica de los AGVs se describe mediante un modelo no lineal no-holonómico con dos entradas: un eje que genera el par para el movimiento lineal y otro para la orientación del vehículo. Considerar las restricciones cinemáticas más comunes puede que no siempre sea suficiente para generar

trayectorias factibles, por eso también hay que considerar las limitaciones de velocidad y de aceleración de los motores. El artículo propone un método de generación de trayectorias suaves de velocidad óptima para el AGV a partir de una ruta dada cuya curvatura es conocida, considerando las restricciones tanto cinemáticas como dinámicas. Se propone una estrategia de control no lineal, basada en una aproximación de Lyapunov, de modo que se garantice la estabilidad, aun cuando no sea correcto el valor de los parámetros estimados del modelo que describe el movimiento del vehículo.

A diferencia de las técnicas anteriores, en [82] se considera el control de seguimiento de caminos no desde una perspectiva genérica sino desde el punto de vista de una plataforma robótica específica. El vehículo utilizado para la experimentación es un robot con cuatro ruedas que pueden actuar tanto en el movimiento de tracción como en el de orientación, tal y como se ve en la Figura 65.



Figura 65. Vehículo utilizado en la experimentación.

El robot utiliza un sistema de posicionamiento global en tiempo real para cinemática diferencial, de modo que fácilmente se determina tanto la posición como la orientación relativa a la trayectoria que se desea seguir. En la Figura 66 aparece la ruta real trazada por el vehículo siguiendo una referencia de tipo escalón unitario.

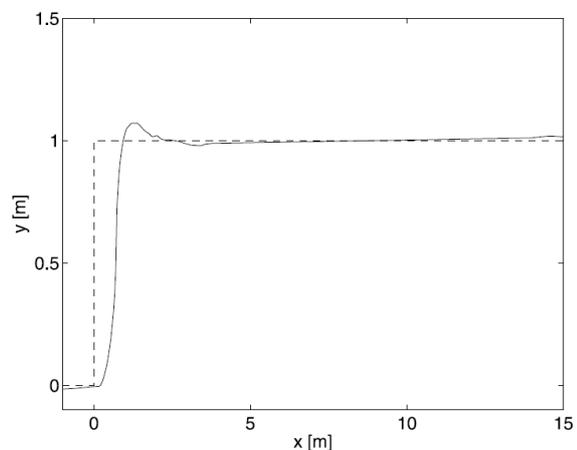


Figura 66. Ruta real dibujada en el pavimento mediante un sistema de pintado montado sobre el robot (izquierda). Trayectoria de referencia y trayectoria seguida por el robot (derecha), con control proporcional  $K_c = -1$  y velocidad  $v = 0.5 \text{ m/s}$  (- - escalón unitario (ruta deseada), — ruta resultante).

### 2.3.3. Detección de Carretera y Obstáculos

Hasta el momento se han explicado algunos trabajos de investigación que fundamentalmente se centran en resolver únicamente el problema de detección del camino y/o control del vehículo, pero también se ha trabajado la detección de obstáculos. Por ejemplo, en [83], al igual que en el trabajo de [71], se utiliza el IPM para facilitar las labores de procesamiento e identificación de la carretera (ver Figura 67).



Figura 67. Reconocimiento de carretera mediante la técnica IPM para deshacer la perspectiva.

[83] se engloba dentro de un proyecto llamado ARGO, que se centra en desarrollar un sistema de seguridad activa que sirva como piloto automático para vehículos de carreteras. Se describe el estado de la implementación del sistema de control basado en un planificador de ganancia, el cual permite al vehículo realizar el seguimiento de la carretera, aun cuando hay otros vehículos y obstáculos o la visibilidad no es buena (ver Figura 68).

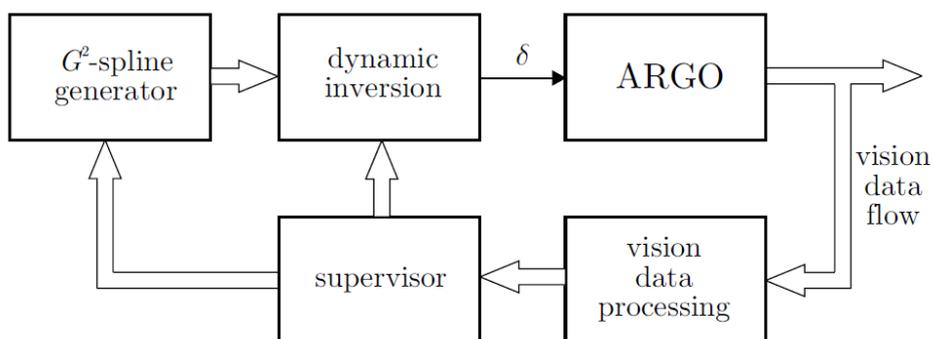


Figura 68. Esquema del sistema recursivo de control de trayectoria.

El sistema de visión estereoscópico que incorpora el vehículo permite la percepción del entorno mediante el procesamiento de las imágenes adquiridas. Se trata de un sistema de dos cámaras de bajo coste sincronizadas, las cuales adquieren simultáneamente pares de imágenes en niveles de gris.

Después de la adquisición hay varias etapas de procesamiento para detectar las marcas de las líneas, así como la detección de los obstáculos. Tras el re-mapeo de la imagen las marcas de la carretera pasan a ser líneas semi-verticales y tienen una anchura constante, además de un mayor brillo con respecto al fondo de la carretera. Por eso se pueden estimar los bordes de la carretera a través del análisis de la distancia de separación de las líneas que delimitan un carril.

El procesamiento de bajo nivel de la imagen se basa en la detección de las transiciones horizontales de color negro-blanco-negro y la posterior reconstrucción de la geometría de la carretera. El modelado matemático de la misma se realiza mediante un proceso iterativo de minimización de la distancia de los puntos a una poli-línea.

A continuación se realiza un procesamiento de alto nivel que se divide en: filtrar el ruido de las distintas características detectadas, seleccionar aquella que mejor se ajusta a las marcas detectadas del carril y unir varios segmentos con el objetivo de rellenar posibles discontinuidades producidas por el trazado discontinuo de las líneas, por estar decoloradas las líneas o por posibles oclusiones de objetos que aparezcan en la escena de forma no deseada. Dicha información sirve para una representación y reconstrucción mejor de la escena, la cual en un momento determinado puede perderse por cualquier situación inesperada. Por lo tanto, en dichas situaciones el procesamiento en un instante servirá como referencia en los siguientes periodos de control.

Como ya se ha comentado, este artículo, aporta un método que no sólo permite detectar carreteras mediante estereovisión, sino que también es capaz de identificar otros vehículos, con tal de aumentar el nivel de seguridad del sistema (ver Figura 69). El método utilizado se basa en la detección de simetrías para delimitar, mediante un *bounding box*, el objeto al cual se le está realizando el seguimiento o *tracking*.

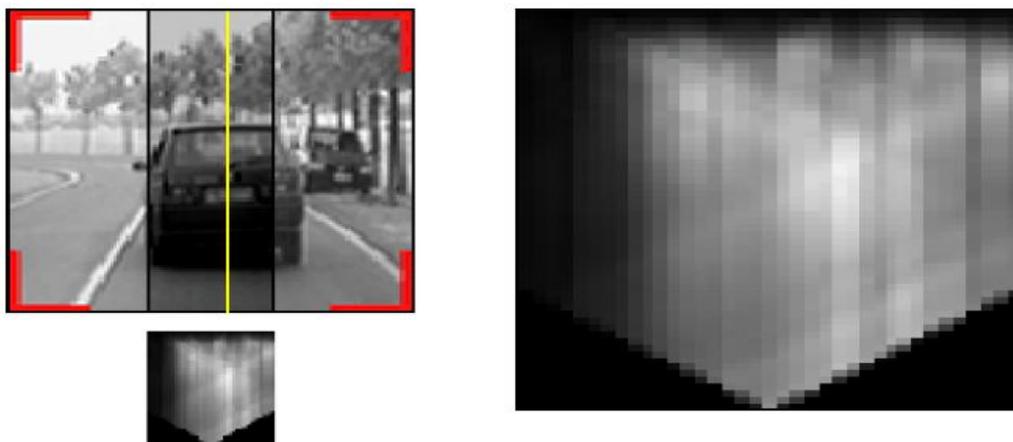


Figura 69. Detección de obstáculos mediante técnicas de simetría.

También se detallan tanto la actual como las futuras técnicas de control a implementar, las cuales están en una etapa de diseño y simulación. En la Figura 70 aparece el modelo junto con el esquema geométrico del problema que se desea resolver.

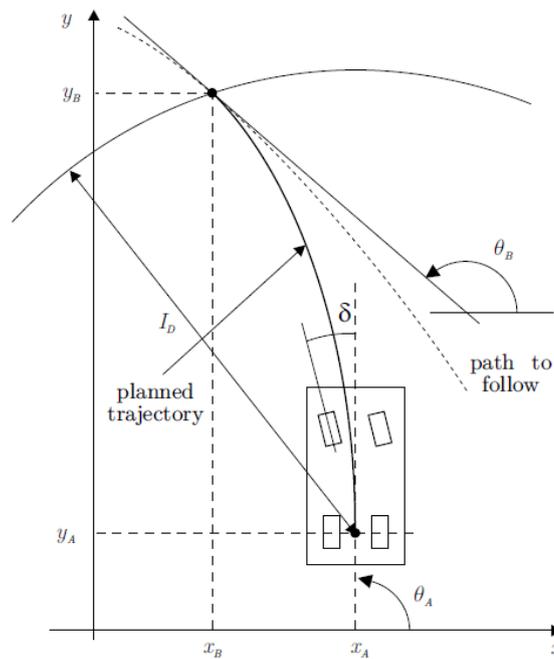


Figura 70. Planificación basada en visión utilizando  $G^2$ -spline.

El principal reto que plantea [84] es utilizar un sistema monocular de visión para el reconocimiento de carreteras, vehículos y otros obstáculos. Para ello en primer lugar se localizan en la imagen los componentes básicos de la carretera, tal y como se ve en la Figura 71.



Figura 71. Ejemplo de reconocimiento de las líneas de un carril de carretera.

Después se combinan las distintas imágenes con un clasificador basado en el sistema SVM (*Support Vector Machines*). Se propone un sistema distribuido de aprendizaje o *learning*, que se simplifica mediante la descomposición de las regiones de interés en tres subregiones que pueden aprenderse sencillamente de forma individual. El objetivo de dicho sistema es mejorar la identificación frente a situaciones como la variabilidad de colores y formas de los vehículos, las condiciones de iluminación, las oclusiones parciales o las rotaciones de los objetos. No obstante es un sistema que requiere mejoras para evitar falsos positivos o negativos y poderse así aplicar con total garantía de robustez en sistemas de asistencia a la conducción para vehículos. En la Figura 72 se muestran algunos de los resultados obtenidos por este sistema de detección de carreteras y vehículos.

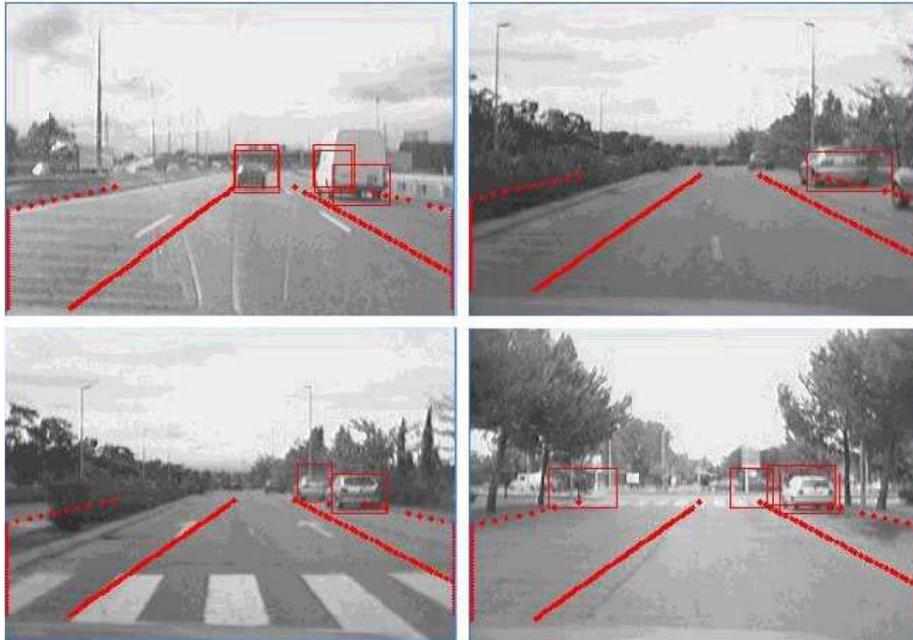


Figura 72. Ejemplos de reconocimiento de carril y detección de obstáculos.

Hasta el momento se han explicado algunas aportaciones que utilizaban sistemas de visión monocular, no obstante, pese a la dificultad que supone utilizar sistemas de estereovisión son muchas más las ventajas y por ello algunos investigadores optan por utilizar sistemas con cámaras estereoscópicas. Por ejemplo en [85] se presenta un método 3D de detección de carriles de carretera basado en estereovisión. Dicho algoritmo permite no hacer simplificaciones y considerar aspectos básicos que otros algoritmos no consideran con el objetivo de simplificar el problema, como por ejemplo: tener en cuenta la altura de la carretera, no fijar como constante el ángulo *pitch* o considerar el ángulo *roll*. Además, al trabajar con información 3D se puede calcular la separación entre la carretera y lo que se puede considerar como obstáculo. El carril se modela como una superficie 3D, definida mediante curvas Clotoides horizontales y verticales, así como su anchura y el ángulo *roll* de inclinación (ver Figura 73). En el proceso de estimación de los parámetros actuales de la línea se utilizan predicciones, mediante filtro de Kalman, basadas en la información de los parámetros

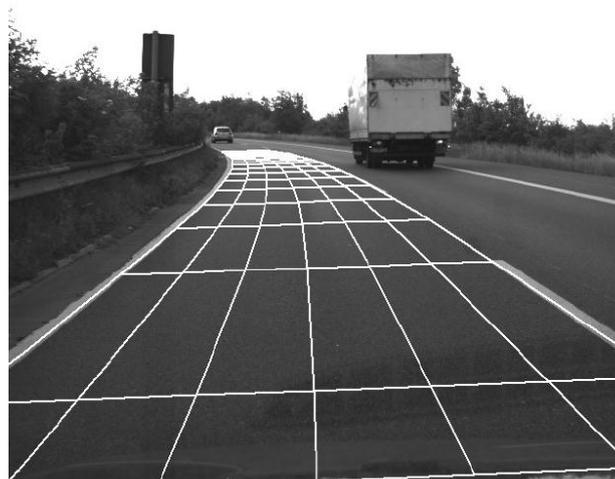


Figura 73. Detección de carril, que se modela como una superficie 3D, definida mediante Clotoides.

Posteriormente, en [86] y [87], los mismos autores presentan un sistema de detección de carriles que combina técnicas específicas de estereovisión con procesamiento de imágenes grises, maximizando la robustez y la aplicabilidad frente a las difíciles condiciones encontradas en las carreteras urbanas. Se extraen las características de las marcas de los carriles utilizando un sistema rápido y robusto de detección de las transiciones negro-blanco-negro. El modelo clotoidal se estima a partir de la extracción de características utilizando segmentos de línea redimensionados según ciertos intervalos de distancia, bajo restricciones especiales que aseguren que las modificaciones son correctas.

Además, para resolver las situaciones erróneas obtenidas con las técnicas más sencillas, se utiliza un método de detección de bordes independiente de las restricciones geométricas, basado únicamente en las características del marcado del carril. Finalmente los resultados de las distintas etapas de detección se combinan mediante fusión sensorial en un filtro de Kalman. Los resultados de las identificaciones de algunas situaciones especialmente difíciles se muestran en la Figura 74.



Figura 74. Ejemplos de detección de línea y modelado de carretera.

### 2.3.4. Detección de Múltiples Líneas

Hasta el momento se han explicado artículos y trabajos de visión que han abordado el problema de detección de una única línea para modelar la carretera, a pesar de que en la imagen aparecieran y se identificaran más líneas. No obstante, los investigadores también han trabajado la problemática de detección y estimación de múltiples líneas en la imagen, utilizando métodos muy diversos.

En [88] se introduce un sistema capaz de identificar y modelar un número arbitrario de carriles de carretera. Se obtiene información a través de visión, un sensor giroscópico y un mapa de datos, de modo que se combina y filtra toda la información para crear un modelo de carretera con un número determinado de líneas (ver Figura 75).



Figura 75. Detección de las líneas de los carriles de la carretera.

El modelo se basa en Clotoides y provee en todo momento las posiciones, ángulos y curvatura de los bordes de las líneas propias del carril en el que está el vehículo así como de los carriles o líneas cercanas. Se considera que la curvatura frente a la longitud del arco recorrido con el modelo de Clotoide es la siguiente:

$$c(l_c) = c_0 + c_1 \cdot l_c \quad (47)$$

Además, se tiene que el ángulo tangente  $\tau$  que describe el cambio de orientación se obtiene mediante integración sobre  $l_c$  de la siguiente manera:

$$\tau(l_c) = c_0 \cdot l_c + \frac{c_1 \cdot l_c^2}{2} = \frac{c_0 + c(l_c)}{2} \cdot l_c \quad (48)$$

En la Figura 76 aparecen distintos modelos clotoidales utilizados para la estimación del parámetro de una Clotoide determinada.

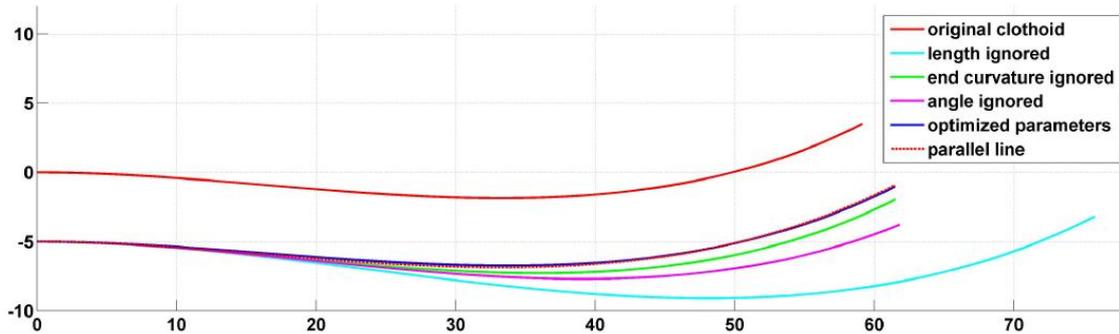


Figura 76. Estimación del modelo de una línea paralela a una Clotoide, con  $c_0 = -0.01$ ,  $c_1 = 0.0006$  y  $l = 60$ .

Como característica interesante se dirá que el sistema propuesto es capaz de seguir funcionando aun cuando el sistema de detección falla o la información recibida no es suficiente para estimar un modelo correcto.

[89] propone una solución mediante visión monocular para la detección simultánea de múltiples líneas en carreteras urbanas. A diferencia de otras aproximaciones, aquí se examina primero el rango de valores de los parámetros del carril que se necesita para un control continuo del vehículo, ya sea autónomo o manual. Se utiliza el conocimiento a priori para seleccionar, entre las distintas posibilidades, los parámetros finales de la carretera (posición y ángulo de orientación) en el plano de la imagen (ver Figura 77). Finalmente, con los resultados obtenidos se realiza el control local del movimiento del vehículo, con tal de que permanezca en su carril.

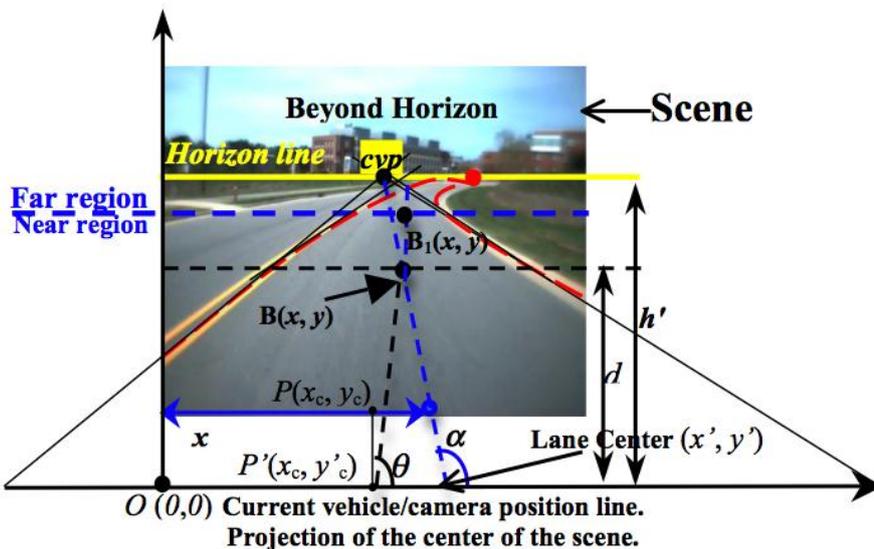


Figura 77. Explicación de la estrategia predictiva del movimiento en un escenario de carretera urbana.

Este sistema de detección y seguimiento de caminos se bautiza como *Parametric Transform for Multi-lane Detection*. El sistema es robusto frente a sombras y variaciones de color, de textura y de anchura de la carretera, además de estar diseñado tanto para líneas continuas como discontinuas, como se ve en la Figura 78.

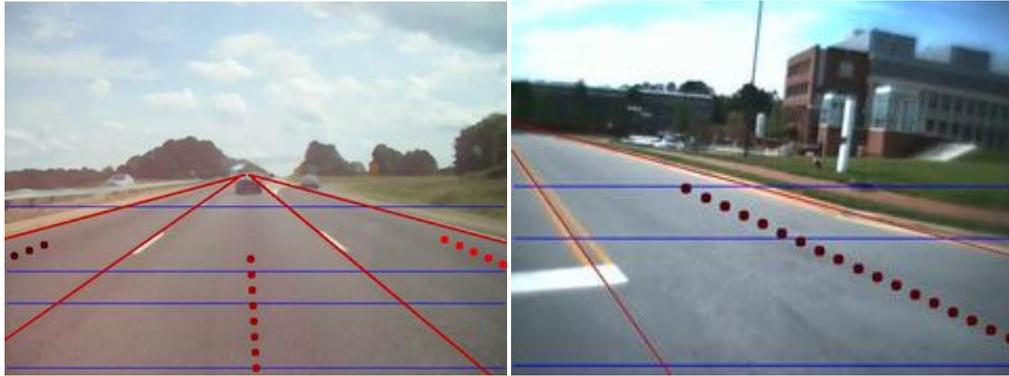


Figura 78. Resultados de la detección concurrente de múltiples líneas.

La detección y modelado de carriles de carreteras en la conducción autónoma se puede formular como un problema de estimación de curvas, donde los datos de los sensores locales proporcionan observaciones parciales y ruidosas de las curvas a nivel espacial. El número de curvas a estimar puede ser inicialmente desconocido y, por tanto, muchas observaciones pueden ser *outliers* o falsas detecciones debidas generalmente a las condiciones de iluminación. El reto es detectar líneas de carriles cuando y donde estén, de modo que sirvan para estimar correctamente la carretera.

En [90] se describe un algoritmo eficiente de estimación probabilística de líneas a partir de una representación novel de curvas. La clave es obtener curvas similares como variaciones de una curva básica simple. A partir de medidas locales de las líneas pintadas en la carretera y las características de la curva, se consigue realizar la fusión sensorial para detectar y estimar todos los carriles cercanos al vehículo. El sistema maneja carreteras con geometrías complejas y no realiza asunciones sobre la posición y orientación del vehículo con respecto a la carretera.

Los resultados de la experimentación demuestran la versatilidad del algoritmo, que es capaz de estimar de forma robusta los carriles frente a condiciones cambiantes de la carretera en entornos no conocidos (ver Figura 79).

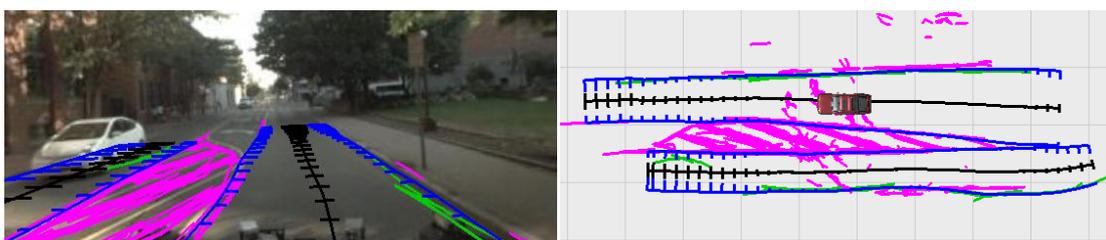


Figura 79. Ejemplo de identificación de la línea central de la carretera mediante el método probabilístico propuesto.

### 2.3.5. Sistemas Avanzados que Integran Detección de Caminos y Control del Vehículo

Gracias a los avances en los últimos años en las etapas de identificación y estimación de carreteras, así como en la fase de control, ha sido posible para muchos grupos de investigación generar sistemas completamente autónomos capaces de recorrer centenares de kilómetros por carreteras y caminos urbanos sin colisionar con ningún objeto.

Por ejemplo, en [91] se presenta un sistema de visión desarrollado para llevar a bordo del vehículo utilizado en el proyecto DIPLODOC. La principal tarea de este sistema es detectar situaciones de peligro con el objetivo de alertar al conductor.

En la Figura 80 se muestra la arquitectura general del sistema *on-board* de visión que está formado por los siguientes módulos: Active Contours (AC), Ego Motion (EM), Horizon line estimation (HZ), Inverse Perspective Mapping (IPM), Plane Fitting (PF), Road Tracking (RT), Stereo obstacle Outliner (SO), TextureLess region detection (TL) and V-Disparity mapping (VD).

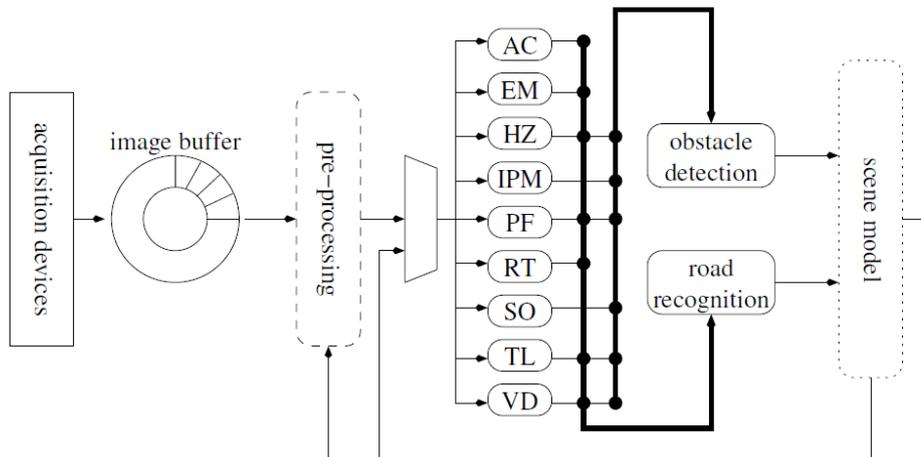


Figura 80. Arquitectura general del sistema *on-board* de visión DIPLODOC.

En la Tabla 8 se describe el estado de los módulos de detección de obstáculos y de reconocimiento de carretera, mientras que la Figura 81 muestra una de las fases de procesamiento para la identificación de la carretera.

<i>name</i>	<i>description</i>
AC	Active B-splines follow the roadway center by integrating edge information from both roadway sides [4, 5].
EM	It estimates the vehicle motion from optical flow and focus of expansion [6] by extracting and tracking salient features.
HZ	It filters temporally the position of the apparent horizon based on stereo analysis.
IPM	The anti-perspective transformations, <i>i.e.</i> the Inverse Perspective Map of both the left and right images are computed. By analyzing their difference [7, 8] this module estimates the free frontal space and the candidate obstacles position.
PF	It fits a plane on the disparity map by means of linear regression of disparity values. It works under the assumption of a locally flat road.
RT	It implements Kalman filtering with an innovative high-curvature road model specifically designed for high-curvature roads [9].
SO	It segments the regions exhibiting a constant disparity in the stereo disparity map. Thus, it generates candidates for obstacles.
TL	It extracts regions having certain radiometric characteristics in the color space and analyses their statistics signature. Textureless regions are typical of asphalt roads and large obstacles, like walls.
VD	It computes the v-disparity map, <i>i.e.</i> the histogram of disparity values for each image line. It allows the estimation of road geometry and obstacle distance [10, 11].

Tabla 8. Módulos del sistema de visión DIPLODOC.

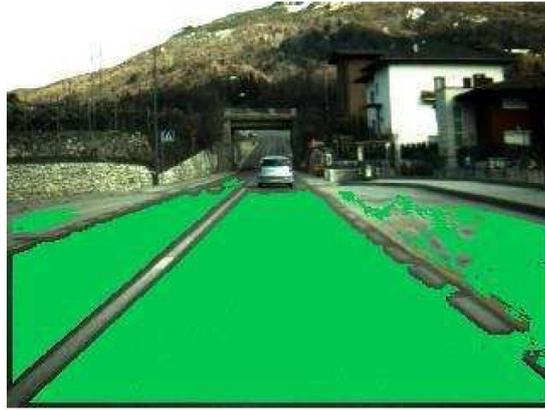


Figura 81. Resultados del reconocimiento de carretera del algoritmo.

En [92] se aplican varias técnicas estudiadas previamente que utilizan el filtro de Kalman extendido para la estimación de la geometría de caminos en el campo de la navegación autónoma de vehículos por carreteras. Al igual que en investigaciones previas se utiliza un modelo clotoidal para la geometría de la carretera (ver Figura 82), cuya línea central se estima recursivamente a partir de la extracción de características de las medidas tomadas por un sensor de rango de tipo LADAR de un único eje (ver Figura 83).

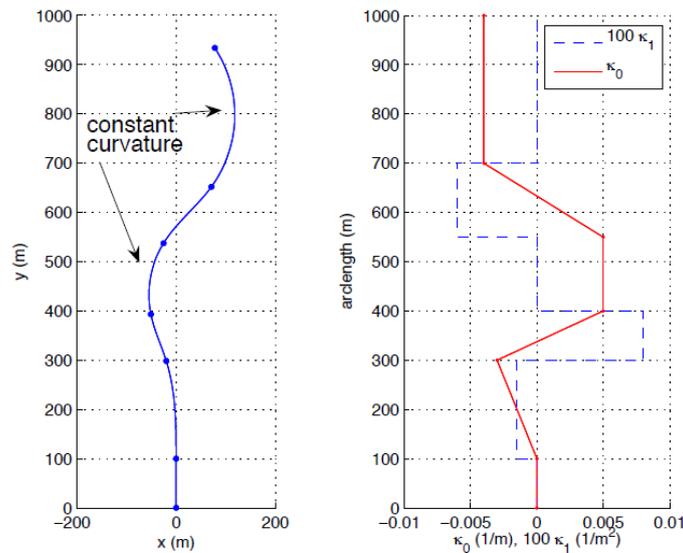


Figura 82. Geometría de la carretera (izquierda) y curvatura en función de la longitud del arco (derecha).



Figura 83. El vehículo utilizado en los experimentos es un modelo Ford E-350 Econoline, el cual lleva instalado un sistema LADAR para obtener los datos utilizados en la navegación.

[93] describe un sistema para detectar y estimar las características de múltiples carriles en redes de carreteras urbanas. La detección se realiza a partir de imágenes de vídeo y datos de un láser de rango, los cuales son adquiridos por un vehículo móvil. El vehículo es capaz de operar en varios escenarios ya que dispone de múltiples etapas de procesamiento: fusión sensorial de las marcas de la carretera detectadas por distintos métodos, detección de obstáculos y estimación estable de los carriles más cercanos mediante un modelo no paramétrico.

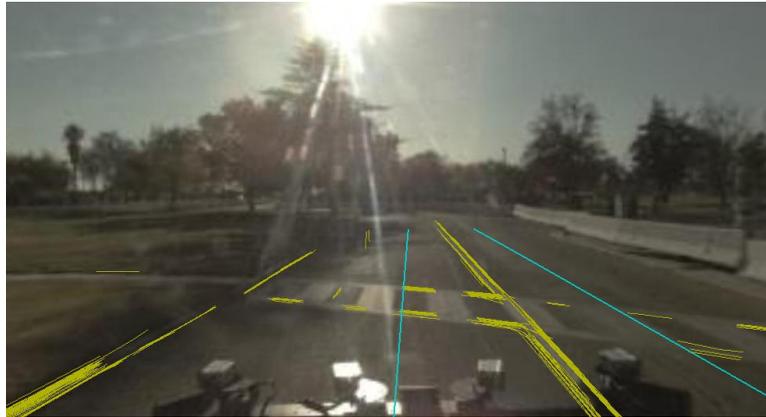


Figura 84. Para detectar el camino pintado y los bordes de la carretera el sistema utiliza varios sensores que aportan streams de datos asíncronos y heterogéneos (amarillo) y estimar la línea central de múltiples carriles (cian).

Las distintas etapas de procesamiento seguidas por el detector de líneas pintadas en la carretera propuesto en este artículo se observa en la Figura 85 y el resultado final de otro caso de complejidad similar en la Figura 84.

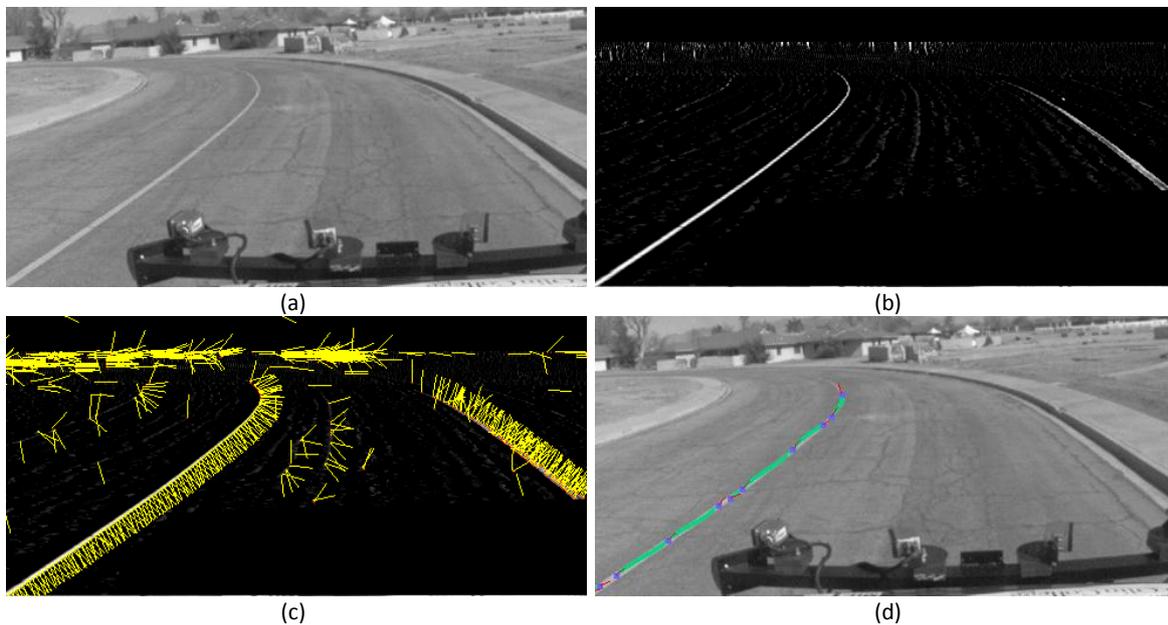


Figura 85. Detector de las líneas pintadas en la carretera: (a) Imagen original. (b) Imagen segmentada por filas mediante un filtro horizontal. (c) Imagen con los máximos locales de la respuesta del filtro y sus orientaciones, en amarillo se observan las orientaciones perpendiculares a cada máximo. (d) Los máximos cercanos se conectan mediante *splines* cúbicas de Hermite.

Hay que destacar que el vehículo incorpora información a priori de la red de carreteras modelada a modo de un conjunto de elementos lineales interconectados, los cuales sirven como referencia en el movimiento. En la Figura 86 se observa una vista aérea de los caminos recorridos de forma autónoma por el vehículo en la carrera *DARPA Urban Challenge* del año 2007.

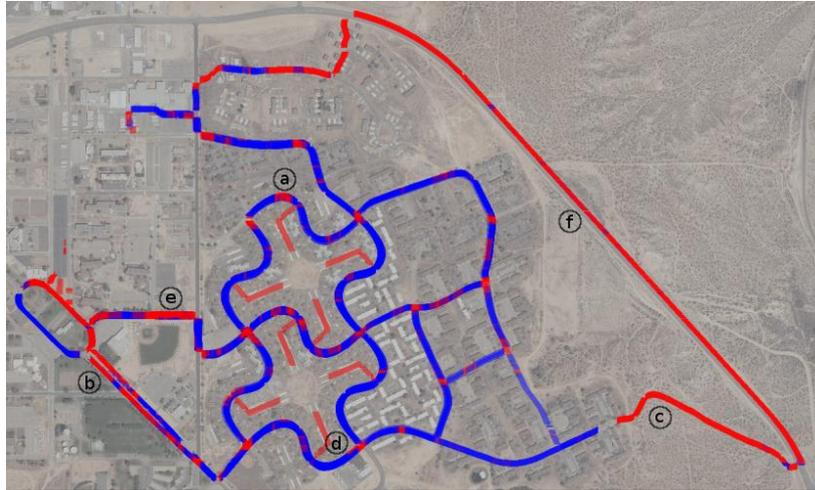


Figura 86. Vista aérea de los caminos recorridos de forma autónoma por el vehículo. Regiones en las que el seguimiento es preciso, robusto y seguro (azul). Zonas en las que la seguridad es baja (rojo). Algunas de las zonas de seguridad baja corresponden a intersecciones.

En [94] se describe una aproximación al seguimiento autónomo de carreteras sin líneas delimitadoras, es decir, por caminos de tierra. El algoritmo es capaz de reconocer curvas de caminos en imágenes de color no estructuradas, que normalmente aparecen en escenarios no urbanos (ver Figura 87). Para que sea compatible con curvaturas altas se modela el camino mediante dos segmentos de Clotoide diferentes.

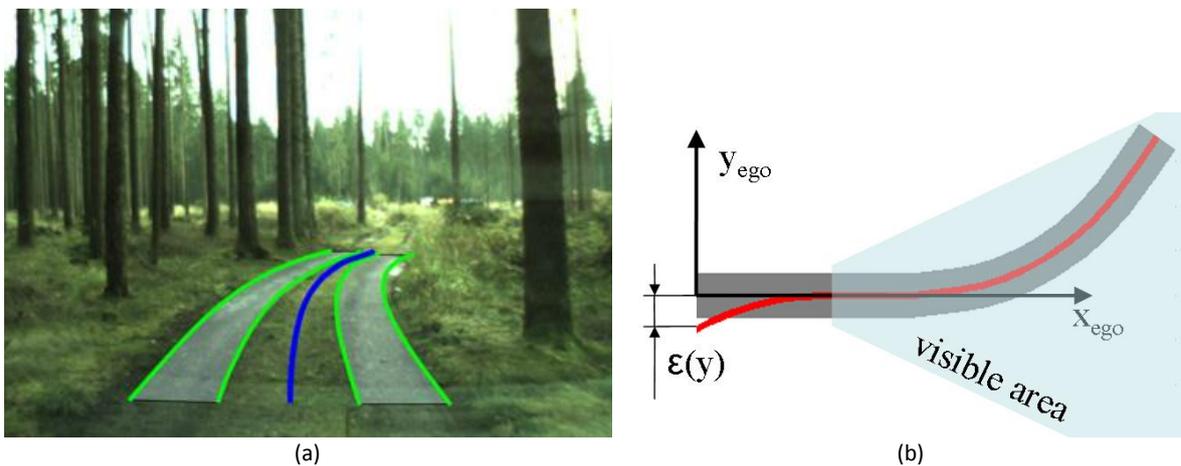
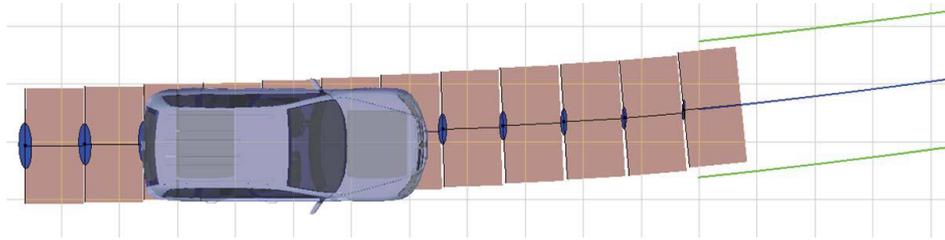


Figura 87. (a) Imagen típica de color de un camino sucio. (b) Error de desplazamiento  $\epsilon(y)$  del centro del vehículo cuando el seguimiento se realiza con un único segmento de Clotoide (línea roja) en una carretera (gris) a partir del área de visión de la cámara (cian).

Para poder detectar y seguir simultáneamente la carretera se utiliza un filtro de partículas que incorpora información de los bordes y de intensidad de color. Además, en el modelo de estimación se considera el propio movimiento del vehículo (ver Figura 88).



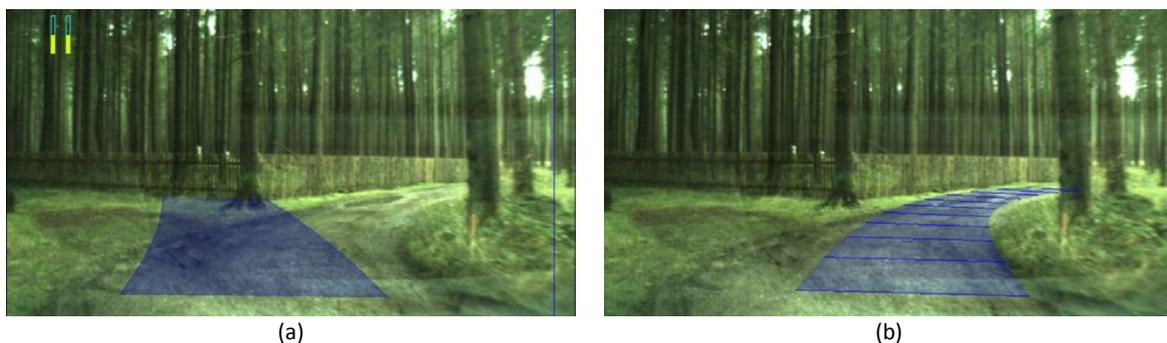
**Figura 88.** Generación sucesiva de segmentos estáticos de carretera (áreas marrones), con la predicción de las covarianzas del error (elipses azules) y la estimación de los segmentos locales de Clotoide (línea gris).

El proceso de reconocimiento y modelado de la carretera consta de las siguientes etapas:

- Modelado de la carretera.
- Modelado de la dinámica del vehículo.
- Medida del modelo mediante visión.
  - Saturación.
  - Bordes con intensidad de color.
  - Bordes con dirección de color (gradiente).
  - Anchura heurística de la carretera.

Además, el filtro de partículas se utiliza para generar segmentos estáticos de estimación de la carretera, para una distancia objetivo determinada. Dichos estimadores se predicen con respecto al movimiento propio del vehículo y se fusionan utilizando la técnica del filtro de Kalman para generar segmentos locales de Clotoide para el control lateral del vehículo.

En la Figura 89 aparecen para la misma imagen dos identificaciones distintas, una correcta, que sólo utiliza el filtro EKF en la identificación, y otra incorrecta, que además considera el filtro de partículas.



**Figura 89.** (a) Error en la detección de la carretera mediante el uso del EKF. (b) Detección correcta de la carretera tras la utilización del filtro de partículas propuesto.

### 2.3.6. Conclusión

Como conclusión se dirá que en la bibliografía estudiada se observa el relevante interés que tiene en la comunidad robótica la resolución del problema de seguimiento de líneas o caminos. Se ha demostrado que durante las últimas tres décadas ha sido un tema ampliamente estudiado y trabajado, desde la detección y estimación de un modelo matemático para representar el camino a seguir por el vehículo, hasta el algoritmo utilizado para el control del robot, o bien la solución final a nivel de implementación del sistema autónomo, mediante la integración de la etapa de visión, la de planificación de la trayectoria y la de control.

Además de los trabajos expuestos hay infinidad de estudios realizados con el objeto de agrupar y comparar las técnicas existentes para resolver el problema del seguimiento de líneas. Algunos ejemplos interesantes que pueden servir a modo de *survey* son los siguientes [95], [96], [97], [98], [99].



# **CAPÍTULO 3**

## **Generación de Trayectorias de Curvatura Continua**



## 3. Generación de Trayectorias de Curvatura Continua

### 3.1. Introducción

Como ya se ha definido en el Capítulo 1, el objetivo de la presente Tesina es la generación de trayectorias de curvatura continua para el problema del seguimiento de líneas con curvatura y brusquedad acotadas. No obstante es importante entender previamente los conceptos que rodean al propio objetivo, como son la generación de trayectorias y el control cinemático de robots móviles.

Además, antes de presentar los desarrollos más importantes del presente trabajo de investigación se explicará la evolución lógica seguida por el autor para llegar hasta la solución final propuesta. Para ello se introducirá inicialmente el concepto de control cinemático para seguimiento de trayectorias y caminos, explicando las limitaciones de los métodos existentes. A continuación, se detallarán las técnicas más básicas de generación de trayectorias continuas, explicando algunas de sus principales propiedades. Finalmente se desarrollará la solución propuesta, que resuelve el problema del seguimiento de trayectorias incluyendo restricciones cinemáticas y dinámicas tanto en la generación como en el control.

### 3.2. Control Cinemático

El control cinemático de robots móviles consiste en determinar las acciones del robot necesarias para llevarlo desde su posición actual a una posición final deseada, considerando las velocidades y la orientación. De forma genérica, según la distribución de sus ruedas y en cuáles de ellas se disponen los motores o actuadores encargados del movimiento se pueden obtener distintos modelos cinemáticos de robots [100].

Habitualmente se suelen utilizar dos tipos de controladores que tienen en cuenta la propia cinemática del robot (ver Anexo 2):

- Aquellos que utilizan la cinemática inversa del robot, normalmente orientados para el seguimiento de trayectorias [101].
- Aquellos que proporcionan directamente las acciones de rueda, normalmente orientados al seguimiento de caminos [102].

A continuación conviene determinar cuáles son las definiciones prácticas de los dos conceptos que se acaban de introducir:

- **Trayectoria:** Es una curva temporal para cada una de las coordenadas sobre las que se debe realizar un control del robot.

$$x(t), y(t), \theta(t) \quad (49)$$

- **Camino:** Es una curva definida en el espacio Cartesiano o de configuración que debe de seguir el robot. No se tiene en cuenta el factor tiempo.

$$C = f(x, y, \theta) \tag{50}$$

### 3.2.1. Seguimiento de Trayectorias

Después de haber realizado las definiciones anteriores es conveniente especificar cuál es la arquitectura de control típica en el seguimiento de trayectorias. El diagrama de bloques del bucle de control se puede ver en la Figura 90.

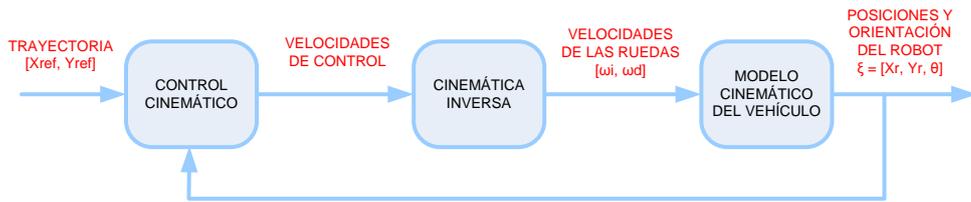


Figura 90. Bucle de control para seguimiento de trayectorias.

La estrategia que con mayor diferencia se aplica en el control cinemático de robots móviles para el seguimiento de trayectorias es la ley de control proporcional con pre-alimentación de velocidad. Dicha estrategia de control sólo tiene sentido en la configuración diferencial con punto descentrado. Las ecuaciones del modelo son las siguientes:

$$\begin{bmatrix} x'_c \\ y'_c \end{bmatrix} = \begin{bmatrix} x'_{ref} \\ y'_{ref} \end{bmatrix} + \begin{bmatrix} K_x & 0 \\ 0 & K_y \end{bmatrix} \cdot \begin{bmatrix} x_{ref} - (x + e \cdot \cos\theta) \\ y_{ref} - (y + e \cdot \sin\theta) \end{bmatrix} \tag{51}$$

$$\begin{bmatrix} v_i \\ v_d \end{bmatrix} = \frac{1}{e} \cdot \begin{bmatrix} e \cdot \cos\theta + b \cdot \sin\theta & e \cdot \sin\theta - b \cdot \cos\theta \\ e \cdot \cos\theta - b \cdot \sin\theta & e \cdot \sin\theta + b \cdot \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x'_c \\ y'_c \end{bmatrix} \tag{52}$$

En la Figura 91 se muestran los resultados de una simulación del movimiento de seguimiento de trayectorias para la configuración diferencial con punto descentralizado.

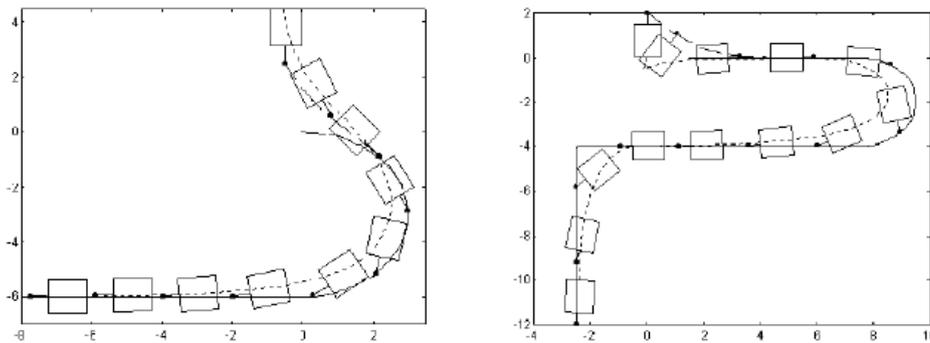


Figura 91. Simulación de control de seguimiento de trayectorias.

### 3.2.2. Seguimiento de Caminos

Por seguimiento de caminos se entiende una ley de control que permita seguir un camino prefijado o dinámico. La principal diferencia es que no se tiene en cuenta la variable de tiempo y por tanto, según la posición real del robot, se busca en todo momento un punto objetivo del camino. Así mismo, la velocidad de seguimiento del camino es un parámetro que hace que el robot avance más o menos, pudiendo estar incluida en la propia ley de control.

Con un control de trayectoria, si aparece un obstáculo el robot colisionará si no se hace nada al respecto. Para solucionarlo hay que re-planificar la trayectoria, lo cual puede ser bastante costoso. En cambio, con un seguimiento de caminos podemos disminuir o aumentar fácilmente la velocidad de seguimiento a fin de adelantar o dejar pasar al obstáculo. La principal dificultad reside en definir caminos de forma paramétrica y calcular un punto objetivo (en general suelen utilizarse relaciones geométricas).

En este apartado se explicará la estrategia que más se utiliza, el *Pure Pursuit* o persecución pura. Se trata de un método que determina geoméricamente la curvatura del vehículo a un punto determinado del camino, denominado punto objetivo. Dicho punto se sitúa sobre el camino y se encuentra a cierta distancia  $LA$  (*Look-Ahead*) de la posición actual del vehículo. El objetivo es calcular la curvatura del arco que une el origen del sistema con el punto objetivo ( $x_{ob}, y_{ob}$ ) y cuya longitud de cuerda es  $LA$ .

Para plantear el método de persecución pura se utiliza un sistema de referencia local asociado al vehículo móvil, como se muestra en la Figura 92. Se supone que en el intervalo de control  $T$ , la curvatura es constante, describiendo el vehículo un arco de circunferencia.

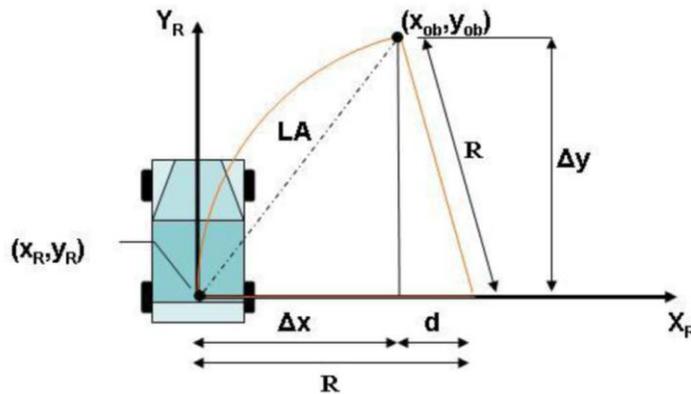


Figura 92. Modelo para el control de persecución pura.

Dónde:

- $R$ : radio de curvatura o giro
- $\gamma_r$ : curvatura
- $LA$ : distancia a la que se encuentra el punto objetivo del camino a seguir
- $\Delta x$ : desplazamiento lateral

Por lo tanto el robot está separado una distancia  $(\Delta x, \Delta y)$  del punto objetivo. El radio de curvatura para alcanzar el punto objetivo es  $R$ . Además la velocidad  $v$  con la que queremos seguir el camino la suponemos constante.

A continuación se va a realizar una pequeña explicación de la técnica de persecución pura de cara a la implementación práctica, para ello tomaremos como referencia la Figura 93.

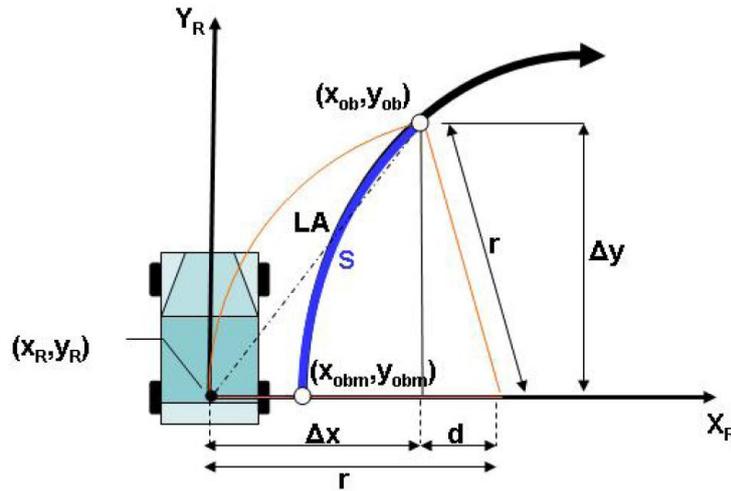


Figura 93. Esquema para calcular la curvatura de la trayectoria con persecución pura.

El algoritmo para la aplicación de esta ley de control es muy sencillo, ya que como ya se ha dicho anteriormente, basta con determinar el punto del camino que se encuentra a una distancia previamente definida ( $LA$ ) y calcular el error lateral ( $\Delta x$ ) con respecto a la posición actual del vehículo. También es necesario tener en cuenta la orientación del vehículo para obtener  $\Delta x$ , dado que en un sistema de coordenadas global se tendría lo siguiente:

$$\Delta x = (x_{ob} - x_R) \cdot \cos\theta + (y_{ob} - y_R) \cdot \sin\theta \quad (53)$$

$$LA = \sqrt{(x_{ob} - x)^2 + (y_{ob} - y)^2} \quad (54)$$

De la Figura 93 se derivan las siguientes expresiones:

$$(\Delta x)^2 + (\Delta y)^2 = LA^2 \quad (55)$$

$$\Delta x + d = R \quad (56)$$

El radio de curvatura es aquel con el que gira el robot para una velocidad y ángulo de rueda determinados y constantes. Las siguientes ecuaciones relacionan la curvatura del arco con la distancia  $LA$ .

$$d = R - \Delta x \quad (57)$$

$$(R - \Delta x)^2 + (\Delta y)^2 = R^2 \quad (58)$$

Si hacemos el desarrollo y despejamos la variable  $R$  de la expresión se obtiene lo siguiente:

$$R = \frac{(\Delta x)^2 + (\Delta y)^2}{2\Delta x} = \frac{LA^2}{2\Delta x} \quad (59)$$

Del análisis de la ecuación anterior se obtiene que la curvatura que es necesario suministrar al vehículo para alcanzar el punto objetivo sobre la trayectoria viene dada por:

$$\gamma_r = \frac{1}{R} = -\frac{2\Delta x}{LA^2} \quad (60)$$

Por lo tanto la ley de control de persecución pura se obtiene como un control proporcional al error lateral  $\Delta x$  con respecto al punto objetivo: Además se observa que la constante de proporcionalidad varía con la inversa del cuadrado de la distancia  $LA$ .

### 3.2.2.1. Seguimiento de Líneas

Con un sistema de visión, podemos calcular la distancia de separación del robot a una línea y también su inclinación, es decir la orientación de la línea con respecto al robot (ver Figura 94).

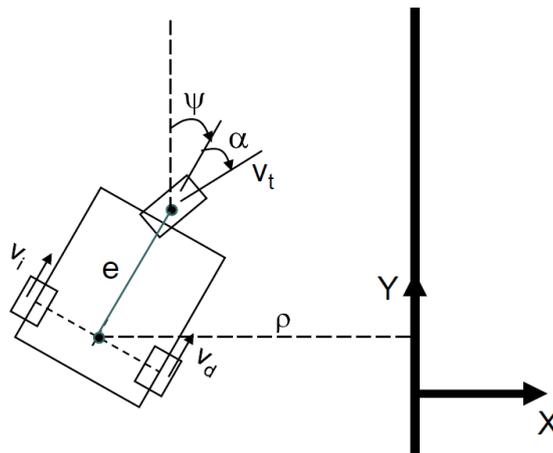


Figura 94. Modelo de seguimiento de línea

Se puede adaptar la ley de control anterior para que el sistema tome primero un punto perteneciente a la línea como referencia y después realice el seguimiento del mismo, de modo que al final el vehículo converja a la línea.

$$\alpha = -\psi + \arctan\left(\frac{2 \cdot e \cdot \rho}{L^2}\right) \quad (61)$$

$$v_t = v_{seguimiento} \quad (62)$$

### 3.2.3. Control con Reguladores Convencionales

#### 3.2.3.1. Regulador PID

Como ya se ha comentado existen técnicas específicas para realizar el control cinemático de un robot en cualquiera de sus configuraciones típicas. No obstante, el control de este tipo de sistemas dinámicos también se puede abordar desde un punto de vista genérico de la teoría de control.

La palabra control tiene diferentes significados en robótica:

- Asegurar que el movimiento del robot sea estable y que se produzca de acuerdo con ciertos criterios de control.
- Conseguir que la velocidad del motor/ruedas se mantenga lo más próximo posible a la consigna fijada (error en estado estacionario o tiempo de estabilización inferiores a cierto valor, etc.).
- Lograr que la orientación del vehículo sea la deseada.
- Hacer que el vehículo se mantenga horizontal pese a las irregularidades del terreno.

Dentro del área del control y la automatización, los reguladores PID (Proporcional-Integral-Derivativo) son ampliamente utilizados y conocidos, de hecho al PID se le considera el controlador estándar para aplicaciones industriales. Su ecuación genérica es la siguiente:

$$u(t) = K_p \left( e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (63)$$

Para realizar el diseño de un controlador PID es necesario disponer de un modelo matemático de la dinámica de la planta. En este caso se trata del modelo cinemático del robot móvil en cualquiera de sus configuraciones. En este sentido se podría intentar controlar el robot desde dos puntos de vista:

- Referencias de posición, velocidad y aceleración del robot (Figura 95).

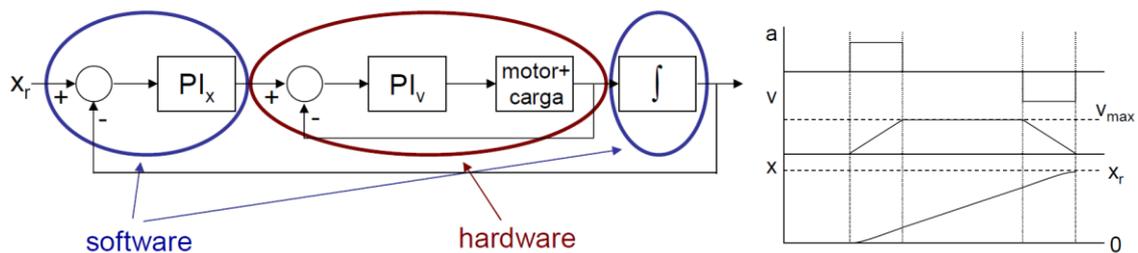


Figura 95. Esquema de control de posición con PID (izquierda) y referencias cinemáticas (derecha).

- Referencias de velocidad y orientación de cada una de las ruedas (Figura 96).

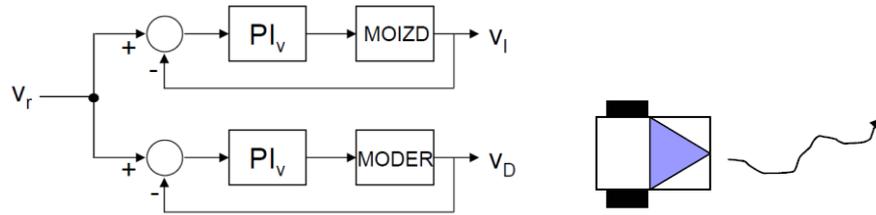


Figura 96. Esquema de control de ruedas con PID para un modelo diferencial de robot.

### 3.2.3.2. Regulador “Fuzzy” o Borroso

Por último hay que destacar la navegación basada en comportamiento *Fuzzy* o borroso, es decir, el comportamiento basado en reglas. De manera sencilla con un conjunto de reglas se pueden definir el comportamiento reactivo del robot (ver Tablas 9 (a) y 9 (b)).

$d \backslash \phi$	L	LC	C	RC	R
L	VL	VL	VL	VL	VL
M	VL	VL	M	VL	VL
H	VL	M	VH	M	VL

(a)

$d \backslash \phi$	L	LC	C	RC	R
L	P	P	Z	N	N
M	VP	P	Z	N	VN
H	VP	P	Z	N	VN

(b)

Tabla 9. (a) Reglas de Velocidad Lineal y (b) Reglas de Velocidad Angular.

Mediante la lectura directa de sensores (o alguna combinación sencilla de ellos), se establecen zonas borrosas en las que se encuentran las configuraciones objetivo en caso de seguimiento de líneas o los objetos detectados en caso de evitación de obstáculos. Según la proximidad y la orientación de dichos objetos, generan una acción u otra según el comportamiento deseado.

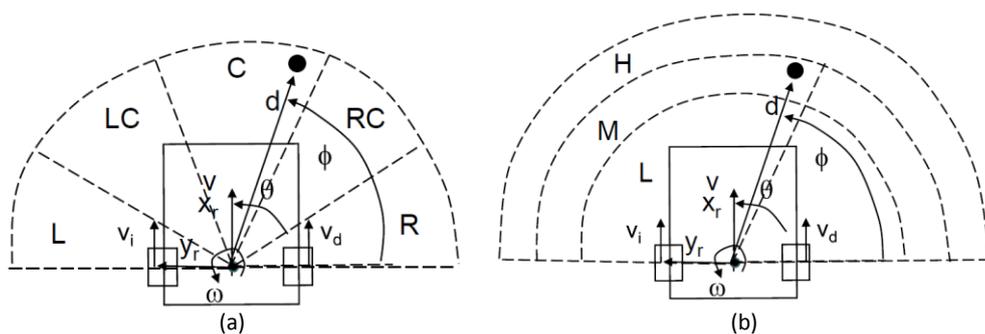


Figura 97. Ejemplo de evaluación de las reglas: (a) en función del ángulo o (b) en función de la distancia.

Pese a que las técnicas de control que acaban de exponerse se utilizan desde hace mucho, debido a su facilidad de implementación y su bajo coste computacional, ninguna de ellas es capaz de generar trayectorias de curvatura continua. Eso se debe al hecho de que en sí mismas no consideran el estado previo del sistema dinámico desde el punto de vista de la continuidad de la trayectoria generada y basan su funcionamiento en la teoría de control clásico.

Como conclusión se dirá que el objetivo común de dichas técnicas es definir configuraciones (generalmente libres de colisión) que sirven como referencia a leyes de control que en cada iteración calculan una nueva acción totalmente independiente de la anterior, puesto que se trata de expresiones matemáticas que sólo garantizan estabilidad y convergencia del sistema dinámico. Se considera la configuración del robot pero en ningún momento se tiene en cuenta la configuración de las ruedas del robot, es decir, no importa la trayectoria seguida por la posición y orientación de las mismas a lo largo del tiempo. Para ello es necesario analizar las herramientas matemáticas existentes para generar trayectorias de curvatura continua.

### 3.3. *RS-Path*

Las técnicas introducidas en el Anexo 1 sólo permiten determinar un conjunto de puntos intermedios o configuraciones libres de colisión, pero no el camino completo en sí mismo que une dos o más configuraciones. Esto significa que dichos algoritmos sólo son capaces de unir los distintos puntos mediante segmentos de recta, de modo que el camino para ir de una configuración libre de colisión a la siguiente no queda claramente definido. Existen muchas opciones para generar un camino que resuelva el problema, siendo el método más simple y rápido la concatenación de segmentos de recta y círculos (ver Figura 98), lo que se conoce como *RS-paths* (*Reeds and Shepp's paths*) [6].

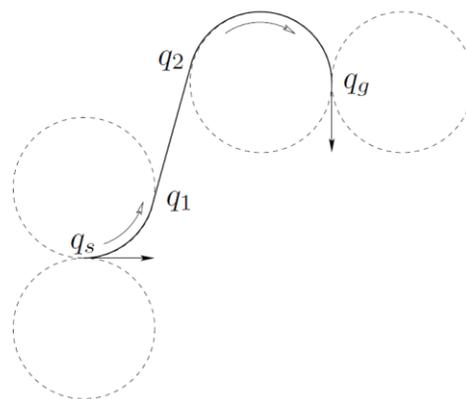


Figura 98. Ejemplo de *RS-path*.

Aunque la trayectoria generada mediante *RS-paths* es continua en su trazada en realidad es discontinua en su curvatura, ya que en el cambio de trayectoria recta a trayectoria circular y viceversa, existe una discontinuidad de la curvatura. Esto es cierto por el hecho de que se pasa de forma instantánea de una curvatura nula (recta con radio infinito) a otra de valor constante (arco circular con radio constante). Dicha discontinuidad impone zonas de aceleración infinitas, provocando que el robot no siga la trayectoria de forma correcta, lo cual puede resultar inaceptable tanto en el problema de seguimiento de trayectorias como en el de auto-localización. Es por ello que se requiere de algún tipo de curva de transición para pasar de una trayectoria a otra.

### 3.4. Clotoide

Los investigadores han estudiado diversas formas de generar trayectorias continuas, llegando a la conclusión de que para solucionar el problema de la discontinuidad de la curvatura lo ideal es la inclusión de Clotoides, las cuales pueden utilizarse de forma independiente o como tramos de transición entre rectas y arcos circulares.

La Clotoide (también llamada espiral de Cornu o espiral de Euler) es una curva plana en forma de espiral doble, con simetría central (ver Figura 99). Desde su origen  $O$  (con curvatura nula y radio infinito) su radio de curvatura va disminuyendo a lo largo que se avanza por las dos ramas, de manera que el producto entre el radio de curvatura y la distancia recorrida medida sobre la curva permanece constante. Así, las dos ramas de la Clotoide tienden a converger en los dos puntos impropios de la curva  $C$  y  $C'$  (con radio nulo), a los que se llegará tras recorrer sobre la curva una distancia infinita, después de haber hecho infinitas vueltas.

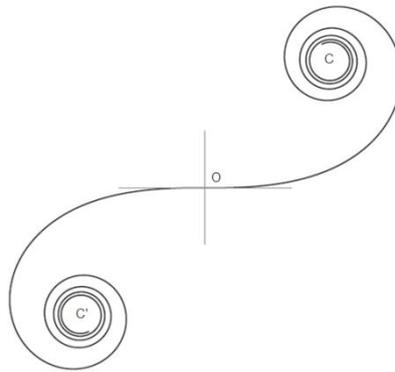


Figura 99. Ejemplo de curva Clotoide.

Directamente relacionada con la Clotoide está su inversa, la anti-Clotoide (ver Figura 100), que se define como la trayectoria que pasa de un tramo de curvatura de una valor determinado (tramo circular) a otro tramo de curvatura nula (tramo rectilíneo). De hecho, la anti-Clotoide es una curva Clotoide recorrida en sentido inverso, es decir, yendo desde un punto determinado  $P$  hasta su origen  $O$ .

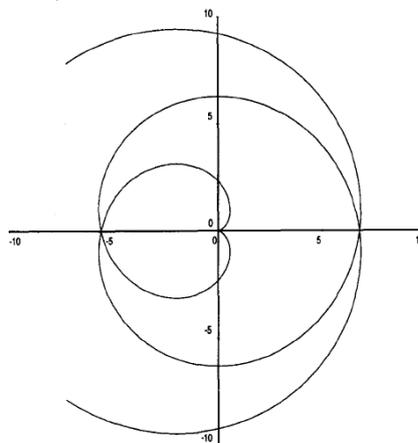


Figura 100. Ejemplo de curva anti-Clotoide.

### 3.4.1. Propiedades

La Clotoide es una curva definida paramétricamente en términos de las integrales de Fresnel como se muestra en la ecuación siguiente:

$$C(\gamma) = \begin{pmatrix} x(\gamma) \\ y(\gamma) \end{pmatrix} = K \cdot \begin{pmatrix} C_x(\gamma) \\ S_y(\gamma) \end{pmatrix} = K \cdot \begin{pmatrix} \int_0^\gamma \cos\left(\frac{\pi}{2} \cdot \xi^2\right) \cdot d\xi \\ \int_0^\gamma \sin\left(\frac{\pi}{2} \cdot \xi^2\right) \cdot d\xi \end{pmatrix} \quad (64)$$

Donde la constante de proporcionalidad  $K$  es un número real positivo y el límite de integración  $\gamma$  es un número real no-negativo.

La propiedad fundamental de la Clotoide es que guarda una relación inversamente proporcional entre la distancia recorrida  $L$  desde su origen  $O$  y el radio de curvatura  $\rho$  en cualquier punto  $P$  de la curva. La raíz cuadrada de la constante de proporcionalidad se llama parámetro de la Clotoide  $A$  y tiene dimensiones de longitud. Por definición, la ecuación intrínseca de la Clotoide es:

$$\rho \cdot L = A^2 \quad (65)$$

Dónde:

- $\rho$  es el radio de curvatura.
- $L$  es la longitud del arco.
- $A$  es el parámetro característico de la Clotoide.

También se puede definir la relación anterior diciendo que la longitud del arco es proporcional a la curvatura o inversamente proporcional al radio en cada punto de la curva.

$$\kappa = \frac{1}{\rho} = \frac{L}{A^2} \quad (66)$$

Aunque la anterior es la propiedad más atractiva, existen otras propiedades geométricas. Entre las propiedades más interesantes de las Clotoides destacan las siguientes:

- Ángulo de tangencia:

$$\tau = \pi \cdot \gamma^2 / 2 \quad (67)$$

- Curvatura:

$$\kappa = \pi \cdot \gamma / K \quad (68)$$

- Longitud del arco:

$$L = K \cdot \gamma = \sqrt{\pi} \cdot A \cdot \gamma \quad (69)$$

Donde  $A = K/\sqrt{\pi}$  es el parámetro constante que caracteriza la Clotoide.

- Factor de Homotecia:

$$K = \sqrt{\pi} \cdot A \quad (70)$$

De las expresiones anteriores se pueden deducir directamente algunas relaciones entre las variables y sus derivadas. Suponiendo que se tiene una velocidad constante  $v$ , el *sharpness* de la trayectoria es:

$$\sigma = A^{-2} \quad (71)$$

Puesto que si se deriva la curvatura respecto al tiempo, derivada de la ecuación (71) se obtiene:

$$\frac{d\kappa}{dt} = \frac{1}{A^2} \cdot \frac{dL}{dt} = \sigma \cdot v \quad (72)$$

Además, según [24] se pueden calcular analíticamente las derivadas de una curva, de modo que las componentes de la velocidad son:

$$\mathbf{v}(t) = \dot{\mathbf{R}} = v \cdot \mathbf{t} \quad (73)$$

Utilizando la relación existente entre dos vectores normales:

$$\dot{\mathbf{t}} = \left(\frac{v}{\rho}\right) \cdot \mathbf{n} \quad (74)$$

$$\dot{\mathbf{n}} = -\left(\frac{v}{\rho}\right) \cdot \mathbf{t} \quad (75)$$

Junto con la relación matemática de la derivada de un producto:

$$\dot{\mathbf{F}} = \frac{d(A \cdot B)}{dt} = \dot{A} \cdot B + A \cdot \dot{B} \quad (76)$$

Se obtiene que las componentes genéricas de la aceleración son las mostradas a continuación:

$$\mathbf{a}(t) = \dot{\mathbf{v}} = \dot{v} \cdot \mathbf{t} + \frac{v^2}{\rho} \cdot \mathbf{n} \quad (77)$$

Si se particulariza para el caso de una Clotoide con velocidad  $v$  constante, se obtiene:

$$\mathbf{a}(t) = \dot{\mathbf{v}} = 0 \cdot \mathbf{t} + v^2 \cdot \kappa \cdot \mathbf{n} \quad (78)$$

Finalmente, las componentes de la derivada de la aceleración o *jerk* para una curva genérica son:

$$\mathbf{j}(t) = \left(\ddot{v} - \frac{v^3}{\rho^2}\right) \cdot \mathbf{t} + \left(\frac{v}{\rho}\right) \left[3\dot{v} - \left(\frac{v}{\rho}\right)\dot{\rho}\right] \cdot \mathbf{n} \quad (79)$$

Si se particulariza para el caso de una Clotoide, se obtiene lo siguiente:

$$\mathbf{j}(t) = \left(0 - \frac{v^3}{\rho^2}\right) \cdot \mathbf{t} + \left[0 - \left(\frac{v}{\rho}\right)^2 \dot{\rho}\right] \cdot \mathbf{n} \quad (80)$$

Sabiendo que la relación matemática de la derivada de una división es:

$$\dot{F} = \frac{d(A/B)}{dt} = \frac{\dot{A} \cdot B - A \cdot \dot{B}}{B^2} \quad (81)$$

Se puede calcular la derivada del radio de curvatura:

$$\frac{1}{\rho^2} = \kappa^2 \quad (82)$$

Con dicha ecuación y a partir de la relación entre el radio y su curvatura:

$$\dot{\rho} = \left( \frac{1}{\kappa} \right) = -\frac{\dot{\kappa}}{\kappa^2} \quad (83)$$

Se desarrolla el segundo término, componente normal del *jerk*.

$$\mathbf{j}_n = -v^2 \cdot \kappa^2 \cdot \left( -\frac{\dot{\kappa}}{\kappa^2} \right) \cdot \mathbf{n} = v^2 \cdot \dot{\kappa} \cdot \mathbf{n} = v^2 \cdot (v \cdot \sigma) \cdot \mathbf{n} = v^3 \cdot \sigma \cdot \mathbf{n} \quad (84)$$

Finalmente, si se compacta más la ecuación (84) se obtiene:

$$\mathbf{j}(t) = -v^3 \cdot \kappa^2 \cdot \mathbf{t} + v^3 \cdot \sigma \cdot \mathbf{n} \quad (85)$$

La conclusión más interesante que se deduce del desarrollo anterior es que tanto la curvatura como su derivada (el *sharpness*) están directamente relacionadas con las componentes tangencial y normal de la aceleración y el *jerk*. Por lo tanto, queda demostrado que limitando la curvatura y su derivada se limitan también la aceleración y su derivada, lo cual es una propiedad muy interesante desde el punto de vista de la generación de trayectorias y el control cinemático en robótica móvil.

### 3.4.2. Métodos de Aproximación

Como se ha comentado en el Capítulo 2, el mayor inconveniente de este tipo de trayectorias es el alto coste computacional que conlleva trabajar con las coordenadas de las Clotoides de forma exacta, lo cual hace más difícil su inclusión en el planificador de trayectorias del robot, dados los requerimientos de tiempo real.

Se pueden aproximar las integrales de Fresnel mediante discretización:

$$C = \sum_{\gamma_s}^{\gamma_e} \cos\left(\frac{\pi}{2} \cdot \xi^2\right) \cdot \Delta\xi \quad (86)$$

$$S = \sum_{\gamma_s}^{\gamma_e} \sin\left(\frac{\pi}{2} \cdot \xi^2\right) \cdot \Delta\xi \quad (87)$$

Donde  $\gamma = \sqrt{\frac{2}{\pi}} \cdot \tau$  es la relación entre el parámetro de integración y el ángulo de tangencia de la trayectoria.

El problema es que como cualquier otro método de integración discreta, la precisión depende del periodo de muestreo y a su vez el coste computacional es inversamente proporcional respecto al tamaño de dicho periodo. De hecho, el tiempo de cómputo se incrementa cuando el periodo de muestreo se hace pequeño y en consecuencia aumenta el número de muestras en un mismo periodo temporal. Por lo tanto, precisión y coste computacional están reñidos, no siendo esta la mejor forma de aproximar las integrales de Fresnel.

No obstante, los últimos avances en el campo de la robótica y el diseño CAD permiten la aproximación de segmentos clotoidales obteniendo un error despreciable con un coste computacional muy bajo, lo cual permite generar trayectorias continuas para la navegación de vehículos de forma rápida y sencilla, pudiéndose incluir fácilmente en los planificadores de trayectorias.

Entre las aproximaciones presentadas en el Capítulo 2 destacan [54] y [60] por su gran precisión y bajo coste computacional, debido al hecho de que se trata de expresiones polinómicas de bajo orden y fáciles de calcular.

En [53] y posteriormente en [54] se hace una aproximación de las integrales de Fresnel mediante un método recursivo con un error  $|e| < 1 \cdot 10^{-9}$ . Se asume que la aproximación continua de las componentes de la Clotoide es la misma que en [51].

$$C(x) = \frac{1}{2} + f(x) \sin\left(\frac{\pi x^2}{2}\right) - g(x) \cos\left(\frac{\pi x^2}{2}\right) \quad (88)$$

$$S(x) = \frac{1}{2} - f(x) \cos\left(\frac{\pi x^2}{2}\right) - g(x) \sin\left(\frac{\pi x^2}{2}\right) \quad (89)$$

Se consigue una expresión genérica para aproximar las integrales de Fresnel y compensar los errores, mediante una ligera modificación de los parámetros  $f(x)$  y  $g(x)$  que aparecen en el método utilizado en [52]:

$$f(x) \approx f_0(x) = \frac{1+0.926x}{2+1.792x+3.104x^2} + \varepsilon(x) \quad (90)$$

$$g(x) \approx g_0(x) = \frac{1}{2+4.142x+3.492x^2+6.67x^3} + \varepsilon(x) \quad (91)$$

$$|\varepsilon(x)| \leq 2 \cdot 10^{-3}, \quad 0 < x < \infty \quad (92)$$

Además se propone una forma de calcularlos de forma recursiva para cualquier orden, aunque se particulariza para el caso de orden 11, presentando una tabla en la que aparecen pre-calculados los términos de la ecuación (ver Tabla 6).

$$f(x) = \sum_{n=0}^{11} f_n \cdot x^{-2n-1} \quad (93)$$

$$g(x) = \sum_{n=0}^{11} g_n \cdot x^{-2n-1} \quad (94)$$

Siendo:

$$f_n = q_n \cdot \left(\frac{8}{\pi}\right)^{n+1/2} \quad (95)$$

$$g_n = -p_n \cdot \left(\frac{8}{\pi}\right)^{n+1/2} \quad (96)$$

Con  $p_n$  y  $q_n$  constantes numéricas tabuladas en [49].

El método recursivo propuesto utiliza las ecuaciones anteriores para  $|x| > 1.6$ , puesto que para  $|x| \leq 1.6$  es mucho más precisa la aproximación mediante la expansión en serie de Taylor propuesta en [49].

$$C(x) = \sum_{n=0}^{\infty} c_n \cdot x^{4n+1} \quad (97)$$

$$S(x) = \sum_{n=0}^{\infty} s_n \cdot x^{4n+3} \quad (98)$$

Siendo:

$$c_0 = 1 \quad (99)$$

$$s_0 = \frac{\pi}{6} \quad (100)$$

El resto de parámetros se obtienen mediante estas fórmulas:

$$c_{n+1} = \frac{-\pi^2(4n+1)c_n}{4(2n+1)(2n+2)(4n+5)} \quad (101)$$

$$s_{n+1} = \frac{-\pi^2(4n+3)s_n}{4(2n+2)(2n+3)(4n+7)} \quad (102)$$

La otra posibilidad rápida y precisa de aproximar las componentes de una Clotoide se desarrolla ampliamente en [60], donde se estudian las ventajas de aproximación mediante curvas RBC. El procedimiento es muy sencillo y se va a explicar a continuación.

Lo primero que hay que hacer es obtener una secuencia de puntos pertenecientes a la Clotoide, por cualquiera de los métodos de aproximación de las integrales de Fresnel expuestos en el Capítulo 2. En esta primera etapa lo importante es la precisión de la aproximación, sin importar el coste computacional, ya que este proceso sólo se hará una vez de forma *off-line*.

En segundo lugar se realiza una aproximación de los puntos obtenidos a una curva de tipo Bezier. Para ello se recuerdan las principales características de este tipo de curva paramétrica. La formulación de una curva Bezier es la siguiente:

$$P(u) = \sum_{k=0}^N C_k \cdot \frac{N!}{k!(N-k)!} \cdot u^k \cdot (1-u)^{N-k} \quad (103)$$

Dónde:

- $C_k$ : Puntos de control Bezier.
- $u$ : Parámetro intrínseco  $[0..1]$ .
- $N$ : Orden de la ecuación de Bezier.

El parámetro normalizado  $t$  comprendido en el intervalo  $t_s \leq t \leq t_e$ , se introduce en la ecuación tal y como sigue:

$$P(t) = \sum_{k=0}^N C_k \cdot \frac{N!}{k!(N-k)!} \cdot \left(\frac{t_e-t}{t_e-t_s}\right)^k \cdot \left(1 - \frac{t_e-t}{t_e-t_s}\right)^{N-k} \quad (104)$$

La curva Bezier se puede expresar como una ecuación lineal:

$$P(t) = C_0 \cdot B_0(t) + C_1 \cdot B_1(t) + \dots + C_N \cdot B_N(t) \quad (105)$$

Donde  $B_t^k$  es la función  $k$ -ésima de la base de Bernstein, la cual se expresa como:

$$B_t^k = \frac{N!}{k!(N-k)!} \cdot \left(\frac{t_e-t}{t_e-t_s}\right)^k \cdot \left(1 - \frac{t_e-t}{t_e-t_s}\right)^{N-k} \quad (106)$$

Esta ecuación se puede expresar matricialmente de la siguiente forma:

$$\begin{bmatrix} P_{t_s} \\ \vdots \\ P_{t_e} \end{bmatrix} = \begin{bmatrix} B_{t_s}^0 & \dots & B_{t_s}^N \\ \vdots & \ddots & \vdots \\ B_{t_e}^0 & \dots & B_{t_e}^N \end{bmatrix} \cdot \begin{bmatrix} C_0 \\ \vdots \\ C_N \end{bmatrix} \quad (107)$$

$$P = B \cdot C$$

Dicha representación permite el uso de la técnica de los mínimos cuadrados (*Least Squares* - LS) para computar los puntos de control Bezier que mejor se ajusten a los puntos obtenidos  $P$  de la curva discreta. Por lo tanto, la estimación de los puntos de control de la curva Bezier se puede obtener mediante el uso de la pseudo-inversa:

$$\hat{C} = (B^T \cdot B)^{-1} \cdot B^T \cdot P \quad (108)$$

Si se implementan las ecuaciones anteriores se pueden obtener los puntos de control de una curva Bezier de orden  $N=11$ , como se muestra en la gráfica de la Figura 101.

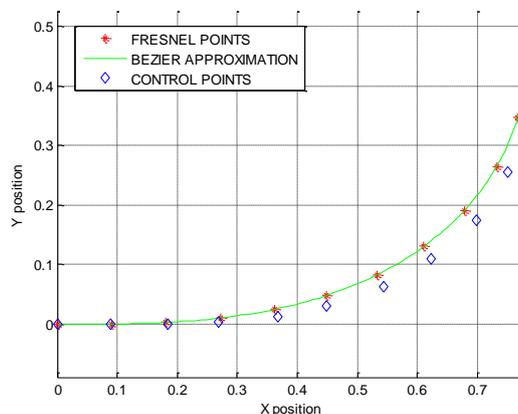


Figura 101. Aproximación de los puntos de Fresnel con una curva Bezier de orden  $N=11$ .

El problema de este tipo de soluciones es que los puntos primero y último de la curva sólo se aproximan, es decir, no se garantiza que se pase por estos puntos, lo cual es un problema para la generación de trayectorias en robótica móvil. Por lo tanto, no es posible la continuidad de posición y en consecuencia no se pueden unir dos segmentos de Clotoide para formar trayectorias más complejas.

Para resolver el problema se obtiene una curva equivalente mediante una Bezier Racional (RBC), calculando sus puntos de control y pesos. Pero para utilizar las curvas racionales de Bezier primero hay que conocer la ecuación que modela este tipo de curvas.

$$P(u) = \frac{\sum_{k=0}^N w_k \cdot C_k \cdot \frac{N!}{k!(N-k)!} \cdot u^k \cdot (1-u)^{N-k}}{\sum_{k=0}^N w_k \cdot \frac{N!}{k!(N-k)!} \cdot u^k \cdot (1-u)^{N-k}} \quad (109)$$

Dónde:

- $C_k$ : Puntos de control de Bezier.
- $W$ : Pesos de los puntos de control de Bezier.
- $N$ : Orden de la ecuación de Bezier (determina el número de puntos de control).
- $u$ : Parámetro intrínseco  $[0 - 1]$ .

Se introduce en la ecuación el parámetro normalizado  $t$ , comprendido en el intervalo de trabajo  $t_s \leq t \leq t_e$ :

$$P(t) = \frac{\sum_{k=0}^N w_k \cdot C_k \cdot \frac{N!}{k!(N-k)!} \cdot \left(\frac{t_e-t}{t_e-t_s}\right)^k \cdot \left(1 - \frac{t_e-t}{t_e-t_s}\right)^{N-k}}{\sum_{k=0}^N w_k \cdot \frac{N!}{k!(N-k)!} \cdot \left(\frac{t_e-t}{t_e-t_s}\right)^k \cdot \left(1 - \frac{t_e-t}{t_e-t_s}\right)^{N-k}} \quad (110)$$

Para obtener una coincidencia exacta entre los puntos de control y los puntos reales de las integrales de Fresnel es necesario fijar a valor unitario el primer y el último peso ( $w_k = 1$ ) y forzar tanto el primer como el último punto de control al valor exacto deseado. A partir de esta situación es posible obtener los pesos del resto de puntos utilizando mínimos cuadrados, tal y como se desarrolla a continuación.

Al igual que para la curva Bezier, la RBC también se puede expresar como una ecuación lineal para un parámetro determinado  $t$ :

$$P(t) = \frac{w_0 \cdot C_0 \cdot B_0(t) + w_1 \cdot C_1 \cdot B_1(t) + \dots + w_N \cdot C_N \cdot B_N(t)}{w_0 \cdot B_0(t) + w_1 \cdot B_1(t) + \dots + w_N \cdot B_N(t)} \quad (111)$$

Esta ecuación puede expresarse de la siguiente forma matricial:

$$P = B \cdot w \quad (112)$$

Donde las matrices son:

$$P = \begin{bmatrix} P(t_s) \cdot [B_0(t_s) + B_N(t_s)] - P(t_s) \cdot B_0(t_s) - P(t_e) \cdot B_N(t_s) \\ \vdots \\ P(t_e) \cdot [B_0(t_e) + B_N(t_e)] - P(t_s) \cdot B_0(t_e) - P(t_e) \cdot B_N(t_e) \end{bmatrix} \quad (113)$$

$$B = \begin{bmatrix} (C_1 - P(t_s)) \cdot B_1(t_s) & \dots & (C_{N-1} - P(t_s)) \cdot B_{N-1}(t_s) \\ \vdots & \ddots & \vdots \\ (C_1 - P(t_e)) \cdot B_1(t_e) & \dots & (C_{N-1} - P(t_e)) \cdot B_{N-1}(t_e) \end{bmatrix} \quad (114)$$

$$w = [w_1 \quad \dots \quad w_{N-1}]^T \quad (115)$$

La estimación de los pesos de la curva R-Bezier puede obtenerse mediante el uso de la técnica LS:

$$\hat{w} = (B^T \cdot B)^{-1} \cdot B^T \cdot P \quad (116)$$

Por lo tanto, tras el ajuste *off-line* de las integrales de Fresnel con una curva Bezier se obtiene una primera aproximación. Pero como se ha comentado ésta no pasa a través de los puntos de control inicial y final, aunque a primera vista pueda parecer que sí lo hace (ver Figura 101). En cambio, cuando la aproximación se realiza mediante RBC el ajuste pasa tanto por el primero como por el último de los puntos que se desea ajustar (ver Figura 102). Esto es muy deseable en robótica móvil, ya que el objetivo es unir dos configuraciones dadas y por las cuales para un instante determinado es obligatorio pasar.

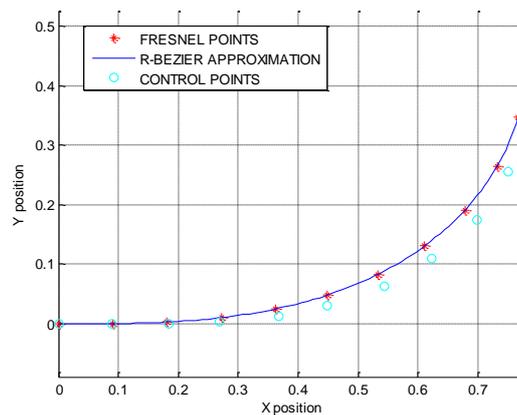


Figura 102. Aproximación de los puntos de Fresnel con una curva R-Bezier de orden N=11.

En la Figura 103 se muestran ambas aproximaciones superpuestas en la misma gráfica:

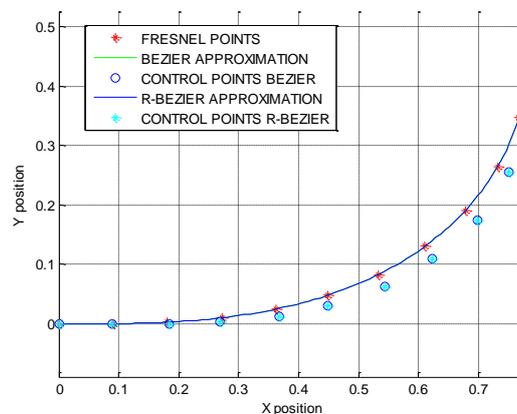


Figura 103. Aproximación de los puntos de Fresnel con Bezier y R-Bezier para N=11 en el rango  $[0 - \pi/2]$ .

Aunque parece que están exactamente superpuestas, si se hace un zoom, podemos ver en la Figura 104 que las curvas obtenidas con ambas aproximaciones no son exactamente iguales.

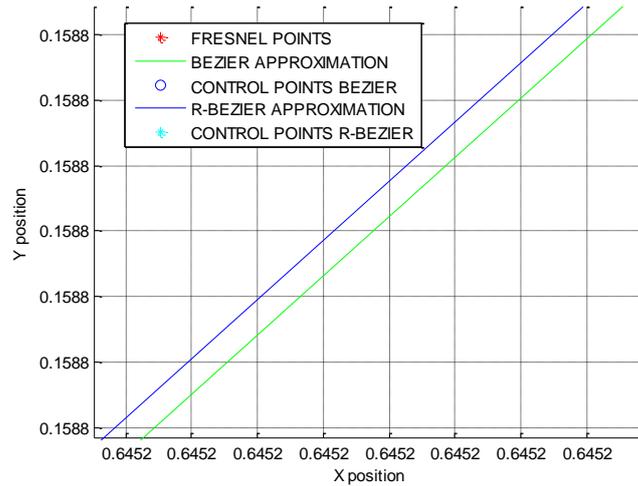


Figura 104. Detalle de la diferencia entre la aproximación Bezier y la R-Bezier.

A continuación se presenta la tabla correspondiente de pesos y puntos de control, obtenidos por el autor de la presente Tesina, tanto para la coordenada X como para la Y.

N	$C_{RB}(X)$	$C_{RB}(Y)$	$W_{RB}(X)$	$W_{RB}(Y)$
1	0	0	1	1
2	0.090909093855915	-0.000000060427673	0.999999947585571	0.999992888174795
3	0.181818171498556	0.000000372754417	1.000000106570532	0.999997302580205
4	0.272727269370114	0.003171898349694	0.999999737635535	1.000016398760370
5	0.363636543367710	0.012697158815914	1.000000361740489	0.999973605574094
6	0.454010467930419	0.031725309912622	0.999999605831691	1.000026962736480
7	0.542253189312309	0.063479948515684	1.000000217927580	0.999978631035690
8	0.625139690865974	0.110766552745435	0.999999987918934	1.000012611142058
9	0.697387604730645	0.175499400405351	0.999999818487950	0.999993711561134
10	0.751333319352615	0.256441409926406	1.000000197625849	1.000002067397683
11	0.779893421317649	0.347349986518348	0.999999844500825	0.999999120508584
12	0.779893400376823	0.438259147390355	1	1

Tabla 10. Pesos y puntos de control obtenidos para las coordenadas X e Y (C y S).

Como se observa, difieren ligeramente respecto a las obtenidas en la Tabla 7. Esto se debe al hecho de que los puntos tomados para aproximar la Clotoide real no son los mismos ni en valor (métodos de aproximación distintos) ni en número y dicha diferencia hace que los valores varíen ligeramente. No obstante esto no es un problema porque se trata de diferencias insignificantes.

Una vez se ha calculado y fijado como base la aproximación de la Clotoide unitaria, *on-line* sólo es necesario calcular la rotación y la translación correspondiente para generar la trayectoria deseada. En este proceso, los coeficientes, los pesos y los puntos de control se mantienen invariables.

Como conclusión se dirá que las Clotoides tienen la propiedad de que la variación de su radio de curvatura es constante, permitiendo una transición suave desde los tramos rectos a los arcos de circunferencia y viceversa. Además de por sus características geométricas también conviene utilizar trayectorias de este tipo por el hecho de que se pueden aproximar de forma precisa, con poco coste computacional, y también porque el radio de la curva Clotoide es fácilmente trasladable a un parámetro controlable del vehículo como es el giro de las ruedas.

### 3.5. Elementary Path

Una Clotoide completa empieza con radio infinito y termina con radio nulo, de esta forma no es útil desde el punto de vista de generación de trayectorias y control en robótica móvil, por eso se trabaja con segmentos clotoidales como el mostrado en la Figura 105.

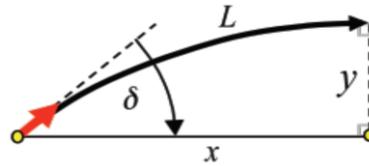


Figura 105. Segmento clotoidal.

Un tramo clotoidal es aquél que permite ir desde un punto con una curvatura determinada a otro de curvatura distinta desplazándose a lo largo de la Clotoide. Por lo tanto, en el caso más genérico no es necesario empezar ni terminar con curvatura nula, aunque en muchos casos así se haga.

En la generación de trayectorias existen 3 configuraciones posibles del punto final a alcanzar: posición  $q(x, y)$ , pose  $q(x, y, \theta)$  y postura  $q(x, y, \theta, \kappa)$ . Pese a que la Clotoide permite ir a cualquier posición, está muy restringida geoméricamente y la mayor limitación de esta curva es que no es posible ir a cualquier configuración del espacio, ni de pose  $q_E(x, y, \theta)$  ni de postura  $q_E(x, y, \theta, \kappa)$ . Es por ello que usualmente las Clotoides no se utilizan solas sino combinadas entre ellas o con otros elementos.

En [13] se introduce el concepto de *Elementary path*, que es una combinación de dos Clotoides iguales, es decir, con el mismo factor de homotecia. Dichas curvas se conectan en serie de modo que para que dos posturas puedan ser enlazadas deben ser simétricas, ya que de este modo se conserva la continuidad de la curvatura.

El hecho de que se tenga que cumplir la ecuación (117) hace que esta solución también esté restringida geoméricamente, pudiéndose alcanzar cualquier posición, pero no cualquier pose  $q_E(x, y, \theta)$  ni cualquier postura  $q_E(x, y, \theta, \kappa)$ . Además sólo permite enlazar tramos de curvatura nula puesto que las Clotoides se unen por su extremo con curvatura distinta de cero. En la Figura 106 se muestra el ejemplo de un *Elementary path*, en el que se observa la simetría entre la configuración inicial y la final.

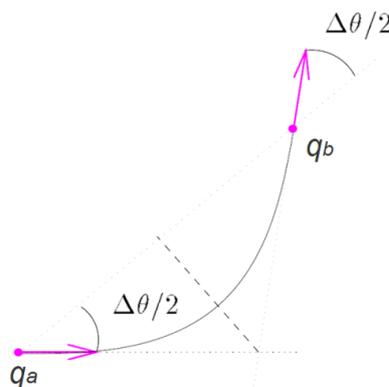


Figura 106. Elementary path.

Las coordenadas  $q_a$  y  $q_b$  deben verificar:

$$(x_b - x_a) \cdot \sin\left(\frac{\tau_b + \tau_a}{2}\right) = (y_b - y_a) \cdot \cos\left(\frac{\tau_b + \tau_a}{2}\right) \quad (117)$$

Siendo  $\frac{\tau_b + \tau_a}{2}$  el ángulo  $\frac{\Delta\theta}{2}$  que aparece en la Figura 111, así como  $(x_a, y_a)$  y  $(x_b, y_b)$  las coordenadas cartesianas de las configuraciones inicial  $q_a$  y final  $q_b$ , respectivamente.

### 3.6. BiElementary Path

La aportación fundamental de [14] es el desarrollo del concepto de *BiElementary path*, que es la combinación de dos *Elementary paths*, tal y como aparece en la Figura 107. En el Capítulo 2 ya se anticipó que pese a que las configuraciones inicial y final de un *BiElementary path* no sean simétricas éstas pueden unirse mediante una configuración intermedia que garantice que es simétrica con ambas, de modo que existen infinitas candidatas para la localización de dicha configuración. Pese a haber infinitas soluciones, el objetivo es obtener la trayectoria que minimiza la longitud del arco que une ambas configuraciones.

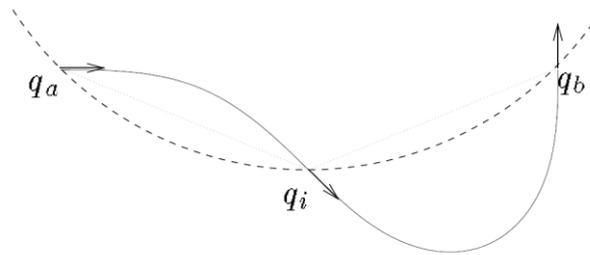


Figura 107. BiElementary Path.

Como ya se ha dicho, las poses inicial y final no son simétricas, pero pueden ser unidas mediante una configuración intermedia simétrica con ambas, de modo que sea enlazable mediante dos *Elementary paths*. La búsqueda de la configuración intermedia correcta es bastante simple, una vez dado el punto  $(X, Y)$  de dicha configuración, el objetivo es encontrar la orientación que haga cumplir las ecuaciones (118) y (119).

$$(x_i - x_a) \cdot \sin\left(\frac{\tau_i + \tau_a}{2}\right) = (y_i - y_a) \cdot \cos\left(\frac{\tau_i + \tau_a}{2}\right) \quad (118)$$

$$(x_b - x_i) \cdot \sin\left(\frac{\tau_b + \tau_i}{2}\right) = (y_b - y_i) \cdot \cos\left(\frac{\tau_b + \tau_i}{2}\right) \quad (119)$$

La principal problemática de estas trayectorias es que unen puntos de curvatura nula y por lo tanto, pese a que se puede alcanzar cualquier pose  $q_E(x, y, \theta)$  no se puede llegar a cualquier postura  $q_S(x, y, \theta, \kappa)$ . Una posible solución sería incluir un tramo clotoidal en cada uno de los extremos del *BiElementary path*, de este modo se podría llegar desde una postura determinada  $q_S(x, y, \theta, \kappa)$  hasta otra cualquiera  $q_E(x, y, \theta, \kappa)$  mediante la combinación de seis Clotoides consecutivas.

### 3.7. Continuous Curvature Turn (CC-turn)

Una limitación importante de las trayectorias que se han explicado anteriormente es el hecho de que no consideran restricciones, ni cinemáticas ni dinámicas. De hecho, con un *BiElementary path* se puede alcanzar cualquier pose geoméricamente, pero a la hora de generar trayectorias reales para el control cinemático de robots móviles existen una serie de restricciones que hay que considerar, como el ángulo máximo de giro en vehículos no holonómicos o las limitaciones máximas de curvatura o brusquedad.

Con tal de solucionar el problema de las restricciones geométricas y físicas, y así seguir utilizando las Clotoides dadas las ventajas que ofrecen para generación de trayectorias continuas, los autores de [15] crean los conocidos *CC-turns* (ver Figura 108), que son una generalización del *Elementary path*. Se trata de dos Clotoides unidas mediante un arco circular de la misma curvatura que las Clotoides en su lado de unión. Si se analiza bien, un *Elementary path* es un *CC-turn* con un arco circular de longitud nula y cuyas Clotoides son simétricas, aunque en un *CC-turn* no necesariamente deban serlo.

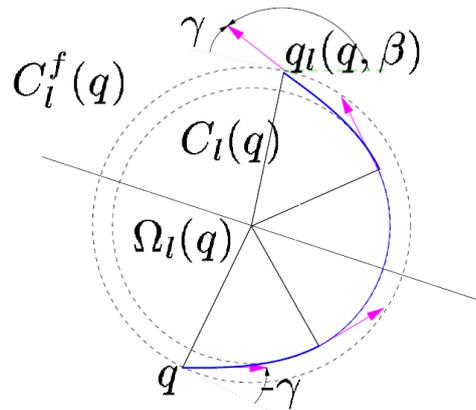


Figura 108. *CC-turn*.

Gracias a la inclusión del tramo circular se consigue tener un elemento básico de generación que permite girar más que un *Elementary path* para una curvatura máxima determinada. Por lo tanto, si se utilizan *CC-turns* se pueden considerar restricciones cinemáticas, de modo que aunque existan limitaciones de curvatura se pueda llegar al objetivo deseado incluyendo tramos circulares con dicha curvatura máxima.

### 3.8. Continuous Curvature Path (CC-path)

En [15] además de los *CC-turns* también introducen el concepto de *SCC-path*, compuesto por dos *CC-turns* unidos por un segmento de recta (Figura 109). Definen cada elemento fundamental de unión entre dos configuraciones consecutivas como la combinación de como máximo 7 piezas entre Clotoides, arcos circulares y segmentos de línea.

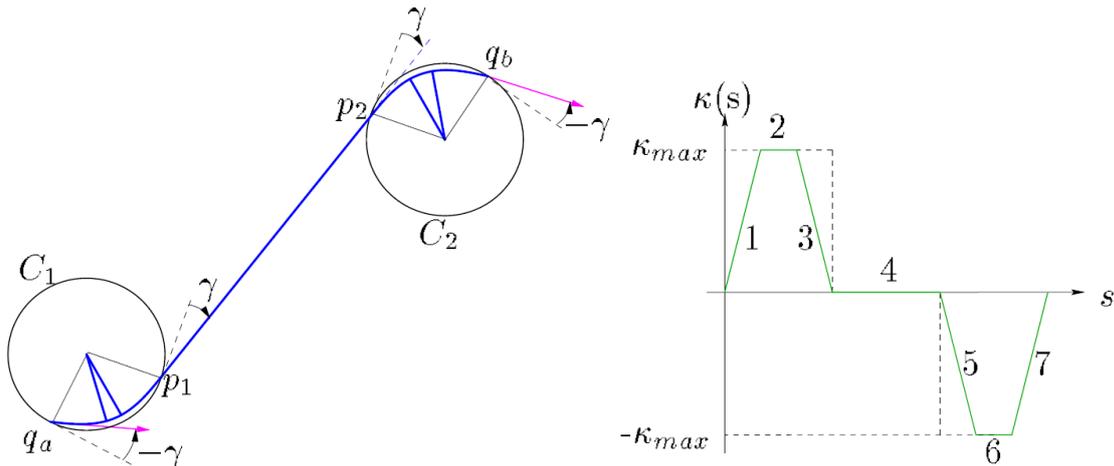


Figura 109. *SCC-path* (izquierda) y su perfil de curvatura (derecha).

No obstante, como se ha comentado en el Capítulo 2, la aportación más significativa para la comunidad robótica sobre generación de trayectorias continuas con restricciones cinemáticas y dinámicas es la de [30]. Tal y como se ha dicho en el párrafo anterior, en [15] los autores derivaron los *RS-paths* introducidos en [6] y crearon los *SCC-paths*, que son similares pero que además aseguran la continuidad, ya que sustituyen los arcos circulares por lo que denominan *CC-turns*, que son una generalización de los *BiElementary paths*. En la Figura 110 se muestra un ejemplo de perfil de curvatura de los *RS-paths* y de los *SCC-paths*.

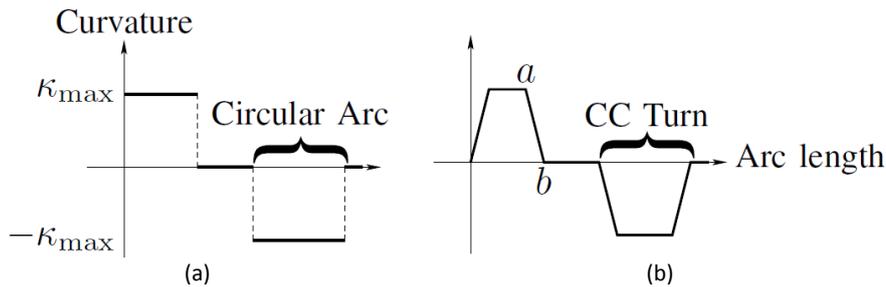


Figura 110. Perfiles de curvatura: (a) *RS-path* y (b) *SCC-path*. El tramo  $a - b$  es un arco de Clotoide.

La principal aportación de este artículo es el hecho de obtener trayectorias limitadas tanto cinemática como dinámicamente. En primer lugar establece el modelo del vehículo, el cual, al tener una configuración igual que la de un coche y no como un robot, tiene limitaciones cinemáticas en el ángulo de giro de las ruedas orientables.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\kappa} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \kappa \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \sigma \quad (120)$$

Este modelo extiende el introducido en [3] ya que incorpora un cuarto estado del sistema, la curvatura. Tal y como se observa, este modelo dinámico tiene dos entradas de control:

- $v$  es la velocidad de conducción de las ruedas de tracción (traseras).
- $\sigma$  es la aceleración angular que está relacionada con  $\dot{\phi}$ , la velocidad de giro de las ruedas orientables (delanteras).

En la Figura 111 se observa el modelo gráfico del vehículo utilizado.

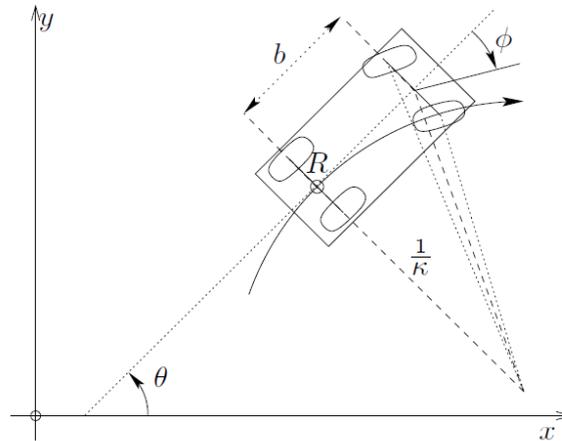


Figura 111. Modelo de tipo coche del vehículo:  $R = (x, y)$  es el punto de referencia y  $\theta$  es la orientación del vehículo,  $\phi$  es el ángulo de orientación de las ruedas y  $b$  es la separación entre los ejes de las ruedas.

Si se considera  $b$  como la separación de las ruedas se mantiene la siguiente relación:

$$\kappa = \frac{\tan \phi}{b} \quad (121)$$

$$\sigma = \dot{\kappa} = \frac{\dot{\phi}}{b \cos^2 \phi} \quad (122)$$

En [25] el objetivo es obtener la planificación del camino más corto sujeto a restricciones. Se asume una velocidad de conducción unitaria con un límite superior de la aceleración angular (suponiendo que la velocidad actual de la dirección del vehículo está limitada):

$$v = 1 \rightarrow |\sigma| \leq \sigma_{max} \quad (123)$$

Bajo esas suposiciones el trabajo realizado en [30] determina que la planificación de la trayectoria entre dos configuraciones es equivalente a obtener una curva plana de curvatura continua que permita unir dos puntos en el plano con una orientación y una curvatura determinadas. Como ya se ha comentado los *CC-paths* combinan *CC-turns* y tramos rectos para planificar rutas de curvatura continua. Pero existen problemas a la hora de implementarse puesto que en dichas soluciones no se tiene en cuenta la controlabilidad del sistema en periodos de tiempo pequeños, es decir, que para ir de una configuración a otra pueden requerirse acciones de control elevadas en periodos de tiempo cortos, lo cual puede no ser realizable físicamente por el vehículo debido a sus limitaciones. Para evitar ese problema se presenta un nuevo tipo de *CC-path* en el que existe limitación de *sharpness*  $|\sigma| \leq \sigma_{max}$  y limitación de la curvatura máxima o en consecuencia del radio mínimo  $R_{min} = \kappa_{max}^{-1}$ .

El caso general de un *CC-turn* permite dividir su geometría en tres partes:

- Un arco de Clotoide con *sharpness*  $\sigma = \pm \sigma_{max}$ , con curvatura desde 0 hasta  $\pm \kappa_{max}$ .
- Un tramo circular con radio  $\pm \kappa_{max}^{-1}$ .
- Un arco de Clotoide con *sharpness*  $\sigma = \mp \sigma_{max}$ , con curvatura desde  $\mp \kappa_{max}$  hasta 0.

Si se considera  $q_s = (x_s, y_s, \theta_s, 0)$  como la configuración inicial de un *CC-turn*, el vehículo seguirá un arco de Clotoide con una longitud  $\kappa_{max}/\sigma_{max}$  y un *sharpness*  $\sigma_{max}$  antes de llegar a la configuración  $q_i$ .

$$q_i = \begin{cases} x_i = \sqrt{\pi/\sigma_{max}} \cdot C_f \cdot \left(\sqrt{\kappa_{max}^2/\pi\sigma_{max}}\right) \\ y_i = \sqrt{\pi/\sigma_{max}} \cdot S_f \cdot \left(\sqrt{\kappa_{max}^2/\pi\sigma_{max}}\right) \\ \theta_i = \kappa_{max}^2/2\sigma_{max} \\ \kappa_i = \kappa_{max} \end{cases} \quad (124)$$

Donde  $C_f$  y  $S_f$  son puntos de las Integrales de Fresnel.

A continuación se llega a un tramo circular con radio  $\kappa_{max}^{-1}$  antes de llegar a  $q_j = (x_j, y_j, \theta_j, \kappa_{max})$ . El centro del arco circular  $\Omega$  se sitúa a una distancia  $\kappa_{max}^{-1}$  de  $q_i$  en la dirección normal a  $\theta_i$ .

$$\Omega = \begin{cases} x_\Omega = x_i - \kappa_{max}^{-1} \sin \theta_i \\ y_\Omega = y_i + \kappa_{max}^{-1} \cos \theta_i \end{cases} \quad (125)$$

Y el radio  $r$  de dicho arco circular es  $r = \sqrt{x_\Omega^2 + y_\Omega^2}$ .

Finalmente el vehículo sigue un arco de Clotoide con *sharpness*  $-\sigma_{max}$  antes de llegar a la configuración objetivo  $q_g = (x_g, y_g, \theta_g, 0)$ .

En la Figura 112 se muestra un ejemplo de un *CC-turn* común, en el que la trayectoria seguida no tiene ninguna complicación.

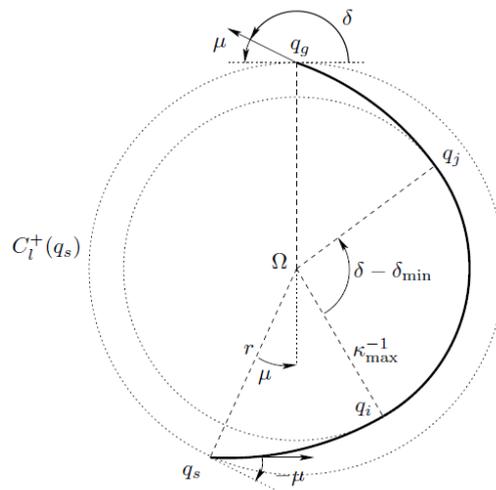


Figura 112. Caso general de *CC-turn*.

Se define el ángulo de deflexión  $\delta = (\theta_g - \theta_s)$  y representa el cambio de orientación entre  $q_s$  y  $q_g$ . Gracias a dicho parámetro se puede caracterizar el *CC-turn*. Un caso particular, pero muy útil para posteriores aproximaciones, es el ángulo de deflexión de un *CC-turn* cuyo arco circular tiene longitud nula, el cual toma el siguiente valor  $\delta_{min} = \kappa_{max}^2 \sigma_{max}^{-1}$ .

En la Figura 113 se muestra un ejemplo de dos casos especiales de *CC-turn* común, uno para  $\delta = 0$  y otro para  $0 < \delta < \delta_{min}$ .

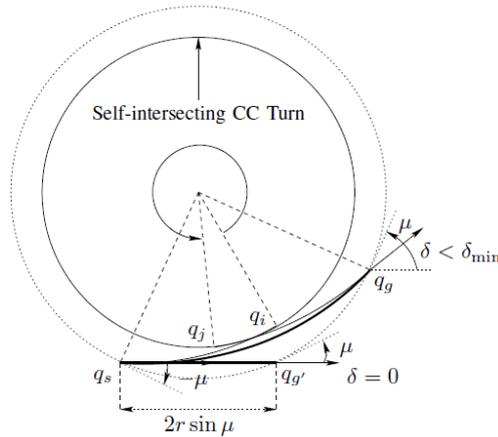


Figura 113. Casos particulares de *CC-turn*:  $\delta = 0$  y  $0 < \delta < \delta_{min}$ .

Existe un caso particular de *CC-turn* en el que se requieren deflexiones muy pequeñas o muy grandes. En el artículo se explica que el método propuesto es capaz de resolverlo puesto que tiene la capacidad de alcanzar configuraciones más atrasadas de modo que siempre se obtiene una solución factible. Este caso particular se resuelve incluyendo una primera Clotoide para llegar desde la configuración inicial  $q_s$  hasta la primera configuración intermedia  $q_i$ . Después se utiliza un arco circular para llegar hasta la segunda configuración intermedia  $q_j$  que está más atrasada y por tanto requiere que el vehículo retroceda. Finalmente se introduce otra Clotoide (igual a la primera) para alcanzar la configuración destino  $q_g$ .

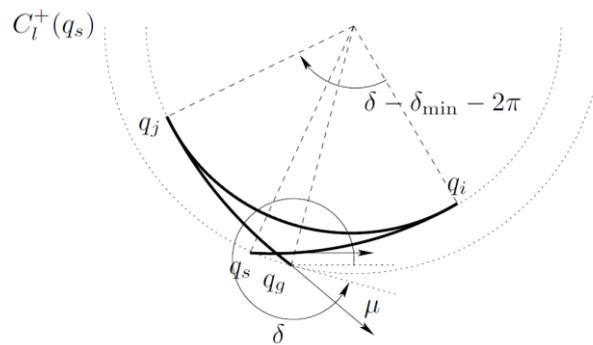


Figura 114. Caso particular de *CC-turn*:  $\delta_{min} + \pi \leq \delta < 2\pi$ .

Por lo tanto, como conclusión hay que destacar que la principal aportación de este artículo es el hecho de que basándose en los conocidos *CC-paths* los autores son capaces de resolver problemas de planificación con restricciones cinemáticas (limitaciones de curvatura debidas a limitaciones físicas en los ángulos de giro del vehículo controlado, que se representan por una  $\kappa_{max}$  o en consecuencia por un  $\delta_{min}$ ) y dinámicas (limitaciones en el *sharpness* o brusquedad y en consecuencia en el *jerk* sufrido por el vehículo en su movimiento, que se representa por  $\sigma_{max}$ ).

### 3.9. Técnica para la Generación Off-line de Trayectorias de Curvatura Continua, para el Problema del Seguimiento de Línea con Curvatura y Brusquedad Acotadas

Como ya se ha comentado en apartados anteriores en [102] se desarrolla un algoritmo de control para realizar el seguimiento de un camino mediante arcos de circunferencia, pero dicha técnica tiene el problema de que ante cambios bruscos en la orientación de la línea detectada se producen discontinuidades en la referencia del vehículo, lo que provoca que aparezcan fuerzas centrípetas instantáneas que provocan *sharpness* elevados y en consecuencia *jerks* elevados, lo cual puede provocar errores de odometría y un comportamiento indeseado del vehículo, como por ejemplo deslizamientos de las ruedas. Para solucionar estos problemas se propone una técnica de seguimiento de línea que incorpora trayectorias clotoideas que permiten generar trayectorias suaves, continuas en curvatura. Además, este método se basa en los *CC-paths* introducidos en [15], de modo que el sistema es capaz de incorporar restricciones tanto cinemáticas como dinámicas, garantizando que en todo momento el vehículo no supere ni la curvatura máxima ni el *sharpness* o *jerk* máximos.

Por lo tanto el objetivo final es el seguimiento de líneas con trayectorias de curvatura continua y considerando restricciones. Se utilizan elementos derivados de los *CC-paths* de [15], de forma que el parámetro de curvatura máxima y la brusquedad (derivada de curvatura) son parámetros de diseño que se pueden controlar fácilmente en la etapa de planificación. Con dichos parámetros determinamos un punto en la trayectoria que es factible de alcanzar mediante un *CC-path* y que además satisfaga las restricciones impuestas.

#### 3.9.1. Planteamiento del Problema

A continuación se detallará el problema que se desea abordar de una forma más estricta desde el punto de vista matemático, de modo que queden definidas todas las variables que intervienen en el problema del seguimiento de líneas.

Dado  $\mathcal{R}$  un robot con ruedas no-holonómico moviéndose en un plano 2D, con un espacio de estados extendido  $q = (x, y, \theta, \kappa)^T \in \mathbb{R}^2 \times S \times \mathbb{R}$ , que contiene la posición en el espacio cartesiano  $x$  e  $y$ , la orientación  $\theta$  y la curvatura  $\kappa$  del robot. El modelo cinemático del robot  $\mathcal{R}$  es:

$$\dot{q}(t) = \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \\ \dot{\kappa}(t) \end{pmatrix} = \begin{pmatrix} v \cdot \cos \theta \\ v \cdot \sin \theta \\ v \cdot \kappa(t) \\ v \cdot \sigma \end{pmatrix} \quad (126)$$

Siendo  $v$  la velocidad y  $\sigma$  el *sharpness* que describen el camino, ambos se asumen constantes por simplicidad. Sin pérdida de generalidad la configuración del robot se sitúa en el origen  $q_A = (x_A, y_A, \theta_A, \kappa_A)^T$  con  $x_A = 0$  e  $y_A = 0$ , la curvatura inicial nula  $\kappa_A = 0$  y una orientación arbitraria  $\theta_A$ .

Además, también se considera que el movimiento del robot  $\mathcal{R}$  por el plano 2D estará acotado en curvatura  $\kappa \in [-\kappa_{max}, \kappa_{max}]$  y en *sharpness*  $\sigma \in [\sigma_{min}, \sigma_{max}]$ . De modo que las restricciones de curvatura se deben a limitaciones mecánicas de las ruedas orientables del robot y los límites de brusquedad se introducen principalmente por seguridad y confortabilidad.

Es muy común relajar las restricciones hasta  $\sigma_{min} = 0$  y factible en la mayoría de los casos de generación de trayectorias para robots móviles, aunque se incluirá  $\sigma_{min} > 0$  con tal de obtener una formulación más genérica.

Hay que remarcar que el objetivo es generar un *continuous-curvature path*  $\mathcal{P}$  para unir la postura actual del robot  $q_A$  (con curvatura inicial nula) con otra configuración  $q_B = (x_B, y_B, \theta_B, \kappa_B)^T$ . El problema se aborda de modo que  $x_B = d_h$  sea una separación lateral con respecto a la posición del robot, la curvatura final sea nula  $\kappa_B = 0$ , la orientación sea  $\theta_B$  y se llegue a una separación vertical  $y_B$  determinada, para la cual el vehículo converge a la línea.

Sin pérdida de generalidad se considera una orientación vertical de la línea objetivo  $\theta_B = \pi/2$  y una separación positiva  $d_h > 0$ , con el criterio tradicional de signos para ángulos y distancias, utilizando un sistema de coordenadas cartesiano.

Otro aspecto que hay que destacar es que el *path* generado no sólo incluye una trayectoria en el espacio 2D cartesiano, sino que también considera la orientación del vehículo (componente tangencial del movimiento), su curvatura y la derivada de su curvatura.

Antes de detallar el procedimiento de generación de trayectorias de curvatura continua para el problema del seguimiento de líneas, se realizará un estudio de las diferentes situaciones en las que se puede encontrar el robot con respecto a la línea que debe seguir, de modo que se demuestre la necesidad del uso de *continuous-curvature paths* por las ventajas que ofrecen.

La Figura 115 (a) muestra el tipo de soluciones que se pueden obtener utilizando dos Clotoides simétricas (*Elementary path*). Dependiendo del ángulo entre la línea y la configuración inicial del robot (ángulo de deflexión  $\delta$ ), se puede ajustar el *sharpness* y la curvatura de la Clotoide para converger de forma paralela a la línea. Pero en general no se puede garantizar que la convergencia no tenga un *offset* o *bias* y por lo tanto es necesario incluir un segmento inicial para compensar dicho error. Además, cuando los ángulos de deflexión son demasiado elevados o los límites de curvatura y *sharpness* demasiado estrechos, el uso de dos Clotoides simétricas no permite llegar a la orientación deseada y se hace necesario el uso de un arco circular para compensar la parte del ángulo de deflexión que no han sido capaz de girar el *Elementary path*. En la Figura 115 (b) se muestra a nivel conceptual lo que ocurre en dichas situaciones y por qué se introduce el segmento de línea el arco circular.

Es necesario aclarar que en los cuatro ejemplos presentados se muestran los casos en los que el error de *bias* debe ser compensado con un segmento positivo (PB) o negativo (NB), para sus valores de *sharpness* mínimo y máximo.

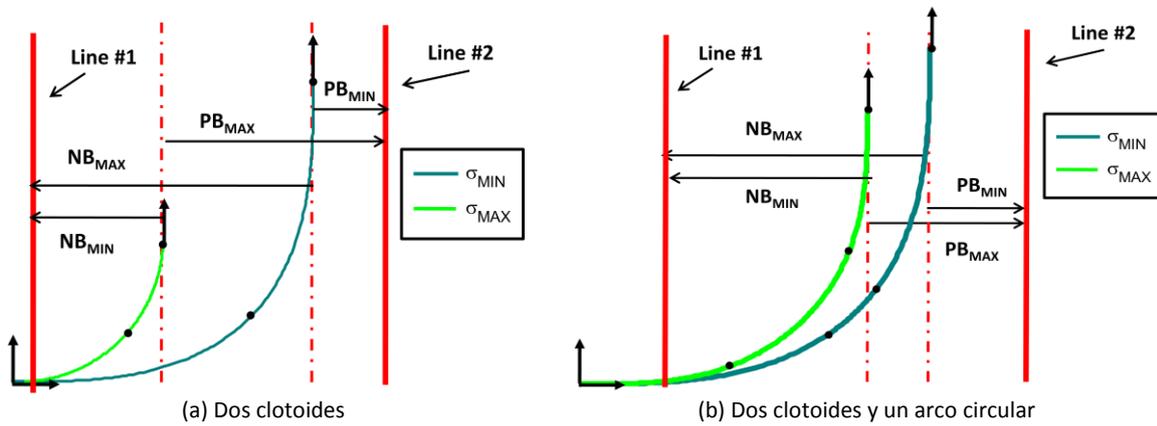


Figura 115. Espacio de soluciones con (a) dos Clotoides y (b) opcionalmente un arco circular para compensar el ángulo de deflexión y un segmento de línea para compensar el error de *bias*.

A diferencia de los casos anteriores, puede haber situaciones donde sea necesario un cambio del signo de curvatura, como muestra la Figura 116 (b). En ese caso particular las orientaciones inicial y final son iguales, pero para compensar el error de *bias* se necesita primero disminuir y después aumentar la orientación y en consecuencia cambiar el signo de la curvatura, por lo que se hace necesario el uso de cuatro Clotoides para completar el *path*.

Al igual que en los ejemplos previos, se pueden necesitar dos arcos circulares en caso de ángulos de deflexión grandes o límites estrechos, cada uno conectando un par de Clotoides en su punto de máxima curvatura. De forma similar se pueden incluir hasta dos segmentos de recta para compensar el error de *bias*, uno al inicio del camino, como se muestra en la Figura 116 (a) o entre las Clotoides intermedias, como aparece en la Figura 116 (b).

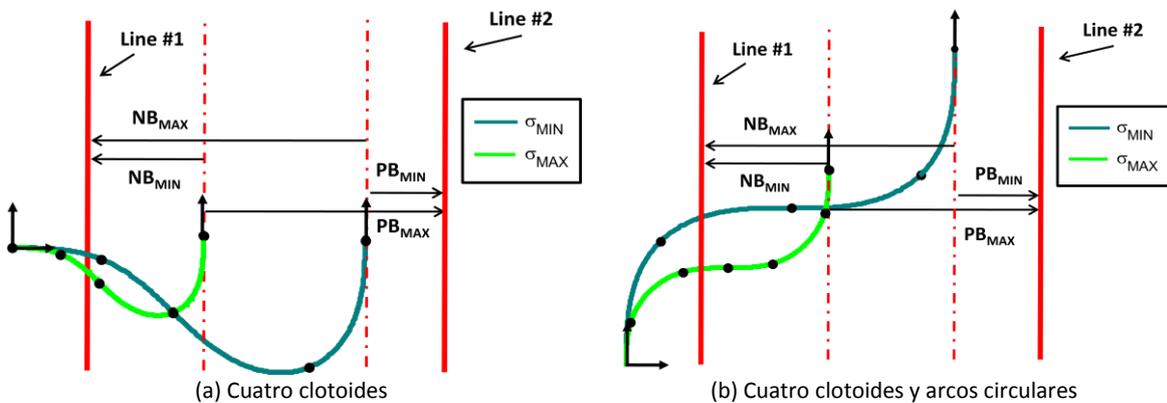


Figura 116. Espacio de soluciones: (a) con cuatro Clotoides y (b) opcionalmente arcos circulares y segmentos de línea para compensar el ángulo de deflexión y el error de *bias* respectivamente.

Como ya se había avanzado queda demostrada la necesidad de utilizar tramos rectilíneos y arcos circulares para resolver cualquier situación y que el vehículo sea capaz de converger a la línea deseada sin errores.

### 3.9.2. Single Continuous Curvature Path (SCC-path)

En este apartado se va a definir el concepto de *Single Continuous-Curvature path (SCC-path)* introducido en [103]. Se explicarán sus características principales y la metodología a seguir para su generación.

En primer lugar se introducen los elementos básicos que forman un *SCC-path*: un segmento de recta, una primera Clotoide, un arco circular y una segunda Clotoide; tal y como muestra la Figura 117 (a). El perfil de curvatura de dicha trayectoria aparece en la Figura 117 (b).

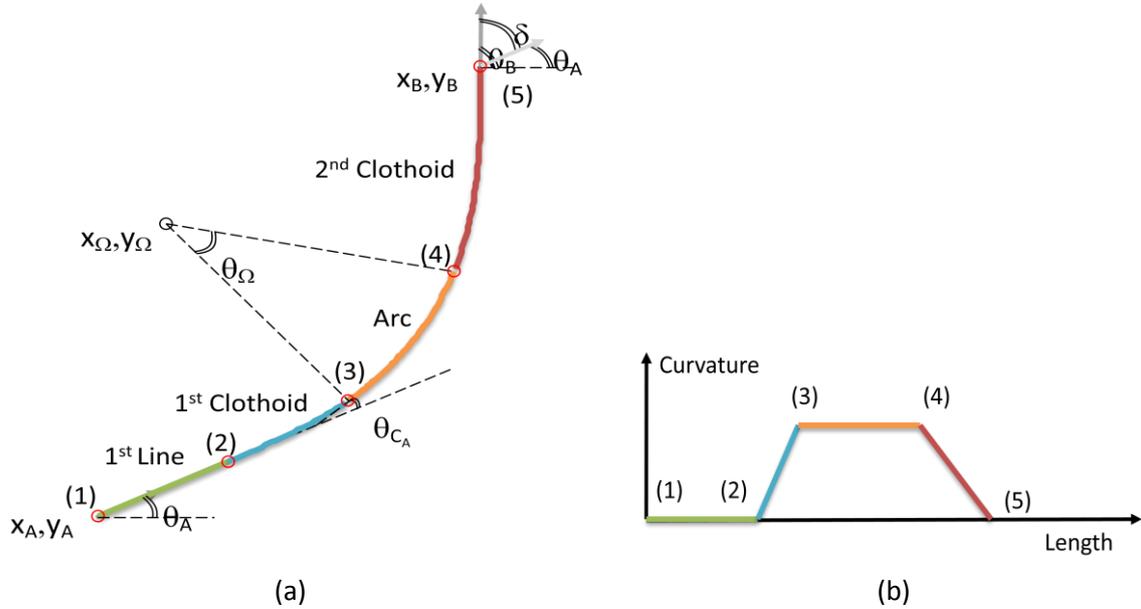


Figura 117. Ejemplo representativo de un *SCC-path*.

La trayectoria generada es muy similar a la introducida en [15], pero aquí las Clotoides no necesariamente deben ser simétricas. Además, un caso particular del *SCC-path* es cuando el arco circular y el segmento de recta tienen longitud nula, de modo que sólo se necesitan dos Clotoides, lo que se conoce como *Elementary path*.

En los *SCC-paths*, el ángulo del arco  $\theta_\Omega$  debe satisfacer la siguiente expresión:

$$\theta_\Omega = \theta_B - \theta_A - \theta_{C_A} - \theta_{C_B} \quad (127)$$

Siendo  $\theta_{C_A} \in ]0, \kappa_{max}^2 \sigma_{min}^{-1} / 2]$  el ángulo de tangencia de la primera Clotoide, con *sharpness*  $\sigma_{C_A}$  y curvatura máxima  $\kappa_{max}$ . Lo mismo ocurre para  $\theta_{C_B} \in ]0, \kappa_{max}^2 \sigma_{min}^{-1} / 2]$ , que es el ángulo de tangencia de la segunda Clotoide, con *sharpness*  $\sigma_{C_B}$  y curvatura máxima  $\kappa_{max}$ .

Además, para que sea necesario el uso de un arco circular los ángulos de tangencia de las dos Clotoides deben satisfacer que  $\theta_{C_A} + \theta_{C_B} < \delta = \theta_B - \theta_A$ . Por lo tanto, el máximo ángulo de deflexión que se puede girar sólo con dos Clotoides es:

$$\delta_{max} \equiv \kappa_{max}^2 \sigma_{min}^{-1} \quad (128)$$

Se asume que el *sharpness* de las Clotoides es conocido y que se obtiene de las siguientes expresiones:

$$\begin{aligned}\sigma_{C_A} &= \alpha_A \cdot (\sigma_{max} - \sigma_{min}) + \sigma_{min} \\ \sigma_{C_B} &= \alpha_B \cdot (\sigma_{max} - \sigma_{min}) + \sigma_{min}\end{aligned}\quad (129)$$

Siendo  $\alpha_A \in [0,1]$  y  $\alpha_B \in [0,1]$  dos parámetros de diseño.

Aunque las Clotoides no tienen que ser simétricas sí que se debe cumplir que en el punto de unión entre el círculo y las Clotoides las curvaturas sean iguales, de modo que se satisfaga la siguiente expresión:

$$\kappa_{\Omega} = \kappa_{C_A} = \kappa_{C_B} = \min\{\sqrt{\sigma_{C_A} \cdot \delta}, \sqrt{\sigma_{C_B} \cdot \delta}, \kappa_{max}\} \quad (130)$$

Variando el ángulo de deflexión se obtiene una familia de perfiles de curvatura para diferentes valores de  $\sigma$ , como se muestra en la Figura 118.

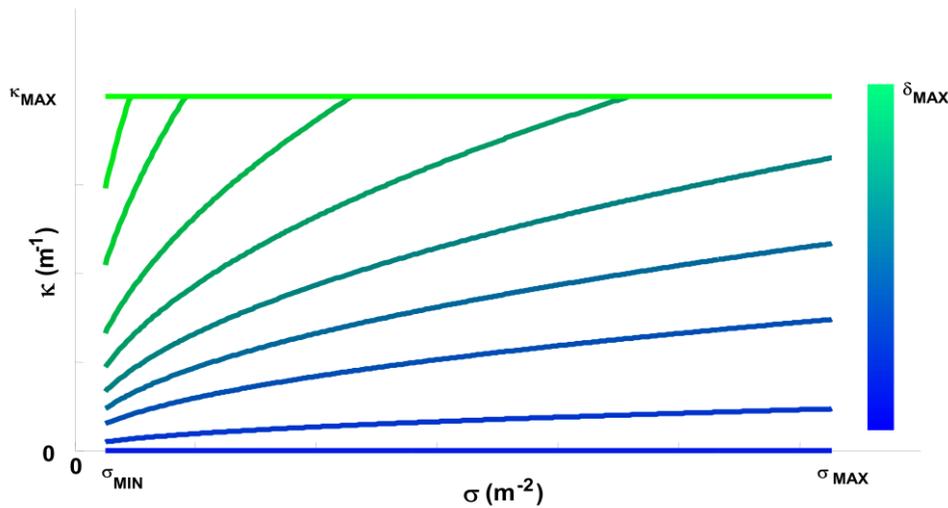


Figura 118. Selección de curvas de curvatura para distintos valores de  $\sigma$ .

Se puede demostrar que el arco circular tiene longitud nula si se cumple la siguiente desigualdad:

$$\sqrt{\sigma_{C_A} \cdot \delta} = \sqrt{\sigma_{C_B} \cdot \delta} < \kappa_{max} \quad (131)$$

En caso de no cumplirse la expresión anterior, el centro del círculo con radio  $r_{\Omega} = \kappa_{C_A}^{-1}$ , cuya referencia se sitúa en el origen de la primera Clotoide, sería el siguiente:

$$\begin{aligned}x_{\Omega} &= x_{C_A} - r_{\Omega} \sin \theta_{C_A} \\ y_{\Omega} &= y_{C_A} + r_{\Omega} \cos \theta_{C_A}\end{aligned}\quad (132)$$

Siendo  $q_{C_A} = (x_{C_A}, y_{C_A}, \theta_{C_A}, \kappa_{C_A})^T$  el punto final de la primera Clotoide, dado por las integrales de Fresnel con el siguiente límite de integración  $\gamma_{C_A} = \sqrt{\frac{\kappa_{C_A}^2}{\pi \cdot \sigma_{C_A}}}$ .

$$\begin{aligned} x_{C_A} &= \pm \sqrt{\frac{\pi}{\sigma_{C_A}}} \cdot \int_0^{\gamma_{C_A}} \cos \frac{\pi}{2} \xi^2 d\xi \\ y_{C_A} &= \pm \sqrt{\frac{\pi}{\sigma_{C_A}}} \cdot \int_0^{\gamma_{C_A}} \sin \frac{\pi}{2} \xi^2 d\xi \\ \theta_{C_A} &= \pm \frac{\kappa_{C_A}^2}{2 \cdot \sigma_{C_A}} \end{aligned} \quad (133)$$

Donde  $q_{C_B} = (x_{C_B}, y_{C_B}, \theta_{C_B}, \kappa_{C_B})^T$  se define de forma similar como:

$$\begin{aligned} x_{C_B} &= \pm \sqrt{\frac{\pi}{\sigma_{C_B}}} \cdot \int_0^{\gamma_{C_B}} \cos \frac{\pi}{2} \xi^2 d\xi \\ y_{C_B} &= \pm \sqrt{\frac{\pi}{\sigma_{C_B}}} \cdot \int_0^{\gamma_{C_B}} \sin \frac{\pi}{2} \xi^2 d\xi \\ \theta_{C_B} &= \pm \frac{\kappa_{C_B}^2}{2 \cdot \sigma_{C_B}} \end{aligned} \quad (134)$$

Con su propio límite de integración  $\gamma_{C_B} = \sqrt{\frac{\kappa_{C_B}^2}{\pi \cdot \sigma_{C_B}}}$ .

De las ecuaciones (133) y (134) se deduce que se puede operar en los cuatro cuadrantes y que puede darse cualquier combinación para la unión de dos Clotoides. La solución dependerá de las condiciones necesitadas, es decir, del signo de la curvatura y del signo de su derivada, tal y como muestra la Figura 119.

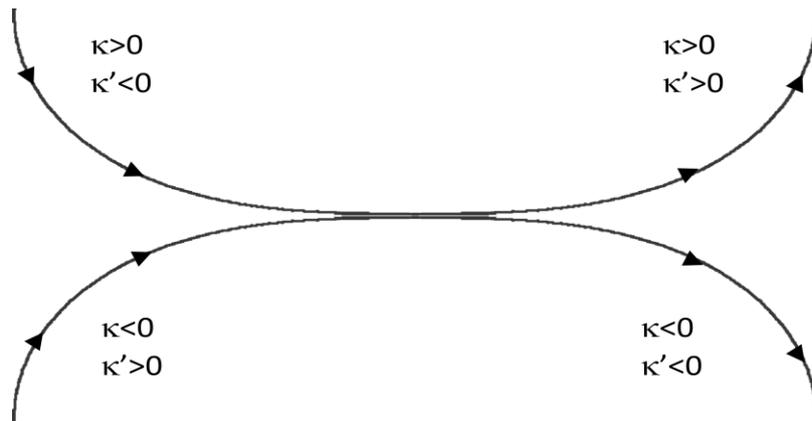


Figura 119. Paths de Clotoides en los cuatro cuadrantes.

La configuración objetivo  $q_B$  dependerá por tanto de los valores seleccionados de  $\alpha_A$  y  $\alpha_B$ , determinando tanto las curvaturas  $\kappa_{C_A}$  y  $\kappa_{C_B}$ , como los *sharpness*  $\sigma_{C_A}$  y  $\sigma_{C_B}$  de cada Clotoide.

$$q_B = \begin{pmatrix} x_A \\ y_A \\ \theta_A \\ \kappa_A \end{pmatrix} + \begin{pmatrix} l_A \cdot \cos\theta_A \\ l_A \cdot \sin\theta_A \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{R}(\theta_A) \cdot \begin{bmatrix} x_{C_A} \\ y_{C_A} \end{bmatrix} \\ \theta_{C_A} \\ \kappa_{C_A} \end{pmatrix} + \begin{pmatrix} r_\Omega \cdot \mathbf{R}(\theta_A + \theta_{C_A}) \cdot \begin{bmatrix} \sin\theta_\Omega \\ 1 - \cos\theta_\Omega \end{bmatrix} \\ \theta_\Omega \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{R}(\theta_B) \cdot \begin{bmatrix} x_{C_B} \\ -y_{C_B} \end{bmatrix} \\ \theta_{C_B} \\ -\kappa_{C_B} \end{pmatrix} \quad (135)$$

Siendo lo distintos operandos de izquierda a derecha y de arriba a abajo los siguientes: origen, segmento de recta, Clotoide A, arco de circunferencia y Clotoide B. Además,  $\mathbf{R}(\cdot)$  representa una matriz de rotación y  $l_A$  es la longitud del segmento de recta.

También hay que destacar que la curvatura del arco circular es  $\kappa_{C_A}$  y permanece constante, por eso en la expresión matricial anterior aparece como cero, puesto que no contribuye al valor final de curvatura.

Para el problema del seguimiento de línea es directa la obtención del valor del segmento de recta  $l_A$  necesario para converger a la línea con unos  $\alpha_A$  y  $\alpha_B$  dados.

$$l_A = \frac{X}{\cos\theta_A} \quad (136)$$

Siendo  $X$  la distancia horizontal recorrida por el *path* sin la inclusión del tramo recto. Dicho valor se puede obtener despejando de la primera fila de la expresión matricial de la ecuación (135).

$$X = x_B - x_A - x_{C_A} \cos\theta_A + y_{C_A} \sin\theta_A - 2 \cdot r_\Omega \cos(\theta_A + \theta_{C_A} + \frac{\theta_\Omega}{2}) \cdot \sin(\frac{\theta_\Omega}{2}) - y_{C_B} \quad (137)$$

Nótese que al fijar la orientación  $\theta_B = \frac{\pi}{2}$  se simplifica la ecuación (135), puesto que:

$$\mathbf{R}(\theta_B) \cdot \begin{bmatrix} x_{C_B} \\ -y_{C_B} \end{bmatrix} = \begin{bmatrix} x_{C_B} \cdot \cos\theta_B & y_{C_B} \cdot \sin\theta_B \\ x_{C_B} \cdot \sin\theta_B & -y_{C_B} \cdot \cos\theta_B \end{bmatrix} = \begin{bmatrix} 0 & y_{C_B} \\ x_{C_B} & 0 \end{bmatrix} \quad (138)$$

Del mismo modo se puede obtener la distancia vertical para la cual el *SCC-path* convergerá a la línea:

$$y_B = x_{C_A} \sin\theta_A + y_{C_A} \cos\theta_A + x_{C_B} + l_A \sin\theta_A + 2 \cdot r_\Omega \sin(\theta_A + \theta_{C_A} + \frac{\theta_\Omega}{2}) \cdot \sin(\frac{\theta_\Omega}{2}) \quad (139)$$

Como inconveniente hay que destacar que el *SSC-path* tiene una singularidad cuando las configuraciones inicial y final tienen la misma orientación, de modo que no hay solución, a menos que se generen bucles o *loops*, lo cual no es deseable para este tipo de aplicaciones. Esto es bastante obvio porque con un perfil de curvatura que no cambia el signo no es posible realizar una trayectoria con forma de "S" para converger a la línea. Con el objetivo de solucionar este problema se introduce el concepto de *Double Continuous-Cuvature path (DCC-path)*.

### 3.9.3. Double Continuous Curvature Path (DCC-path)

El *DCC-path* es la combinación de dos *SCC-paths* de modo que se obtengan un rango de soluciones más genérico que permita un cambio de signo del perfil de curvatura de la trayectoria generada. A continuación se detallará la notación que se va a considerar con tal de diferenciar los distintos elementos que forman el *DCC-path*.

El primer *SCC-path* se identifica con el subíndice A y empieza en el punto  $q_A$ , mientras que el segundo *SCC-path* va con subíndice B y finaliza en el punto  $q_B$ . Además, la configuración intermedia que une ambos *SCC-paths* es  $q_C = (x_C, y_C, \theta_C, \kappa_C)^T$ , con  $\kappa_C = 0$ .

La Figura 120 muestra un ejemplo de *DCC-path*, mientras que su perfil de curvatura aparece en la Figura 121.

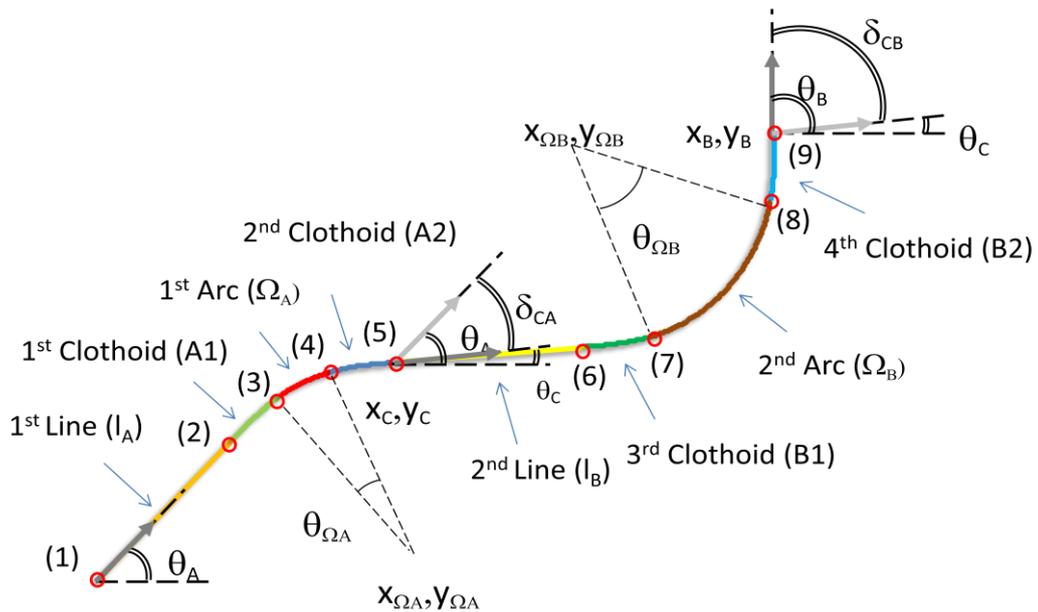


Figura 120. Ejemplo representativo de un *DCC-path*.

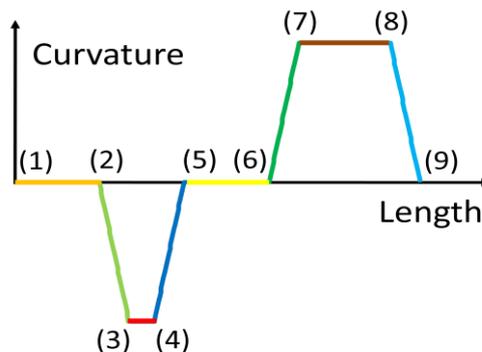


Figura 121. Perfil de curvatura del *DCC-path* de la Figura 120.

En las figuras anteriores se puede apreciar que para garantizar los cambios de curvatura de forma adecuada se necesitan: cuatro Clotoides denominadas A1, A2, B1 y B2; dos arcos circulares  $\Omega_A$  y  $\Omega_B$ ; así como dos segmentos de línea con longitud  $l_A$  y  $l_B$ .

Al igual que en el caso de los *SCC-paths*, para el *DCC-path* también hay una solución particular para la que sólo se necesitan cuatro Clotoides, obteniendo soluciones similares a las de un *BiElementary path*. Sin embargo, es interesante remarcar que la generalización de la formulación para los *DCC-paths* nos permite generar un espectro completo de posibles trayectorias, utilizando Clotoides con distintos *sharpness* de forma que se combinen indistintamente para llegar a la configuración deseada.

Se asume que el *sharpness* de las Clotoides es conocido y se obtiene de las siguientes expresiones:

$$\begin{aligned}\sigma_{C_{A1}} &= \alpha_{A1} \cdot (\sigma_{max} - \sigma_{min}) + \sigma_{min} \\ \sigma_{C_{A2}} &= \alpha_{A2} \cdot (\sigma_{max} - \sigma_{min}) + \sigma_{min} \\ \sigma_{C_{B1}} &= \alpha_{B1} \cdot (\sigma_{max} - \sigma_{min}) + \sigma_{min} \\ \sigma_{C_{B2}} &= \alpha_{B2} \cdot (\sigma_{max} - \sigma_{min}) + \sigma_{min}\end{aligned}\tag{140}$$

Siendo  $\alpha_{Ai} \in [0,1]$  y  $\alpha_{Bi} \in [0,1]$  los parámetros de diseño.

La máxima curvatura para cada par de Clotoides es:

$$\begin{aligned}\kappa_{\Omega_A} = \kappa_{C_{A1}} = \kappa_{C_{A2}} &= \min\{\sqrt{\sigma_{C_{A1}} \cdot \delta_A}, \sqrt{\sigma_{C_{A2}} \cdot \delta_A}, \kappa_{max}\} \\ \kappa_{\Omega_B} = \kappa_{C_{B1}} = \kappa_{C_{B2}} &= \min\{\sqrt{\sigma_{C_{B1}} \cdot \delta_B}, \sqrt{\sigma_{C_{B2}} \cdot \delta_B}, \kappa_{max}\}\end{aligned}\tag{141}$$

Siendo  $\delta_A = |\theta_A - \theta_C|$  y  $\delta_B = |\theta_B - \theta_C|$  los ángulos de deflexión entre las configuraciones  $\angle(q_A, q_C)$  y  $\angle(q_C, q_B)$ .

Sin pérdida de generalidad se considera  $x_A = 0, y_A = 0, x_B > x_A$  y  $\theta_B = \pi/2$ . Además, si  $\theta_A > \theta_B$  implica que  $\delta_A > \delta_B$  y por tanto  $\theta_C$  se debería acotar entre  $\theta_C \in [\theta_A - \frac{\pi}{2}, \theta_B]$ . En cambio, si  $\theta_A \leq \theta_B$  implica que  $\delta_A \leq \delta_B$  y en consecuencia  $\theta_C \in [\theta_B - \frac{\pi}{2}, \theta_A]$ . El límite inferior de  $\theta_C$  se selecciona para obtener ángulos de deflexión  $\delta < \pi$ , mientras que el límite superior asegura que el *DCC-path* converge a la línea, evitando los casos singulares.

Se incluye un parámetro de diseño  $\alpha_C$  para determina la orientación de la configuración intermedia:

$$\theta_C = \begin{cases} \alpha_C \cdot (\theta_B - \theta_A + \frac{\pi}{2}) + \theta_A + \frac{\pi}{2} \rightarrow Si \theta_A > \theta_B \\ \alpha_C \cdot (\theta_A - \theta_B + \frac{\pi}{2}) + \theta_B + \frac{\pi}{2} \rightarrow Si \theta_A \leq \theta_B \end{cases}\tag{142}$$

La configuración intermedia  $q_C$  dependerá por tanto de los valores seleccionados de  $\alpha_{A1}$  y  $\alpha_{A2}$ , determinando tanto las curvaturas  $\kappa_{C_{A1}}$  y  $\kappa_{C_{A2}}$ , como los *sharpness*  $\sigma_{C_{A1}}$  y  $\sigma_{C_{A2}}$  de cada Clotoide.

$$\begin{aligned}q_C &= \begin{pmatrix} x_A \\ y_A \\ \theta_A \\ \kappa_A \end{pmatrix} + \begin{pmatrix} l_A \cdot \cos\theta_A \\ l_A \cdot \sin\theta_A \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{R}(\theta_A) \cdot \begin{bmatrix} x_{C_{A1}} \\ y_{C_{A1}} \end{bmatrix} \\ -\theta_{C_{A1}} \\ -\kappa_{C_{A1}} \end{pmatrix} + \\ &+ \begin{pmatrix} r_{\Omega_A} \cdot \mathbf{R}(\theta_A - \theta_{C_{A1}}) \cdot \begin{bmatrix} \sin\theta_{\Omega_A} \\ \cos\theta_{\Omega_A} - 1 \end{bmatrix} \\ -\theta_{\Omega_A} \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{R}(\theta_C) \cdot \begin{bmatrix} x_{C_{A2}} \\ y_{C_{A2}} \end{bmatrix} \\ -\theta_{C_{A2}} \\ \kappa_{C_{A2}} \end{pmatrix}\end{aligned}\tag{143}$$

Mientras que la configuración final  $q_B$  dependerá por tanto de los valores seleccionados de  $\alpha_{B1}$  y  $\alpha_{B2}$ , determinando tanto las curvaturas  $\kappa_{C_{B1}}$  y  $\kappa_{C_{B2}}$ , como los *sharpness*  $\sigma_{C_{B1}}$  y  $\sigma_{C_{B2}}$  de cada Clotoide.

$$q_B = \begin{pmatrix} x_C \\ y_C \\ \theta_C \\ \kappa_C \end{pmatrix} + \begin{pmatrix} l_B \cdot \cos \theta_C \\ l_B \cdot \sin \theta_C \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{R}(\theta_C) \cdot \begin{bmatrix} x_{C_{B1}} \\ y_{C_{B1}} \end{bmatrix} \\ \theta_{C_{B1}} \\ \kappa_{C_{B1}} \end{pmatrix} + \begin{pmatrix} r_{\Omega_B} \cdot \mathbf{R}(\theta_C + \theta_{C_{B1}}) \cdot \begin{bmatrix} \sin \theta_{\Omega_B} \\ 1 - \cos \theta_{\Omega_B} \end{bmatrix} \\ \theta_{\Omega_B} \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{R}(\theta_B) \cdot \begin{bmatrix} x_{C_{B2}} \\ -y_{C_{B2}} \end{bmatrix} \\ \theta_{C_{B2}} \\ -\kappa_{C_{B2}} \end{pmatrix} \quad (144)$$

Con  $x_{C_{A1}}$ ,  $y_{C_{A1}}$ ,  $\theta_{C_{A1}}$ ,  $x_{C_{A2}}$ ,  $y_{C_{A2}}$ ,  $\theta_{C_{A2}}$ ,  $x_{C_{B1}}$ ,  $y_{C_{B1}}$ ,  $\theta_{C_{B1}}$ ,  $x_{C_{B2}}$ ,  $y_{C_{B2}}$  y  $\theta_{C_{B2}}$ , similares a los definidos en las ecuaciones (133) y (134). Además, para cada arco circular se tiene que el radio es  $r_{\Omega_A} = \kappa_{\Omega_A}^{-1}$  y  $r_{\Omega_B} = \kappa_{\Omega_B}^{-1}$ , y los ángulos  $\theta_{\Omega_A} = \theta_C - \theta_A - \theta_{C_{A1}} - \theta_{C_{A2}}$  y  $\theta_{\Omega_B} = \theta_B - \theta_C - \theta_{C_{B1}} - \theta_{C_{B2}}$ , respectivamente.

La formulación de las ecuaciones (143) y (144) se ha realizado para el caso con un perfil de curvatura como el mostrado en la Figura 120. No obstante se puede generalizar para cualquier combinación, como por ejemplo para situaciones en las que el signo de los dos *SCC-paths* no cambie pese a tener cuatro Clotoides.

Para converger a la línea se debe satisfacer la siguiente ecuación:

$$X = l_A \cdot \cos \theta_A + l_B \cdot \cos \theta_C \quad (145)$$

Siendo  $X$  la distancia horizontal recorrida por el *path* sin la inclusión de los tramos rectos, que son los valores que se están buscando.

Nótese que al fijar la orientación  $\theta_B = \frac{\pi}{2}$  se realiza una simplificación en la ecuación, puesto que:

$$\mathbf{R}(\theta_B) \cdot \begin{bmatrix} x_{C_{B2}} \\ -y_{C_{B2}} \end{bmatrix} = \begin{bmatrix} x_{C_{B2}} \cdot \cos \theta_B & y_{C_{B2}} \cdot \sin \theta_B \\ x_{C_{B2}} \cdot \sin \theta_B & -y_{C_{B2}} \cdot \cos \theta_B \end{bmatrix} = \begin{bmatrix} 0 & y_{C_{B2}} \\ x_{C_{B2}} & 0 \end{bmatrix} \quad (146)$$

No obstante no ocurre lo mismo para la configuración  $\theta_C$  puesto que es arbitraria:

$$\mathbf{R}(\theta_C) \cdot \begin{bmatrix} x_{C_{A2}} \\ y_{C_{A2}} \end{bmatrix} = \begin{bmatrix} x_{C_{A2}} \cdot \cos \theta_C & -y_{C_{A2}} \cdot \sin \theta_C \\ x_{C_{A2}} \cdot \sin \theta_C & y_{C_{A2}} \cdot \cos \theta_C \end{bmatrix} \quad (147)$$

La variable  $X$  buscada se puede obtener despejando la primera fila de la expresión matricial de las ecuaciones (143) y (144).

$$X = x_B - x_A - x_{C_{A1}} \cos \theta_A + y_{C_{A1}} \sin \theta_A - 2 \cdot r_{\Omega_A} \cos \left( \theta_A - \theta_{C_{A1}} + \frac{\theta_{\Omega_A}}{2} \right) \cdot \sin \left( \frac{\theta_{\Omega_A}}{2} \right) - x_{C_{A2}} \cos \theta_C + y_{C_{A2}} \sin \theta_C - x_{C_{B1}} \cos \theta_C + y_{C_{B1}} \sin \theta_C - 2 \cdot r_{\Omega_B} \cos \left( \theta_C + \theta_{C_{B1}} + \frac{\theta_{\Omega_B}}{2} \right) \cdot \sin \left( \frac{\theta_{\Omega_B}}{2} \right) - y_{C_{B2}} \quad (148)$$

Por lo tanto, al no tener una solución única cerrada se pueden obtener múltiples soluciones que satisfagan esta relación. A continuación se detalla el criterio seguido para seleccionar  $l_A$  y  $l_B$ .

- Seleccionar valores positivos (si es posible) para  $l_A$  o  $l_B$ , dependiendo de los signos de  $X$ ,  $\cos \theta_A$  y  $\cos \theta_C$ , forzando una de las longitudes a cero. Esta situación puede suceder cuando los signos de  $X$ ,  $\cos \theta_A$  y  $\cos \theta_C$  son distintos, o  $\cos \theta_A = 0$  y  $\cos \theta_C \neq 0$ , o  $\cos \theta_A \neq 0$  y  $\cos \theta_C = 0$ . Por eso, por ejemplo, si  $X > 0$  y  $\cos \theta_A > 0$ , pero  $\cos \theta_C < 0$ , entonces se fuerza  $l_B = 0$ , siendo  $l_A > 0$ .
- Puede haber situaciones en las que  $l_A > 0$  y  $l_B > 0$  porque los signos de  $X$ ,  $\cos \theta_A$  y  $\cos \theta_C$  son iguales. En esos casos, se fuerza uno de los valores a cero, intentando converger a la línea con el mínimo  $y_B$ . Por lo tanto:
  - Si  $|\cos \theta_A| > |\cos \theta_C|$  entonces  $l_A \neq 0$  y  $l_B = 0$ .
  - Si  $|\cos \theta_A| < |\cos \theta_C|$  entonces  $l_A = 0$  y  $l_B \neq 0$ .

Hay que remarcar que puede darse una situación singular si los cosenos de ambos ángulos son nulos  $\cos \theta_A = 0$  y  $\cos \theta_C = 0$ , lo cual significa que las tres configuraciones  $q_A$ ,  $q_C$  y  $q_B$  tienen la misma orientación. Pero dicha singularidad puede evitarse con una selección apropiada de la orientación de la configuración  $q_C$ .

### 3.9.4. Análisis

En primer lugar se van a analizar los distintos casos que pueden darse en la generación de trayectorias de tipo *DCC-path*. En las Figuras 122 a 125 aparecen algunos ejemplos de *DCC-paths* que trabajan en los cuatro cuadrantes de curvatura y su derivada (ver Figura 119). Es evidente que de ahí se pueden extraer el resto de combinaciones con tan sólo rotar y trasladar el origen del *DCC-path*.

En la Figura 122 (a) se presenta la situación de seguimiento de línea con  $d_h > 0$ ,  $\delta > 0$ ,  $\kappa_A > 0$  y  $\kappa_B > 0$ . Como se observa se trata de una situación sencilla en la que para converger a la línea no es necesario realizar cambios de signo de curvatura. En la Figura 122 (b) se observa el perfil de curvatura de la trayectoria generada.

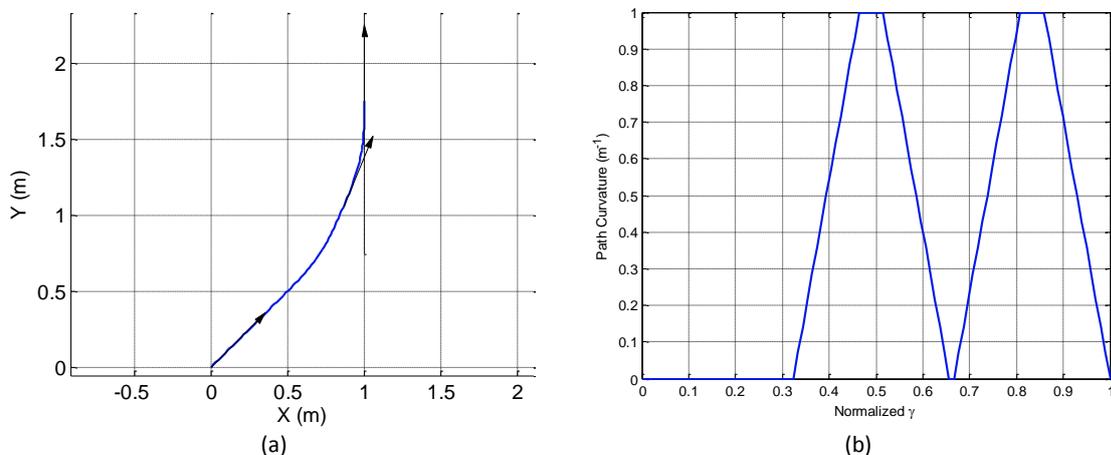


Figura 122. Seguimiento de línea con  $d_h > 0$ ,  $\delta > 0$ ,  $\kappa_A > 0$  y  $\kappa_B > 0$ .

En la Figura 123 (a) se modifica ligeramente la situación anterior, de modo que se tiene  $d_h > 0$ ,  $\delta > 0$ ,  $\kappa_A < 0$  y  $\kappa_B > 0$ . Como se observa se trata de una situación similar, pero ahora para converger a la línea sí se realiza un cambio en el signo de la curvatura. En la Figura 123 (b) se observa el perfil de curvatura de la trayectoria generada.

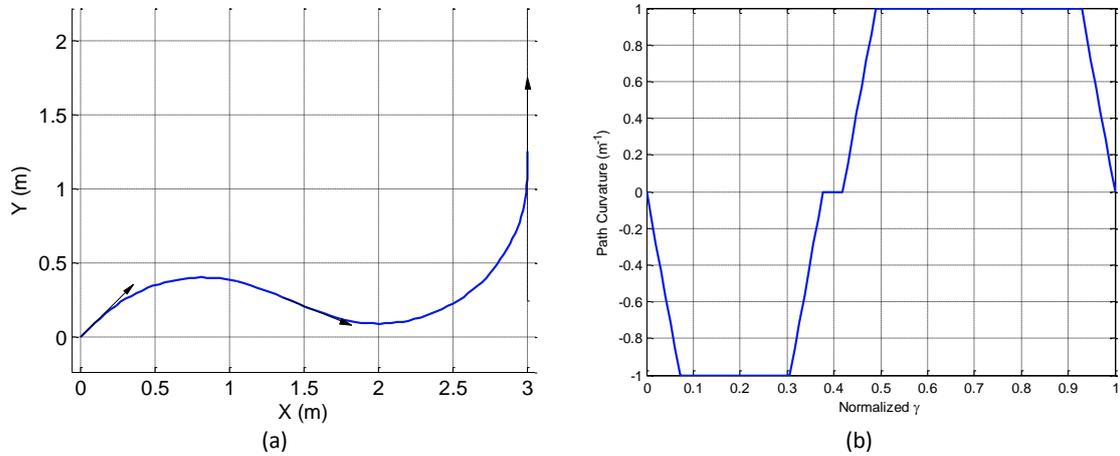


Figura 123. Seguimiento de línea con  $d_h > 0$ ,  $\delta > 0$ ,  $\kappa_A < 0$  y  $\kappa_B > 0$ .

En la Figura 124 (a) se presenta la situación simétrica a la de la Figura 122, con  $d_h < 0$ ,  $\delta < 0$ ,  $\kappa_A < 0$  y  $\kappa_B < 0$ . Como se observa se trata de la misma situación pero desde el lado opuesto de la línea. En la Figura 124 (b) se observa el perfil de curvatura de la trayectoria generada.

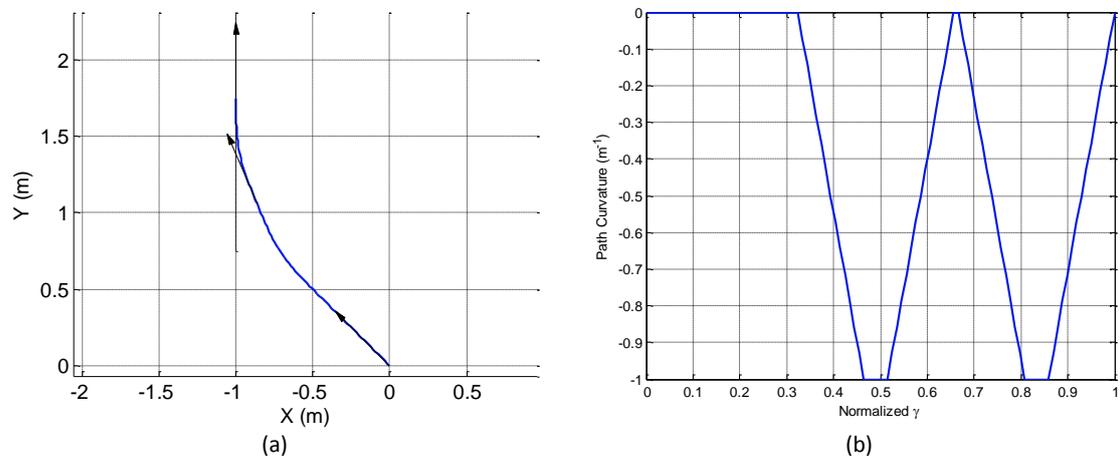


Figura 124. Seguimiento de línea con  $d_h < 0$ ,  $\delta < 0$ ,  $\kappa_A < 0$  y  $\kappa_B < 0$ .

En la Figura 125 (a) se presenta la situación simétrica a la de la Figura 123, con  $d_h < 0$ ,  $\delta < 0$ ,  $\kappa_A > 0$  y  $\kappa_B < 0$ . Como se observa se trata de la misma situación pero desde el lado opuesto de la línea y con los signos de las curvaturas cambiados para ambos *SCC-paths*. En la Figura 125 (b) se observa el perfil de curvatura de la trayectoria generada.

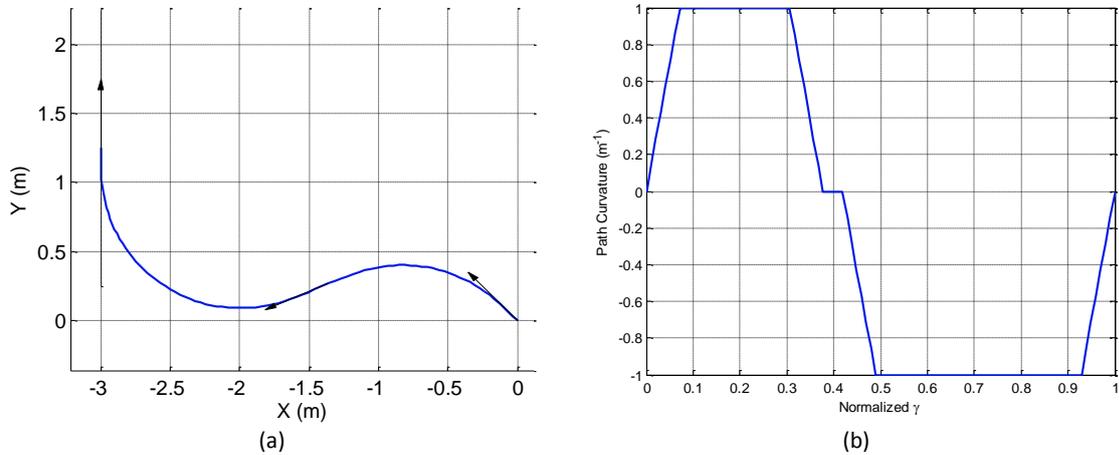


Figura 125. Seguimiento de línea con  $d_h < 0$ ,  $\delta < 0$ ,  $\kappa_A > 0$  y  $\kappa_B < 0$ .

Hasta el momento se han analizado los casos genéricos más habituales en los que se combinaban distancias, ángulos de deflexión y curvaturas tanto positivos como negativos. No obstante, dado que la técnica propuesta es de una herramienta de diseño, hay situaciones que pueden suceder y que también es interesante analizar.

Puede darse el caso de que la distancia  $d_h$  sea tan pequeña que sea necesario realizar marcha atrás para converger a la línea, debido a que por la imposición de signos de las curvaturas no se pueda hacer otra cosa. Eso es lo que ocurre en la Figura 126 (a), que a pesar de ser similar al caso de la Figura 123, puesto que se tiene  $d_h > 0$ ,  $\delta > 0$ ,  $\kappa_A < 0$  y  $\kappa_B > 0$ , en realidad la restricción de tener  $d_h$  de valor pequeño hace que se tenga una longitud  $l_A < 0$ . La Figura 126 (b) muestra el perfil de curvatura de la trayectoria generada en el que se observa un primer tramo de curvatura nula correspondiente a la trayectoria rectilínea.

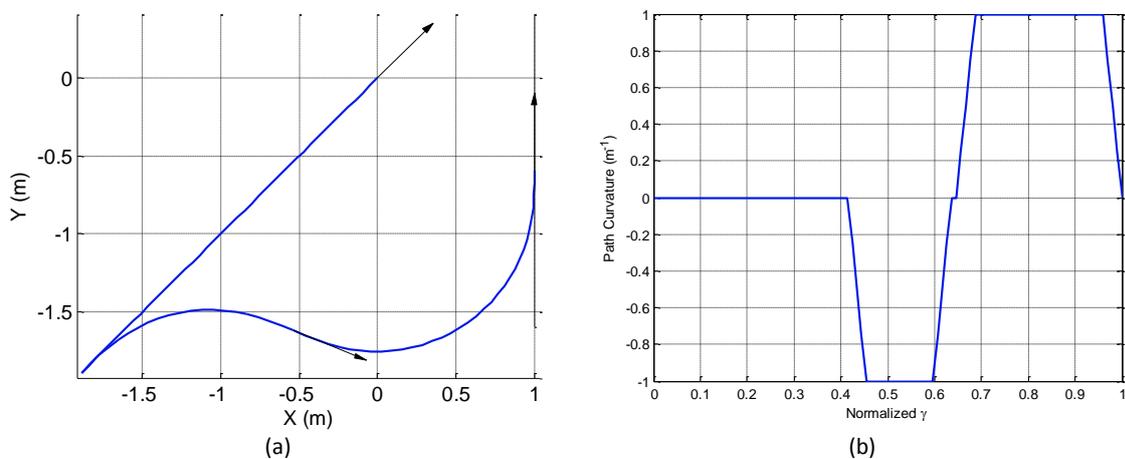


Figura 126. Seguimiento de línea con  $d_h > 0$  pero de valor pequeño,  $\delta > 0$ ,  $\kappa_A < 0$  y  $\kappa_B > 0$ .

Otro caso interesante es el de la Figura 127 (a), es decir, cuando se tiene  $d_h \leq 0$ ,  $\delta = 0$ ,  $\kappa_A < 0$  y  $\kappa_B > 0$ , o la situación simétrica. En este caso singular se obtiene una longitud  $l_c < 0$  con tal de contrarrestar las limitaciones de giro y que el sistema sea capaz de converger a la línea. La Figura 127 (b) muestra el perfil de curvatura de la trayectoria generada en el que se observa un tramo central de curvatura nula correspondiente a la trayectoria rectilínea.

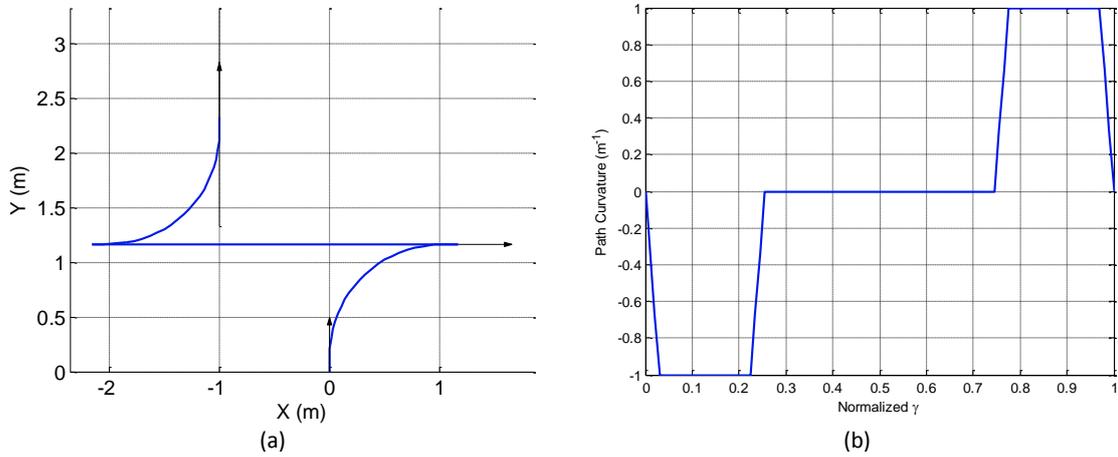


Figura 127. Seguimiento de línea con  $d_h < 0$ ,  $\delta = 0$ ,  $\kappa_A < 0$  y  $\kappa_B > 0$ .

El tercer caso que se analizará es el de la Figura 128 (a), ya que aunque en el transcurso del seguimiento de trayectoria es una situación que normalmente no se dará, sí que es posible en la primera fase del seguimiento, cuando el vehículo empieza su movimiento. En esta situación se tiene  $d_h > 0$ ,  $\delta < 0$ ,  $\kappa_A > 0$  y  $\kappa_B > 0$ . Como se puede ver, dadas las restricciones de curvaturas positivas, el vehículo se ve obligado a girar en sentido contrario al que sería habitual. El perfil de curvatura de la trayectoria generada se presenta en la Figura 128 (b).

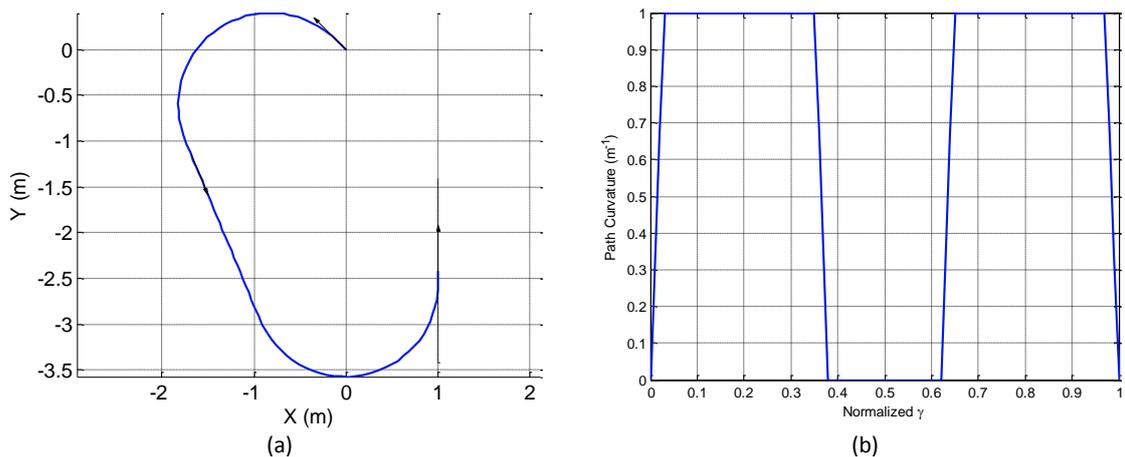


Figura 128. Seguimiento de línea con  $d_h > 0$ ,  $\delta < 0$ ,  $\kappa_A > 0$  y  $\kappa_B > 0$ .

Si se fuerza más la situación, haciendo que la distancia sea negativa  $d_h < 0$  y manteniendo el resto de variables igual, se llega a la necesidad de realizar un bucle o *loop* para converger a la línea (ver Figura 129).

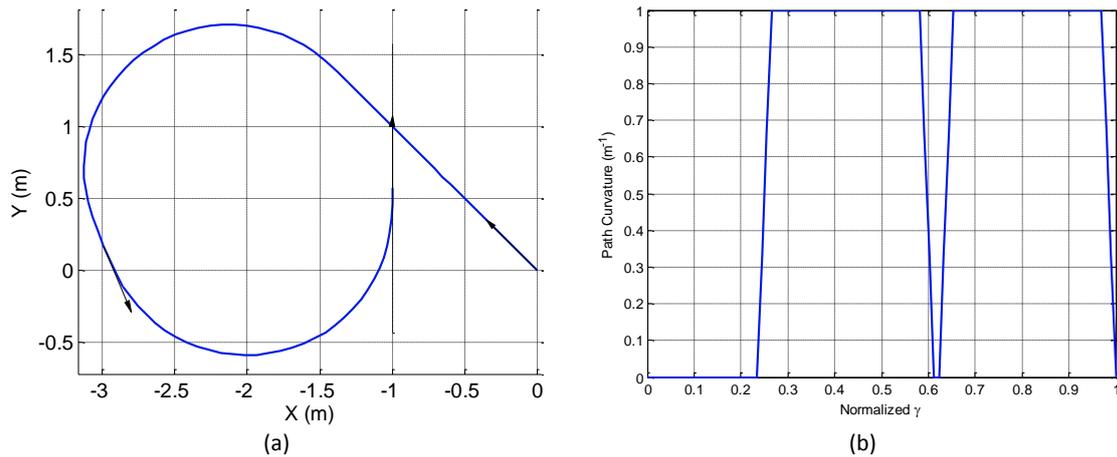


Figura 129. Seguimiento de línea con  $d_h < 0$ ,  $\delta < 0$ ,  $\kappa_A > 0$  y  $\kappa_B > 0$ .

Este tipo de trayectorias no son deseables en el problema de seguimiento de línea con robots móviles, pero puede darse el caso que por restricciones físicas o de diseño no se pueda realizar una trayectoria simple y lógica, debiendo recurrir a maniobras más complejas.

Como se ha podido observar el valor de los parámetros de diseño del *DCC-path* determina el *sharpness* de las Clotoides y en consecuencia la forma de la trayectoria generada. Además, la combinación seleccionada de *sharpness* de las Clotoides no solo afecta a la configuración final  $q_B$ , sino también a la curvatura máxima, a la longitud y también a las derivadas de la trayectoria (aceleración normal y  *jerk* tangencial y normal). Por ese motivo, ahora nos centraremos en el análisis del efecto de dichos parámetros de diseño en el resto de variables cinemáticas y dinámicas.

Para el análisis el ángulo de deflexión se fija a cero  $\delta = 0$ , es decir, las configuraciones inicial y final tendrán la misma orientación. Pero la configuración intermedia  $q_C$  sí que introducirá un ángulo de deflexión  $\delta_{AC} = \frac{\pi}{2}$ , lo que implica tener  $\theta_C = 0$  con  $\alpha_C = 0.5$ . Aunque los resultados pueden cambiar dependiendo de las configuraciones inicial e intermedia, se considera que este caso es uno de los más representativos. También hay que remarcar que el análisis se he realizado utilizando una velocidad constante del vehículo.

Los resultados del análisis demuestran que la aceleración normal  $a_N$ , el  *jerk* tangencial  $j_T$  y el  *jerk* normal  $j_N$  aumentan a medida que aumenta el *sharpness*, tal y como muestran las Figuras 130, 131 y 132, respectivamente. En estas figuras se representa una superficie 3D que relaciona cada una de las variables con los parámetros de diseño  $\alpha_A$  y  $\alpha_B$  comprendidos en el rango  $\alpha_i \in [0,1]$ .

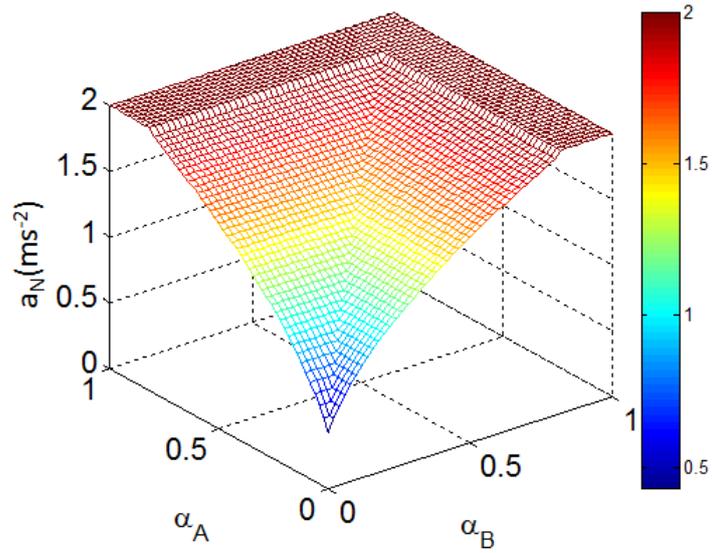


Figura 130. Máximo valor de aceleración normal  $a_N$  para distintos valores de  $\alpha_A$  y  $\alpha_B$ .

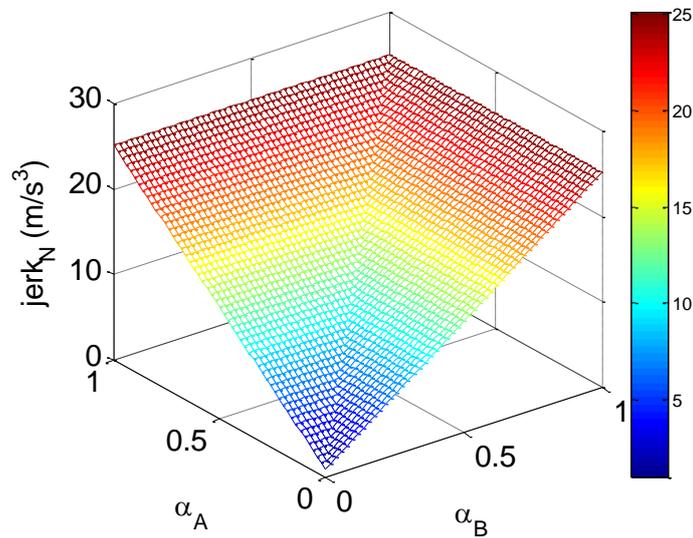


Figura 131. Máximo valor de *jerk* normal  $j_N$  para distintos valores de  $\alpha_A$  y  $\alpha_B$ .

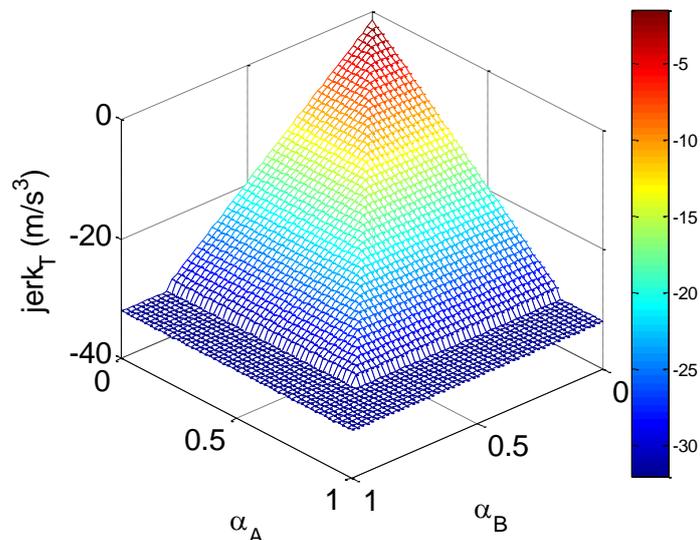


Figura 132. Máximo valor de *jerk* tangencial  $j_T$  para distintos valores de  $\alpha_A$  y  $\alpha_B$ .

De hecho, si se analizan las expresiones que definen cada una de las variables se obtiene lo siguiente:

- La aceleración normal es una función raíz cuadrada con respecto al *sharpness* porque  $a_N = v^2 \cdot \kappa^3 = v^2 \cdot \min\{\sqrt{\sigma_A \delta}, \sqrt{\sigma_B \delta}, \kappa_{max}\}$ .
- El *jerk* tangencial es lineal porque  $j_T = -v^3 \cdot \kappa^2 = v^3 \cdot \min\{\sigma_A \delta, \sigma_B \delta, \kappa_{max}^2\}$ .
- El *jerk* normal es lineal porque  $j_N = v^3 \cdot \max\{\sigma_A, \sigma_B\}$ .

Por otro lado, en la Figura 133 se analiza la longitud total del *path* con respecto a los parámetros de diseño que regulan el *sharpness*, es decir,  $\alpha_A$  y  $\alpha_B$ . En este caso se restringe cada par de Clotoides para que sean simétricas y así facilitar la interpretación de los resultados, aunque en el caso más genérico no es necesario.

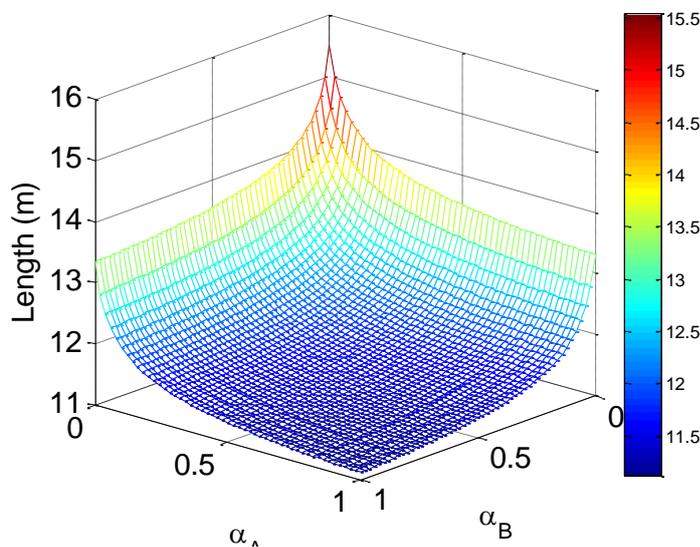


Figura 133. Longitud del *path* para distintos valores de  $\alpha_A$  y  $\alpha_B$ .

Como se esperaba, la longitud de la trayectoria disminuye a medida que aumenta el *sharpness*, debido a la relación entre ambas variables:

$$L = \frac{\kappa}{\sigma} = \frac{\min(\sqrt{\sigma \delta}, \kappa_{max})}{\sigma} \quad (149)$$

Finalmente se analizará el efecto del valor del parámetro de diseño  $\alpha_C$  sobre la trayectoria generada. En este caso las configuraciones inicial y final siguen teniendo la misma orientación, pero ahora los parámetros  $\alpha_A$  y  $\alpha_B$  permanecen constantes.

En la Figura 134 se observa la variación de la longitud del *DCC-path* frente a la variación del parámetro  $\alpha_C$ . Se observa que el caso de longitud mínima sucede para  $\alpha_C = 0.5$ , lo que significa que la configuración intermedia es perpendicular respecto a la línea y a la configuración inicial. Esto se debe a que dicha situación corresponde al caso de ángulo de deflexión mínimo.

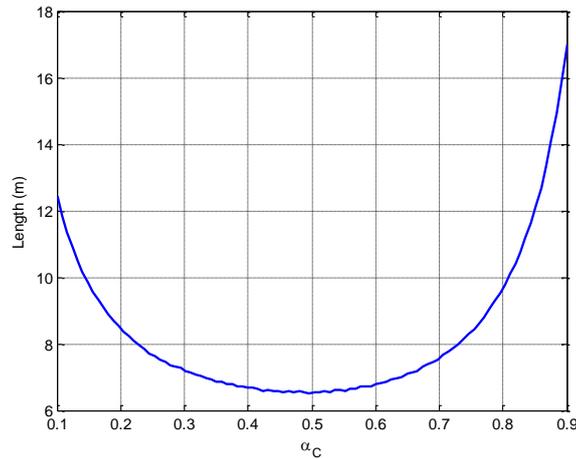


Figura 134. Variación de la longitud del *DCC-path* frente a la variación del parámetro  $\alpha_C$ .

Por otro lado, la Figura 135 muestra la variación de la curvatura máxima respecto al parámetro de diseño  $\alpha_C$ . Se trata de una función raíz cuadrada, aunque  $\alpha_C$  afecta linealmente al ángulo de deflexión.

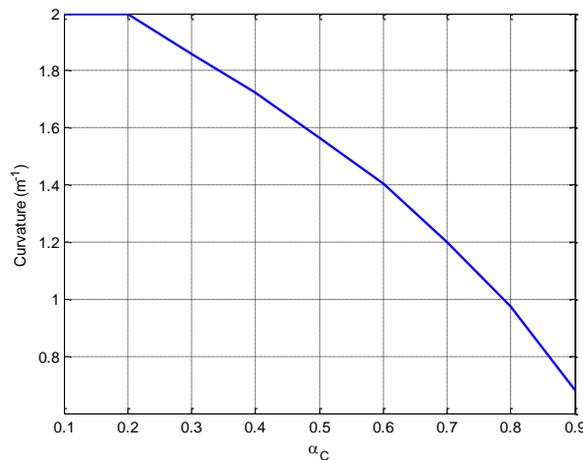


Figura 135. Variación de la curvatura del *DCC-path* frente a la variación del parámetro  $\alpha_C$ .

### 3.9.5. Conclusiones

En este apartado se ha descrito un método para la generación de trayectorias de curvatura continua o *CC-paths*, sujetos a restricciones en el *sharpness* de la curva y en la máxima curvatura alcanzable. Primero se ha descrito el *Single Continuous-Curvature Path (SCC-path)* que garantiza un perfil de curvatura continuo y consiste en una secuencia de elementos formada por: un segmento de línea, una curva de transición de tipo Clotoide, un arco de circunferencia y una segunda Clotoide. Después se ha extendido el *SCC-path* obteniendo el *Double Continuous-Curvature Path (DCC-path)*, que permite un mayor abanico de soluciones y perfiles de curvatura, con posibilidad de tener valores positivos y negativos de la misma en un mismo *path*.

Se ha demostrado que ambas trayectorias pueden aplicarse para resolver el problema del seguimiento de línea, de modo que los robots móviles con ruedas se ven beneficiados por su uso con una reducción del deslizamiento y en consecuencia de los errores de odometría.

Finalmente se ha analizado el efecto de los parámetros de diseño que fijan el *sharpness* de las Clotoides y la orientación de la configuración intermedia. Tras el análisis se ha demostrado que la longitud, la curvatura y las derivadas de la trayectoria están afectadas directamente por los parámetros de diseño.

### ***3.10. Técnica para el Control Cinemático On-line mediante la Generación de Trayectorias de Curvatura Continua, para el Problema de Seguimiento de Línea con Curvatura y Brusquedad Acotadas***

Como se ha comentado diversas veces a lo largo de la lectura de la presente Tesina, existen diversas técnicas de control cinemático que garantizan estabilidad y convergencia, capaces de dotar al robot de la suficiente autonomía como para seguir una línea o un camino. Entre dichas técnicas cabe mencionar tanto las de seguimiento de trayectorias mediante punto descentralizado, como las de seguimiento de caminos o persecución pura (*Pure Pursuit*). El problema de estas técnicas es que no garantizan continuidad en la curvatura, puesto que la acción de control se obtiene aplicando una ley de control que depende del error entre la posición actual y la referencia a seguir, sin tener en cuenta la posición y la orientación actual de las ruedas. Además tampoco se genera una trayectoria o secuencia de acciones de control sino que se obtiene una acción de control única que se recalcula cada iteración. Por lo tanto, la implementación de este tipo de técnicas provoca que el robot tenga ligeros deslizamientos en las ruedas incrementándose los errores de odometría, los cuales en muchas aplicaciones son inaceptables o poco deseables.

Las técnicas anteriores tampoco tienen en cuenta la variación de la curvatura generada en el control, ni tampoco las restricciones cinemáticas y dinámicas del vehículo autónomo. Como ya se sabe, entre las principales restricciones cinemáticas destaca la limitación física que tiene un robot en el ángulo de giro de su rueda o ruedas directrices. Por otro lado, entre las principales limitaciones dinámicas (aceleraciones y sus derivadas) están las impuestas por motivos de seguridad, por ejemplo en vehículos industriales para que estos no vuelquen cuando están cargados, o de confortabilidad en vehículos que transportan pasajeros.

En el apartado anterior se ha presentado una técnica de generación de trayectorias de curvatura continua considerando restricciones cinemáticas y dinámicas para el problema del seguimiento de líneas con robots móviles, pero en ocasiones esto no es suficiente, ya que puede darse una situación en la que sea necesario re-planificar la trayectoria para evitar por ejemplo una colisión ante un obstáculo. La técnica anterior no es suficiente y no garantiza que para cualquier configuración inicial se converja a la línea, puesto que se impone que las curvaturas inicial y final sean nulas. Por ese motivo, en este apartado se presentará una técnica capaz de generalizar la generación de trayectorias de curvatura continua con restricciones, de modo que sea aplicable para el control cinemático de un robot móvil, con requerimientos de tiempo real para la planificación *on-line* de la trayectoria.

### 3.10.1. Continuous Curvature Path Control

Para el problema de generación de *Continuous Curvature Paths* se necesitan cuatro Clotoides denominadas A1, A2, B1 y B2; dos arcos circulares  $\Omega_A$  y  $\Omega_B$ ; así como dos segmentos de línea con longitud  $l_A$  y  $l_B$ . No obstante en la fase de control, para asegurar que el vehículo llegue exactamente al objetivo deseado se añade un tercer segmento de recta  $l_C$  al *DCC-path* (ver Figuras 136 y 137). Nótese que se ha renombrado  $l_B$  como  $l_C$ , llamando al nuevo segmento  $l_B$ .

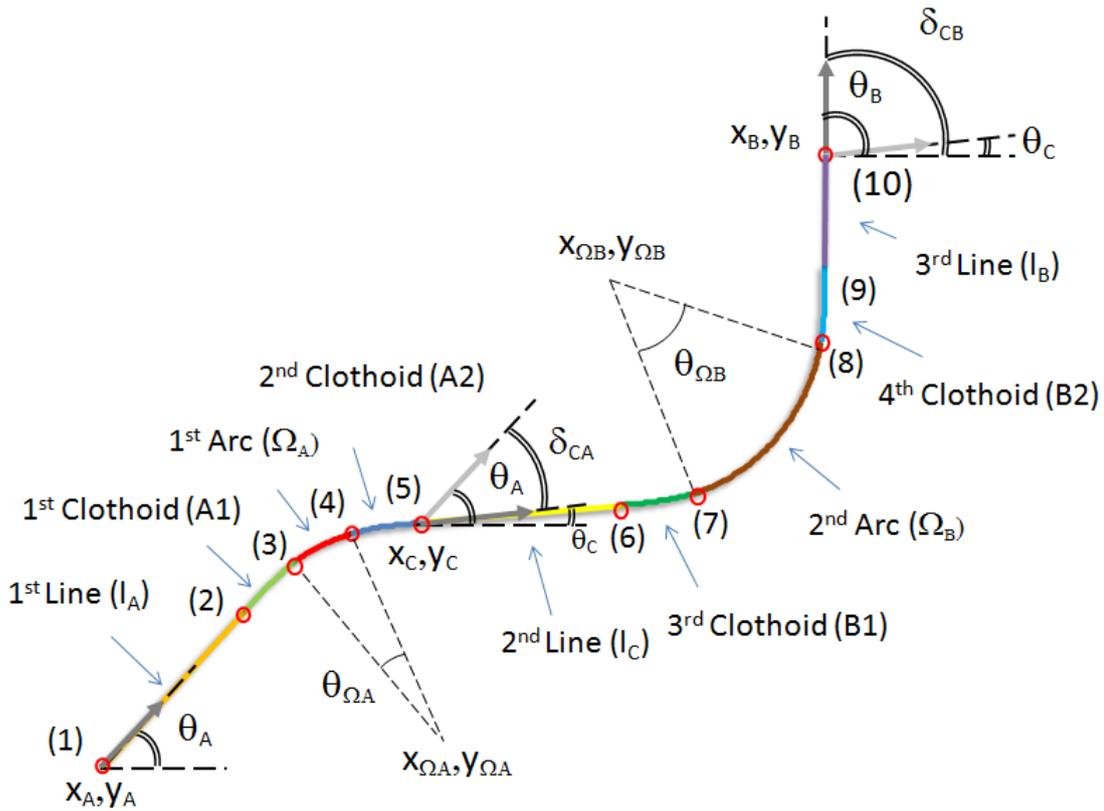


Figura 136. Ejemplo de *DCC-path* para control.

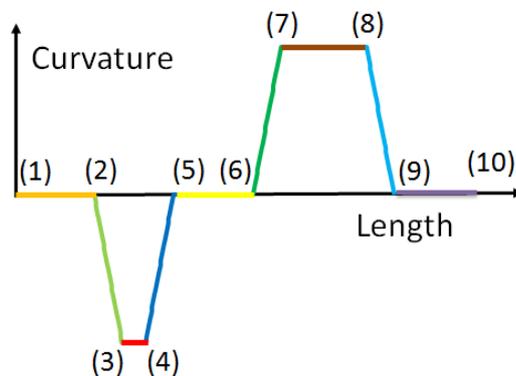


Figura 137. Perfil de curvatura del *DCC-path* de la Figura 136.

Como se ve en la Figura 136, el nuevo segmento se introduce al final del segundo *SCC-path*, de modo que las ecuaciones de éste cambian, pero el primer *SCC-path* no se ve afectado.

$$q_B = \begin{pmatrix} x_C \\ y_C \\ \theta_C \\ \kappa_C \end{pmatrix} + \begin{pmatrix} l_C \cdot \cos \theta_C \\ l_C \cdot \sin \theta_C \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} l_B \cdot \cos \theta_B \\ l_B \cdot \sin \theta_B \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{R}(\theta_C) \cdot \begin{bmatrix} x_{CB1} \\ y_{CB1} \end{bmatrix} \\ \theta_{CB1} \\ \kappa_{CB1} \\ 0 \end{pmatrix} + \begin{pmatrix} r_{\Omega_B} \cdot \mathbf{R}(\theta_C + \theta_{CB1}) \cdot \begin{bmatrix} \sin \theta_{\Omega_B} \\ 1 - \cos \theta_{\Omega_B} \end{bmatrix} \\ \theta_{\Omega_B} \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{R}(\theta_B) \cdot \begin{bmatrix} x_{CB2} \\ -y_{CB2} \end{bmatrix} \\ \theta_{CB2} \\ -\kappa_{CB2} \\ 0 \end{pmatrix} \quad (150)$$

Ahora el objetivo no es converger a la línea para una separación determinada  $d_h$ , sin importar la coordenada  $y_B$  para la cual se alcanza. El problema que se aborda implica llegar exactamente a la configuración deseada  $q_B = (x_B, y_B, \theta_B, \kappa_B)^T$ . Por lo tanto, no se debe satisfacer únicamente una ecuación sino dos, tal y como se observa a continuación:

$$\begin{aligned} X(\theta_C) &= l_A \cdot \cos \theta_A + l_B \cdot \cos \theta_B + l_C \cdot \cos \theta_C \\ Y(\theta_C) &= l_A \cdot \sin \theta_A + l_B \cdot \sin \theta_B + l_C \cdot \sin \theta_C \end{aligned} \quad (151)$$

Siendo  $X(\theta_C)$  e  $Y(\theta_C)$  la distancia horizontal y vertical respectivamente, recorrida por el *path* sin la inclusión de los tramos rectos, que son los valores que se están buscando, junto a la orientación intermedia  $\theta_C$  que garantice que hay solución factible. Por lo tanto el objetivo es obtener unos valores adecuados de  $l_A, l_B, l_C$  y  $\theta_C$  que garanticen que se llega a la postura  $q_B$ .

Al igual que para el caso de generación, para el problema de control también es necesario obtener la expresión que relaciona las variables desconocidas con las que son conocidas:

$$\begin{bmatrix} X(\theta_C) \\ Y(\theta_C) \end{bmatrix} = q_B - q_A - A1 - \Omega_A - A2 - B1 - \Omega_B - B2 \quad (152)$$

Como puede deducirse se trata de un problema sin una solución cerrada puesto que hay dos ecuaciones y cuatro incógnitas. Por eso se establece un criterio heurístico para determinar la solución que minimice la distancia global del camino generado. Con dicho criterio se calcula una orientación intermedia:

$$\theta_C^* = \arg \min_{\theta_C} L = l_A + l_{A1} + l_{\Omega_A} + l_{A2} + l_C + l_{B1} + l_{\Omega_B} + l_{B2} + l_B \quad (153)$$

Sujeto a la ecuación (152), donde las longitudes de las Clotoides se obtienen a partir de las propiedades de las mismas:

$$\begin{aligned} l_{A1} &= \kappa_{A1}(\theta_C) \cdot \sigma_{A1}^{-1} \\ l_{A2} &= \kappa_{A2}(\theta_C) \cdot \sigma_{A2}^{-1} \\ l_{B1} &= \kappa_{B1}(\theta_C) \cdot \sigma_{B1}^{-1} \\ l_{B2} &= \kappa_{B2}(\theta_C) \cdot \sigma_{B2}^{-1} \end{aligned} \quad (154)$$

Y las longitudes de los arcos a partir de la ecuación de la longitud de una circunferencia:

$$\begin{aligned} l_{\Omega_A} &= \Omega_A(\theta_C) \cdot \kappa_{A1}^{-1}(\theta_C) \\ l_{\Omega_B} &= \Omega_B(\theta_C) \cdot \kappa_{B1}^{-1}(\theta_C) \end{aligned} \quad (155)$$

Para resolver el problema se fuerza uno de los segmentos para que tenga longitud nula y poder entonces resolver de forma más sencilla el problema de minimización:

- En general, será interesante tener  $l_B = 0$ , siempre que  $\theta_C \neq \theta_A$ .
- En caso contrario se produce una situación singular que se puede evitar forzando  $l_B = 0$  si  $\theta_A \neq \theta_B$ .
- Si  $\theta_A = \theta_B = \theta_C$ , no se puede obtener una solución válida por tratarse del caso singular más desfavorable, por eso, para evitar caer en dicha situación se fija la longitud del *path* a un valor  $L = \infty$ , para descartar este tipo de soluciones durante el proceso de minimización.

Por otro lado, como ya se había avanzado anteriormente, para extender de forma completa el problema del seguimiento de líneas se debe considerar el caso en el que la curvatura actual del vehículo es distinta de cero. En dicho caso, se asume que el vehículo está situado en un punto dado de la primera Clotoide A1 del *DCC-path*, cuyo origen debe ser determinado puesto que a priori es desconocido. También la curvatura de la configuración final debe considerarse no nula en el caso genérico. Por lo tanto, debe calcularse el origen (o punto final) de la segunda Clotoide del *SCC-path* B.

Si los segmentos de Clotoide se conocen a priori, se pueden computar sus puntos  $x_{C_{A1}}^S$ ,  $y_{C_{A1}}^S$ ,  $x_{C_{B2}}^E$  e  $y_{C_{B2}}^E$ , así como sus respectivas curvaturas  $\kappa_{C_{A1}}^S$  y  $\kappa_{C_{B2}}^E$  y sus ángulos de tangencia en los puntos extremos  $\theta_{C_{A1}}^S$  y  $\theta_{C_{B2}}^E$ . Por lo tanto es posible obtener el origen de la primera y la cuarta Clotoide:

$$q'_A = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} + \begin{pmatrix} R(\theta(t) \pm \theta_{C_{A1}}^S) \begin{bmatrix} \pm x_{C_{A1}}^S \\ \pm y_{C_{A1}}^S \end{bmatrix} \\ \theta(t) \pm \theta_{C_{A1}}^S \\ \kappa_{C_{A1}}^S \end{pmatrix} \quad (156)$$

$$q'_B = \begin{pmatrix} x_B \\ y_B \end{pmatrix} + \begin{pmatrix} R(\theta_B \pm \theta_{C_{B2}}^E) \begin{bmatrix} \pm x_{C_{B2}}^E \\ \pm y_{C_{B2}}^E \end{bmatrix} \\ \theta_B \pm \theta_{C_{B2}}^E \\ \kappa_{C_{B2}}^E \end{pmatrix} \quad (157)$$

Dónde  $x$ ,  $y$  y  $\theta$  corresponden a la pose actual del vehículo y  $q'_A$  y  $q'_B$  son los origen de las Clotoides que deben sustituir a  $q_A$  y  $q_B$  en las ecuaciones (143) y (144). Esto se puede ver gráficamente de forma más detallada en la Figura 138.

En este caso, cuando  $\kappa_{C_{A1}}^S \neq 0$  la longitud del primer y tercer segmento de línea se fijan a  $l_A = 0$  y  $l_B = 0$ , mientras que la longitud del segmento intermedio y su orientación se pueden determinar numéricamente mediante el Método de Punto Fijo (*Fixed Point Method*) a partir de las ecuaciones (152) y (153).

$$\theta_C = \arctan \frac{Y(\theta_C)}{X(\theta_C)} \quad (158)$$

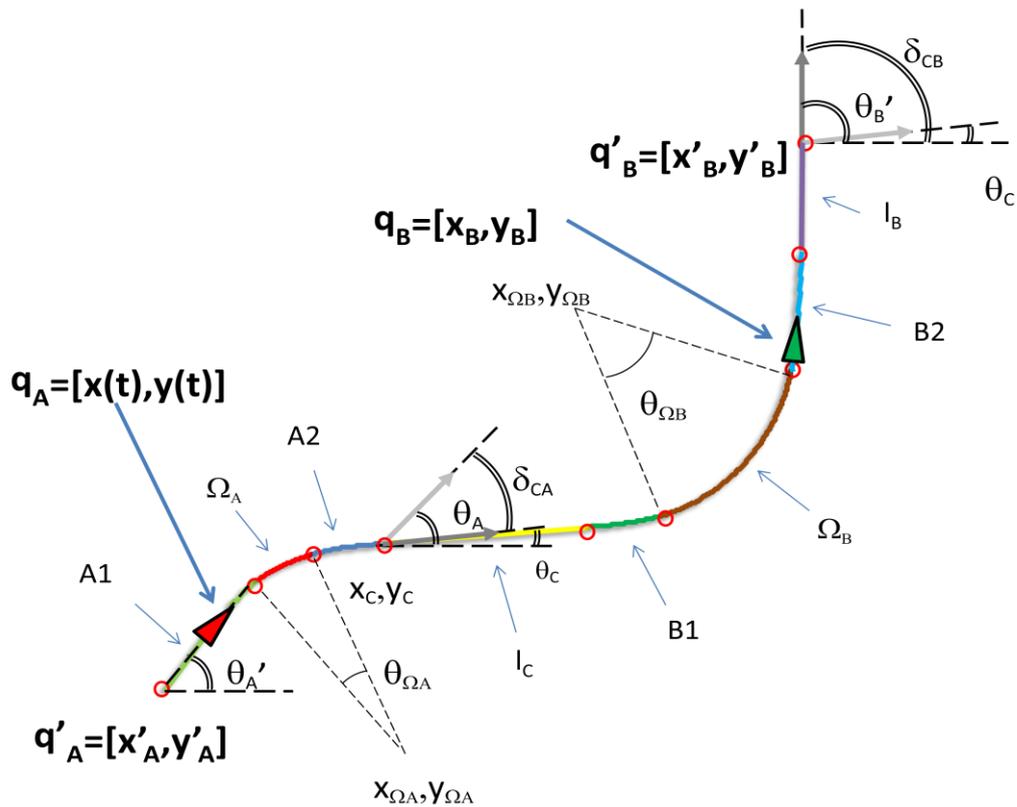


Figura 138. Ejemplo de *DCC-path* en el problema de control para configuraciones inicial  $q_A$  y final  $q_B$  arbitrarias.

### 3.10.2. Análisis

Una vez se ha computado el *DCC-path* para la configuración actual del robot y para un punto objetivo dado, hay que aplicar la curvatura y el *sharpness* calculado durante un cierto tiempo  $T$ , siendo  $T$  el periodo utilizado en el control cinemático. Por eso, sólo los primeros  $T$  segundos de la trayectoria planificada se tendrán en cuenta antes del siguiente periodo de control, despreciándose el resto de la trayectoria por no ser necesaria para el control inmediato. Sin embargo, es necesario computar la trayectoria completa porque la solución está altamente acoplada y pequeños cambios en la solución afectarán a la forma y a la factibilidad del *DCC-path* generado.

En el método de persecución pura (*Pure Pursuit Method*) se determina una configuración objetivo a partir de la obtención de un punto separado a cierta distancia  $LA$  (*Look Ahead*) de la pose actual del robot. Las mismas ideas se han utilizado para resolver el problema del seguimiento de líneas, como por ejemplo el cambio de carril o el adelantamiento en carreteras. No obstante, aquí no se abordarán este tipo de problemas de forma específica como un objetivo a conseguir, sino que el objetivo es ofrecer una solución o herramienta genérica que sea capaz de resolver estos problemas entre otros muchos.

En la Figura 139 se muestran algunos ejemplos de seguimiento de caminos basados en métodos de persecución pura con distintas configuraciones iniciales y para varios valores de *sharpness*, con  $LA = 10m$ . Basado en la orientación inicial se generan cuatro *paths* típicos diferentes que muestran la flexibilidad del método de generación propuesto y sus propiedades de convergencia a la línea.

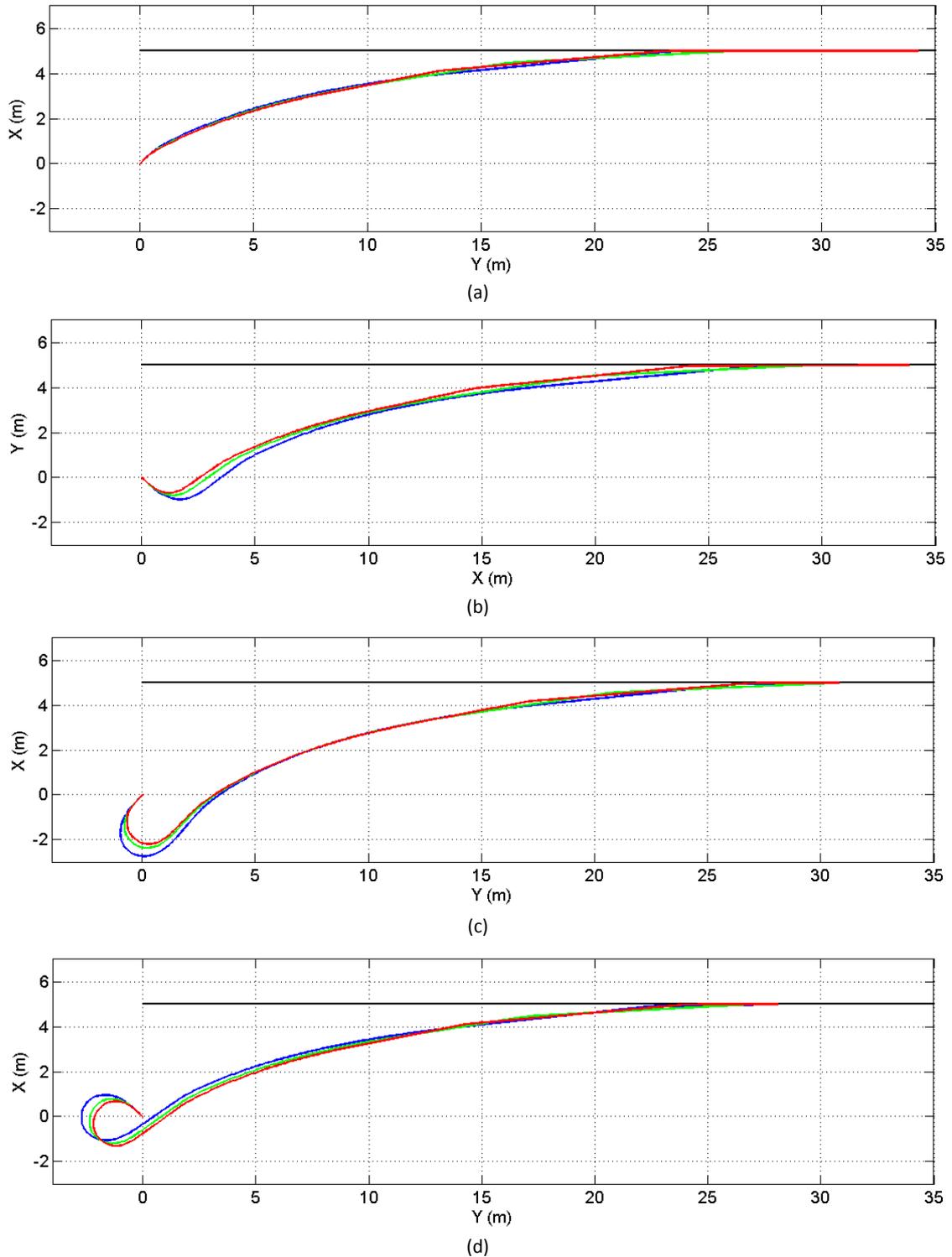


Figura 139. Ejemplos de seguimiento de caminos basados en métodos de persecución pura con distintas configuraciones iniciales y para varios valores de *sharpness*, con  $D=10m$ .

Además, en las Figuras 140 a 143 aparecen el *sharpness*, curvatura y ángulo de tangencia para cada caso, con tal de mostrar mejor el tipo de acciones aplicadas, de forma que se aprecien los perfiles continuos de la curvatura, puesto que es el objetivo principal de esta técnica de generación y control de trayectorias continuas.

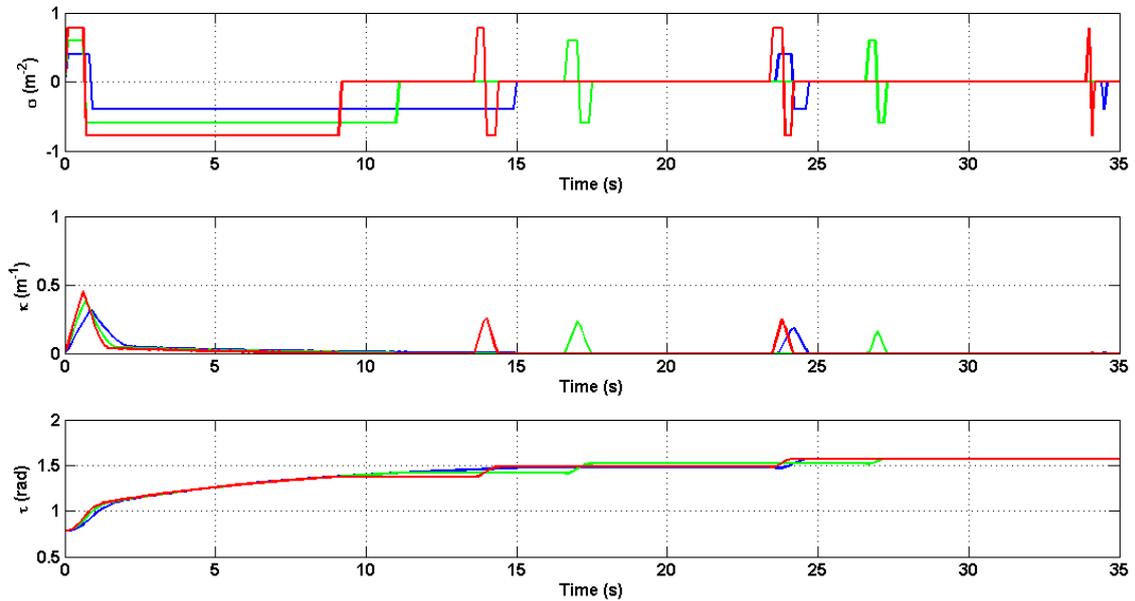


Figura 140. *Sharpness*, curvatura y ángulo de tangencia para las trayectorias de la Figura 139 (a).

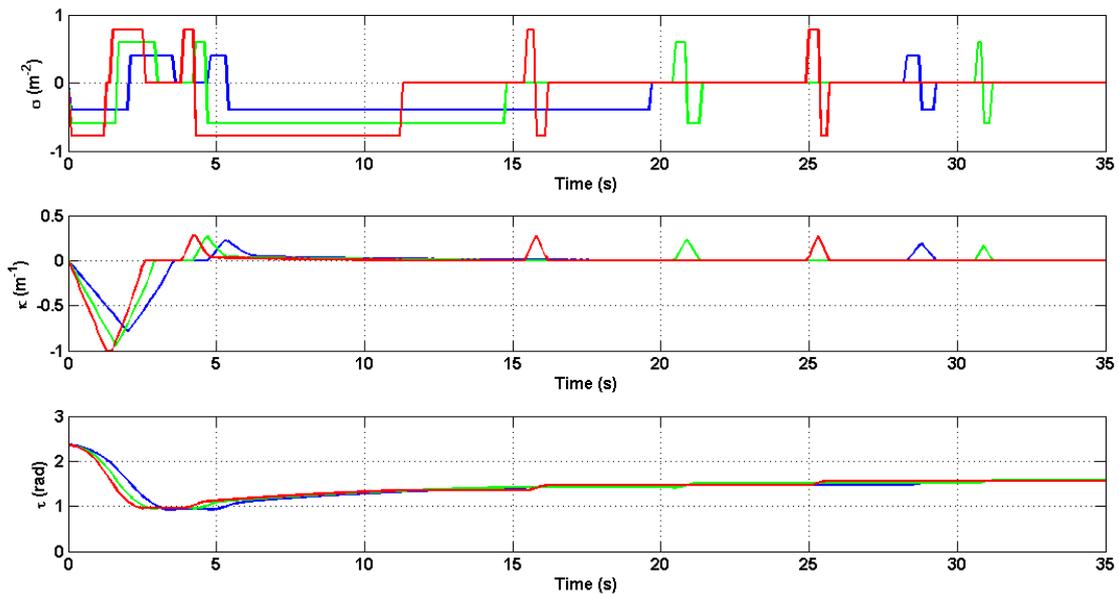


Figura 141. *Sharpness*, curvatura y ángulo de tangencia para las trayectorias de la Figura 139 (b).

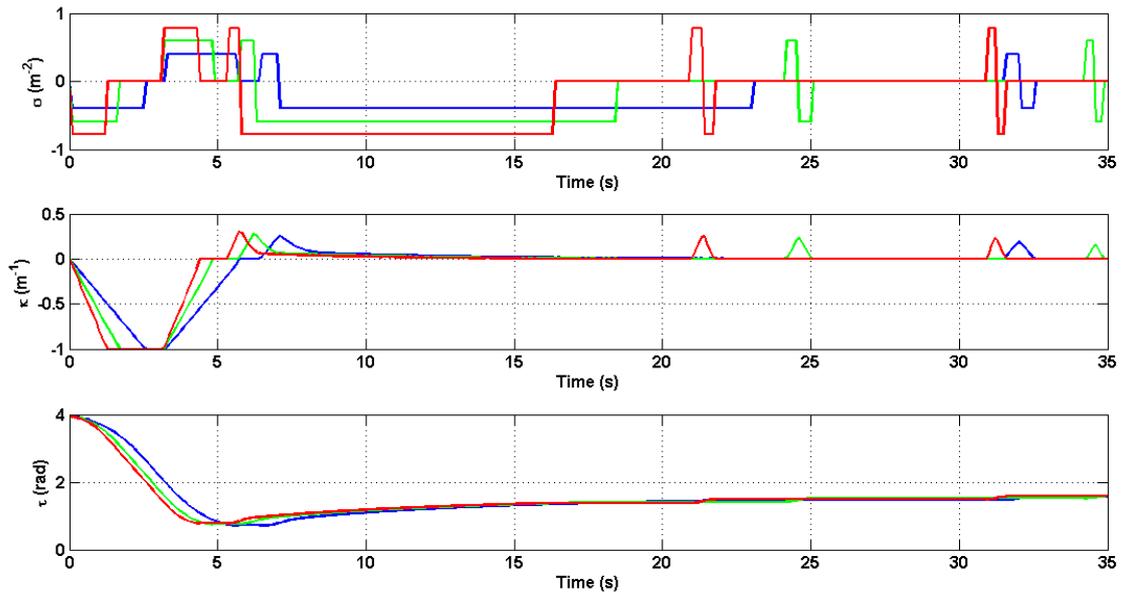


Figura 142. *Sharpness*, curvatura y ángulo de tangencia para las trayectorias de la Figura 139 (c).

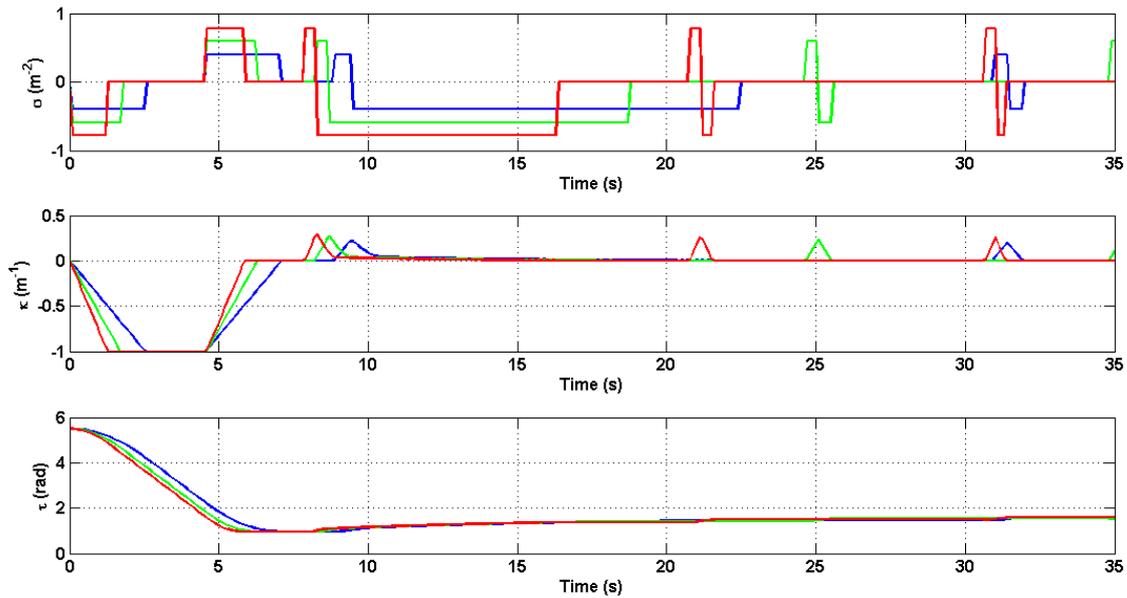


Figura 143. *Sharpness*, curvatura y ángulo de tangencia para las trayectorias de la Figura 139 (d).

Hay que remarcar que el *sharpness* es la derivada de la curvatura y por tanto no necesita ser continuo. También interesa comentar que, tal y como muestran las gráficas de las figuras anteriores, se han considerado trayectorias con diferentes *sharpness* aunque no se han detectado diferencias considerables entre las distintas trazas generadas por el vehículo, lo cual es un dato importante a favor del método propuesto, ya que demuestra que es un sistema bastante robusto.

Las ideas del seguimiento de líneas se pueden extender para generar trayectorias que normalmente se utilizan en una conducción estándar. Por ejemplo, en la Figura 144, se muestran dos ejemplos simples de cambio de carril (a) y de adelantamiento (b), utilizando la técnica propuesta.

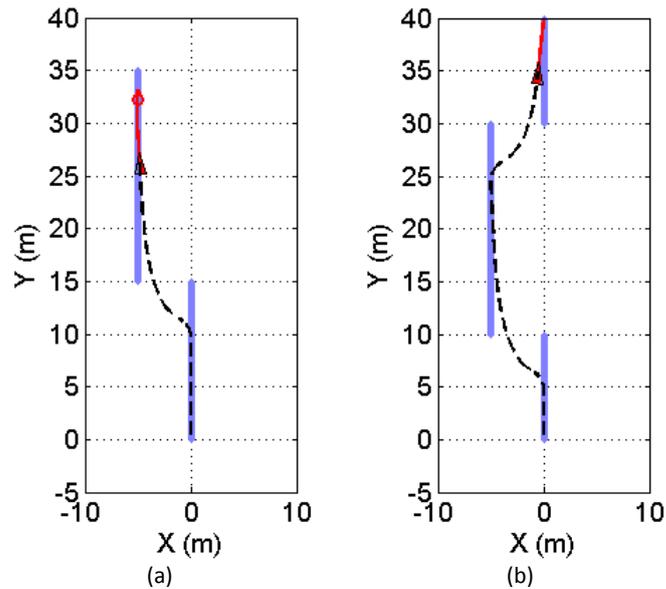


Figura 144. Maniobras típicas de conducción resueltas con *DCC-paths*: (a) cambio de carril y (b) adelantamiento.

Por último se dirá que el controlador cinemático se puede utilizar en combinación con métodos de *path planning* que proporcionen un set de puntos objetivo o *waypoints*. No es objetivo de la presente Tesina determinar dichos puntos, pero sí realizar el control cinemático de bajo nivel para alcanzarlos. Para ello se puede abordar el problema de dos formas distintas: la primera implicaría utilizar el conjunto de puntos como configuraciones objetivo con una orientación determinada y curvatura nula; mientras que la segunda se basaría en técnicas de persecución pura mediante la definición de segmentos de línea entre dos *waypoints* consecutivos. Gracias a eso los puntos objetivo generados por el planificador global se pueden aproximar mediante caminos suaves satisfaciendo las restricciones de curvatura y *sharpness*. En la Figura 145 se muestra un ejemplo de camino generado mediante cinco *waypoints* con la técnica de control propuesta.

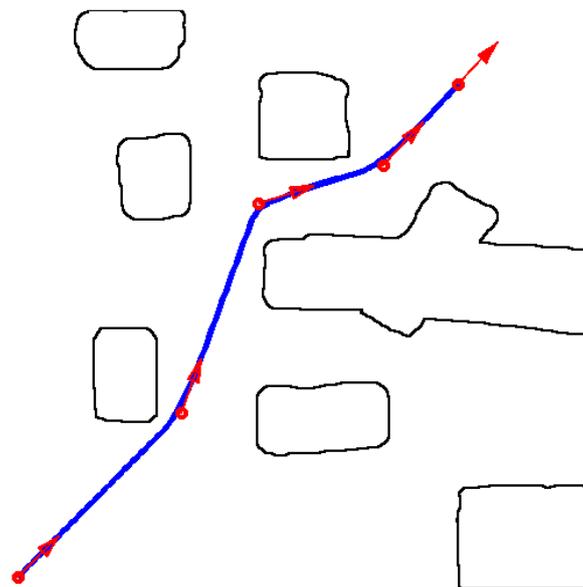


Figura 145. Ejemplo de trayectoria generada mediante control con *DCC-paths* en un entorno con obstáculos.

### 3.10.3. Conclusiones

En este apartado se describe un método de generación de trayectorias de curvatura continua sujeta a restricciones de curvatura y *sharpness* máximos, para su posterior aplicación al control cinemático de un robot en el problema del seguimiento de líneas.

La técnica propuesta se basa en el algoritmo de persecución pura, de modo que tras determinar una configuración objetivo, mediante cualquier método, se genere una acción de control que permita al vehículo llegar a su objetivo. Además, la herramienta propuesta en este apartado permite la combinación con métodos estándar de planificación de caminos.

Por lo tanto, gracias al método desarrollado los robots móviles con ruedas mejoran la estimación de su posición reduciendo los errores de odometría al disminuir los deslizamientos de las ruedas. Esto se consigue mediante la inclusión de trayectorias de transición entre tramos rectos y tramos curvos, la cuales tienen una curvatura continua y un cambio de curvatura constante.

Lo único que no se ha definido en este apartado con cierto nivel de detalle es la obtención del punto o configuración objetivo. Se han comentado varias posibilidades: una opción sería a través de trayectorias pre-planificadas, de modo que el vehículo generase en tiempo real el camino necesario para seguirlas; otra opción sería partir de un planificador que calculase una secuencia de puntos libres de colisión que permitieran planificar a más bajo nivel el control cinemático para llegar a dichos puntos.

No obstante, estos métodos requieren de un conocimiento previo del entorno y del camino a recorrer, pudiéndose en muchos casos programar previamente una trayectoria que considerando las restricciones cinemáticas y dinámicas guiara al vehículo. Pero ese no es el objetivo final, puesto que si se conoce el recorrido a seguir el método propuesto no aporta nada de especial. Por lo tanto, lo mejor es aplicar el método propuesto para el control cinemático en el problema del seguimiento de líneas. Para dicha situación el camino no necesariamente debe conocerse a priori, pero sí que requiere identificarlo y determinar un punto o configuración objetivo que permita al robot seguir la línea. Entre los posibles métodos de detección de caminos presentados en el Anexo 3, el que mayor interés tiene, desde un punto de vista de investigación, es la utilización de sistemas y técnicas de visión artificial para determinar el punto objetivo a seguir.



# **CAPÍTULO 4**

## **Detección de Líneas mediante Sistemas de Visión Artificial**



## **4. Detección de Líneas mediante Sistemas de Visión Artificial**

### ***4.1. Introducción***

Como se ha comentado al final del Capítulo 3, el sistema de visión artificial se utilizará con el objetivo de identificar y modelar el camino a seguir, de modo que se pueda determinar un punto objetivo separado cierta distancia del vehículo. Por lo tanto, hay que destacar que la etapa de visión no es un fin en sí mismo sino un medio para determinar una configuración factible que sirva de referencia al sistema de generación de trayectorias y control cinemático del robot.

En el transcurso de este capítulo se detallarán las técnicas de procesamiento implementadas, explicando cómo y por qué se han utilizado. Se propondrá un algoritmo de visión de forma generalizada, de modo que queden definidas las distintas etapas de que consta, proponiendo algunos métodos para resolver cada una de las fases, pero sin dar una solución cerrada a las mismas, ya que no es objetivo de la presente Tesina el defender las técnicas de visión utilizadas para la detección de la línea, sino más bien explicar su utilización.

### ***4.2. Técnicas de Visión por Computador para la Detección de Líneas***

Antes de explicar el sistema de adquisición y procesamiento de imágenes desarrollado se introducirá el concepto de visión por computador o visión artificial, puesto que es la herramienta que se ha utilizado como elemento principal de sensorización en el desarrollo práctico de la Tesina.

Se entiende por visión artificial la disciplina científica que trata de construir sistemas que obtengan información de las imágenes y así poder interpretarlas debidamente. El objetivo principal es dotar a las máquinas del sentido de la vista.

En la actualidad no existe una forma estándar de resolver los problemas de CV (*Computer Vision*), por eso existe un amplio conjunto de métodos para problemas acotados y bien definidos, aunque en general suelen ser procedimientos específicos que no suelen poderse extrapolar.

Todos los sistemas de visión comparten un conjunto de procesos o etapas genéricas, tal y como se describe en la Figura 146. No obstante, no todos los sistemas de visión requerirán de todas las etapas ni del mismo nivel de detalle en cada una de ellas.

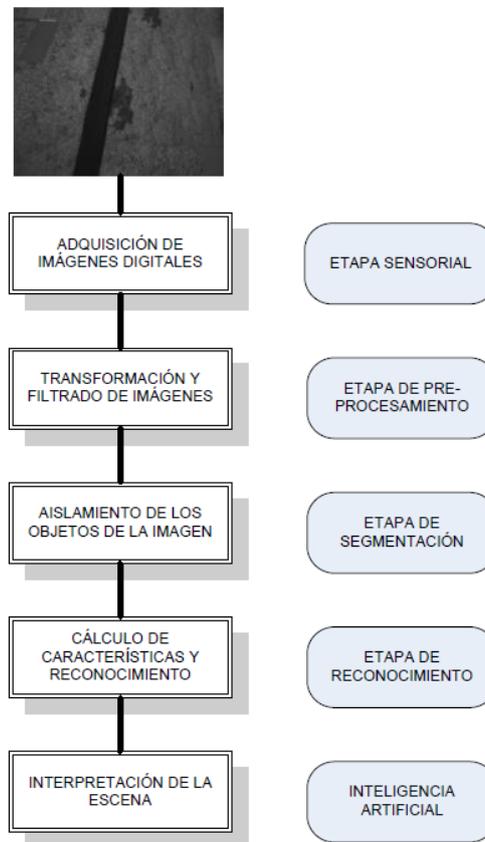


Figura 146. Etapas de procesamiento de un sistema de visión artificial.

- **Adquisición de imágenes.**

La adquisición de imágenes incluye todos los procesos llevados a cabo para que la información luminosa de un cuerpo sea adquirida por un elemento captador, transformada en una señal eléctrica y transmitida a una unidad de control, donde se almacene en formato digital. Esta fase, también llamada etapa sensorial, engloba las tareas de configuración de los parámetros relativos a la iluminación, la selección y configuración de los sensores ópticos, la elección del formato de la imagen, etc.

Finalmente también se lleva a cabo la digitalización de la señal que contiene la información luminosa de la escena, ya que, antes de poder enviar esta imagen electrónica al computador, se deben configurar en binario los niveles de intensidad de cada píxel.

- **Procesado de la información**

El procesamiento es donde se llevan a cabo las acciones para interpretar las imágenes captadas. Estas tareas pueden dividirse en tres grupos principales:

- **Pre-procesamiento:**

Con anterioridad al análisis de la imagen digital es conveniente realizar un pre-procesado de ésta, que consiste en el conjunto de tareas orientadas a mejorar su calidad o acentuar la información explotable para los algoritmos de reconocimiento automático. Este puede consistir en aplicar filtros a la imagen, para eliminar el posible ruido, realzarla, es decir, aumentar las diferencias entre dos zonas de la imagen con distintos valores de intensidad para compensar los efectos de sombras y reflejos, etc.

- **Segmentación:**

Es el proceso orientado a dividir la imagen en zonas disjuntas con significado propio, es decir, es donde se trata de aislar los objetos individuales presentes en la imagen. A partir de aquí el concepto de unidad pasa de ser el píxel a ser la agrupación de estos. Normalmente se trata de la etapa esencial y al tiempo la más compleja de la visión por computador.

Las operaciones de segmentación, así como el número de clases de regiones y objetos a distinguir, dependerán de la propia escena y de la información que se busque en la imagen.

Se pueden diferenciar dos grandes tipos, de regiones homogéneas, donde se agrupan los píxeles que tienen algún tipo de conectividad entre ellos; y de objetos, donde el objetivo es encontrar las fronteras de las entidades y se trabaja sobre los aspectos geométricos de éstas. A nivel conceptual, las técnicas aplicadas en ambos tipos son comunes y se pueden dividir en 3 grupos:

- **Aplicación de umbrales de color o de nivel de gris:**

Procedimientos basados en el uso del histograma. Consistentes en distinguir objetos individuales dentro de una escena a partir de los diferentes niveles de luminosidad.

- **Agrupación por características comunes:**

También conocido como *clustering*, se inspira en las técnicas de reconocimiento de formas y/o texturas.

- Extracción de bordes:

Es el método más empleado en la segmentación de imágenes, ya que dota al sistema de una alta efectividad. De hecho, la mayoría de los objetos se pueden reconocer a partir de su forma, que está representada por sus bordes. Dentro de esta familia de técnicas existen algoritmos muy diversos, como por ejemplo técnicas que se basan en el Gradiente o el Laplaciano de niveles de color, filtros como Sobel, algoritmos completos como Canny o Watershed, etc.

- **Reconocimiento:**

Trata de distinguir e identificar las regiones segmentadas. Consta de dos fases: extracción de características de los objetos segmentados e identificación de los mismos.

- Extracción de características:

A partir de los conjuntos segmentados se obtiene un conjunto de parámetros como perímetro, grado de circularidad, momento, etc. que se compararán en la siguiente fase con la base de conocimiento existente para determinar de qué elemento se trata.

- Clasificación de los objetos:

Con las características aportadas por la fase anterior, y partiendo de una catalogación previa se identificará la entidad en cuestión. En el sistema de visión debe existir un discernimiento almacenado, que debe ser elaborado por un equipo de especialistas. Este conocimiento previo es variado y puede ser muy sencillo, como en robótica, donde prácticamente solo hay que distinguir unas pocas figuras, o también puede ser muy complejo, como el necesario para la detección facial automática.

Antes de desarrollar el algoritmo de detección de caminos en su totalidad de etapas de procesamiento, en este apartado se va a realizar un estudio de los posibles algoritmos de detección de líneas rectas y ajuste de los parámetros de las mismas. De este modo se podrán entender cuáles son las ventajas y los inconvenientes de cada método para finalmente utilizar aquél que garantice mayor versatilidad y facilidad de programación.

### ***4.2.1. Regresión Lineal Simple***

La regresión lineal por el método de los mínimos cuadrados pretende encontrar la ecuación de la recta  $y(x) = a + bx$ , que represente de forma óptima la dependencia lineal de  $y$  sobre  $x$ , la llamada recta de regresión. Para la identificación de ésta se parte del diagrama de dispersión, un conjunto de  $N$  pares de valores  $(x_i, y_i)$ , que en la imagen representarán los píxeles de interés, con la estructura  $p$  (*fila, columna*).

Generalmente más de una curva del mismo tipo puede adaptarse correctamente al conjunto de medidas anteriores. En nuestro caso para encontrar aquella que minimice la diferencia de cada uno de los puntos al modelo final utilizamos el método de los mínimos cuadrados, LS (*Least Squares*).

Para la determinación de  $a$  y  $b$ , esta técnica trata de minimizar la suma de los cuadrados de las desviaciones de las medidas a la recta calculada. La desviación de cada observación  $(x_i, y_i)$  a la recta calculada se representa por  $d_i$  y lógicamente, como se ve en la Figura 147, puede ser positiva, negativa o cero.

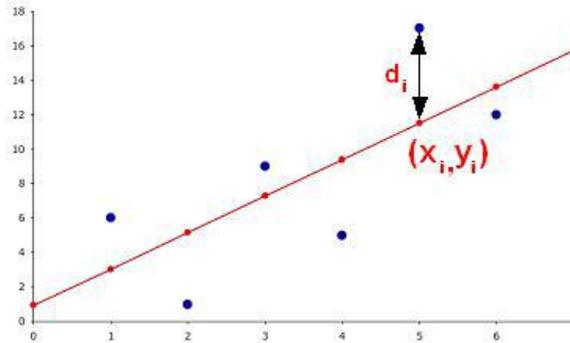


Figura 147. Error de estimación método de mínimos cuadrados.

De esta manera el índice de optimización será:

$$d = \sum_{i=1}^N [y(x_i) - y_i]^2 \tag{159}$$

Es decir:

$$f(a, b) = \sum_{i=1}^N [(a + b \cdot x_i) - y_i]^2 \tag{160}$$

Para que  $f(a, b)$  sea mínima, se deberán de anular las derivadas de la función respecto de  $a$  y  $b$ . Al derivar e igualar a 0, las expresiones resultantes nos indican los valores de los parámetros de la recta que minimizan el sumatorio de los cuadrados de las distancias de todos los puntos a ésta.

$$a = \frac{\sum_{i=1}^N y_i \sum_{i=1}^N x_i^2 - \sum_{i=1}^N x_i \sum_{i=1}^N [x_i \cdot y_i]}{N \cdot \sum_{i=1}^N x_i^2 - (\sum_{i=1}^N x_i)^2} \tag{161}$$

$$b = \frac{N \cdot \sum_{i=1}^N [x_i \cdot y_i] - \sum_{i=1}^N y_i \cdot \sum_{i=1}^N x_i}{N \cdot \sum_{i=1}^N x_i^2 - (\sum_{i=1}^N x_i)^2} \tag{162}$$

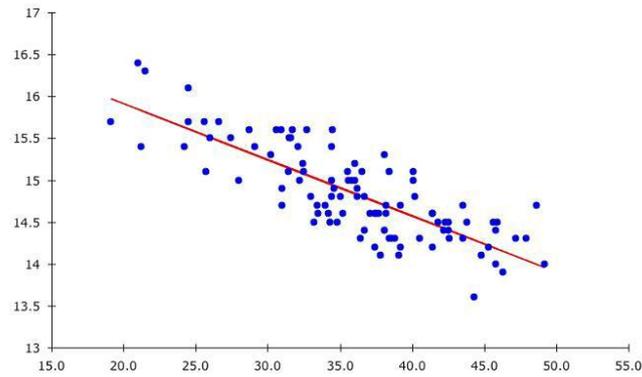


Figura 148. Recta de regresión. Método mínimos cuadrados.

### 4.2.2. Regresión Lineal Ortogonal

En este caso el objetivo es minimizar la distancia de los puntos a la recta, por lo que el resultado del algoritmo serán los parámetros del modelo que minimice la suma de las distancias ortogonales de cada punto a éste. Antes de explicar brevemente el algoritmo, se muestra la representación de la recta de la que hacemos uso. Esta notación llamada Distancia-Ángulo define la recta como se ve en la Figura 149, por medio de las variables  $\rho$  y  $\phi$ .

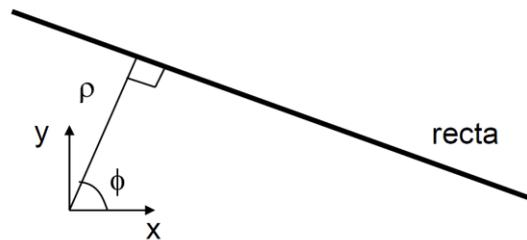


Figura 149. Representación de la recta por el método Distancia-Ángulo.

Siendo la ecuación que caracteriza la recta la siguiente:

$$\rho = x \cos \phi + y \sin \phi \tag{163}$$

Una vez definido el modelo deseado pasaremos a explicar la forma de trabajar del LS Ortogonal. Básicamente lo hace de la misma manera que el LS simple, a diferencia de que el objetivo es minimizar las distancias ortogonales, tal y como se muestra en la Figura 150.

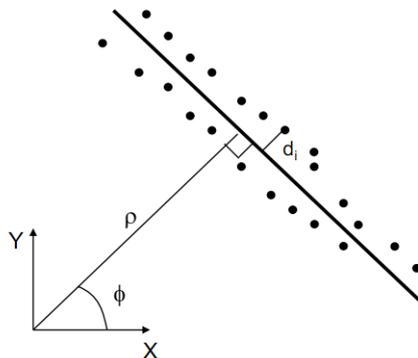


Figura 150. Regresión ortogonal. Variables consideradas.

Así, el índice de optimización a minimizar será el siguiente:

$$J = \frac{1}{2} \sum_{i=0}^{N-1} (x_i \cdot \cos(\phi) + y_i \cdot \sin(\phi) - \rho)^2 \quad (164)$$

Siendo  $N$  el número total de puntos.

Si derivamos la expresión anterior respecto a las variables que se desea estimar e igualamos a cero se llega a la siguiente solución:

$$\frac{\partial J}{\partial \rho} = 0 \rightarrow \hat{\rho} = \bar{x} \cdot \cos\phi + \bar{y} \cdot \sin\phi \quad (165)$$

$$\frac{\partial J}{\partial \phi} = 0 \rightarrow \hat{\phi} = \frac{1}{2} \cdot \arctan\left(\frac{-2 \cdot \sigma_{xy}^2}{\sigma_y^2 - \sigma_x^2}\right) \quad (166)$$

### 4.2.3. Regresión Lineal Ortogonal Robusta. Método RANSAC

Esta técnica se diferencia de las anteriores en que es capaz de realizar una estimación robusta para un modelo paramétrico. Es decir, puede estimar las características del modelo con un alto grado de precisión, aun cuando la muestra de entrada esté formada por un alto porcentaje de datos espurios.

Al igual que los anteriores procedimientos, RANSAC, abreviación de *Random Sample Consensus* [105], [106], [107], trata de estimar los parámetros a partir de un conjunto de observaciones. La diferencia es que este método no es determinista, es decir, produce un resultado razonable dentro de una cierta probabilidad, la cual está directamente ligada al número de iteraciones efectuadas.

RANSAC trabaja sobre un conjunto de datos, que son catalogados como una mezcla de *inliers* y *outliers*, según puedan ser descritas por el modelo a determinar o no (ver Figura 151). Dentro de un conjunto de datos observados los *outliers* pueden estar provocados entre otras razones por valores extremos de ruido o por medidas incorrectas.

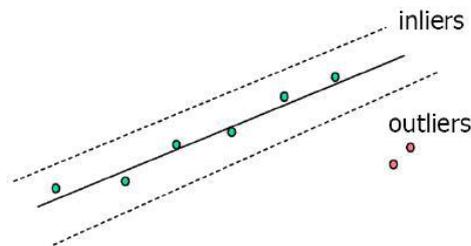


Figura 151. Caracterización de puntos con RANSAC.

Una vez catalogados los datos, RANSAC calculará los parámetros del modelo siguiendo el procedimiento de la regresión lineal ortonormal, pero a diferencia del anterior método solamente empleará los valores adecuados, los que mejor representan al modelo a identificar.

El método RANSAC parte de la hipótesis de que dado un conjunto de *inliers* existe un procedimiento con el que se pueden estimar los parámetros del modelo que los representa de forma óptima. Básicamente su forma de trabajar es la siguiente:

1. Escoge aleatoriamente un subconjunto de puntos, dentro del conjunto global de observaciones  $S$ .
2. Estima los parámetros del modelo con el subconjunto  $s$ .
3. A partir del modelo, determina cuántos puntos de  $S$  cumplen con esa estimación, es decir, cuántos de estos se encuentran dentro de un cierto umbral, clasificando como *inliers* el conjunto  $s_i$ .
4. El proceso se repite un número determinado de iteraciones o hasta que la probabilidad de encontrar una mejor solución sea pequeña. Este último caso se da cuando  $s_i$  es mayor que un umbral. Llegado a este punto se calculan los parámetros del modelo a partir de  $s_i$  y se acaba el algoritmo.
5. Si no se da el caso anterior, en cada iteración nos quedamos con la mejor solución encontrada hasta el momento, es decir, la que mayor número de *inliers* ha generado.
6. Una vez hemos clasificado los puntos como *inliers* y *outliers*, realizamos la estimación del modelo únicamente utilizando los puntos *inliers*.

Las figuras siguientes ilustran lo explicado anteriormente mediante un ejemplo donde se ha aplicado el algoritmo para identificar una recta a partir de una nube de puntos. Así en la Figura 152 se observa que, aunque los datos tienen una tendencia clara, el número de medidas que no la siguen es bastante numeroso.

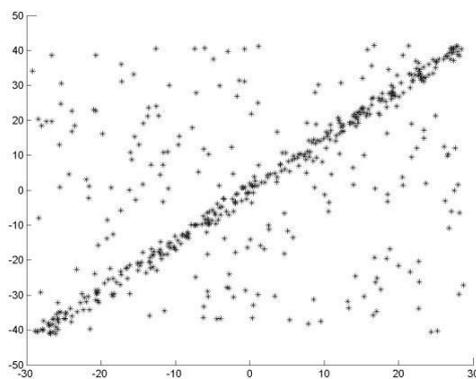


Figura 152. Conjunto de observaciones de partida, algoritmo RANSAC

RANSAC irá tomando subconjuntos de valores y calculando los respectivos modelos, como vemos en la Figura 153, donde se presenta en rojo los valores que forman el subconjunto  $s_i$  y en verde el modelo estimado. Esta ilustración muestra una iteración, el resultado de la cual, se ha descartado ya que el número de puntos pertenecientes a  $s_i$  no superaba el umbral exigido.

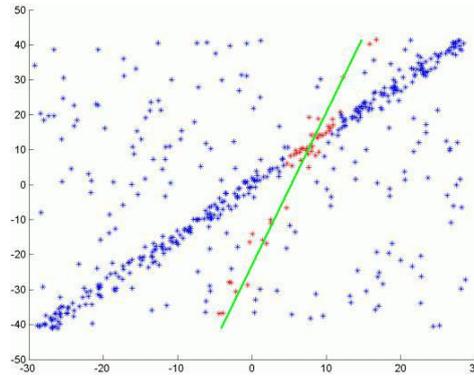


Figura 153. Resultado iteración intermedia, algoritmo RANSAC

Finalmente, en la Figura 154 podemos comprobar el resultado final del algoritmo. Aquí podemos verificar que, para el cálculo de los parámetros del modelo, únicamente se han utilizado *inliers*, es decir, los valores fuera del umbral no han influido en el resultado.

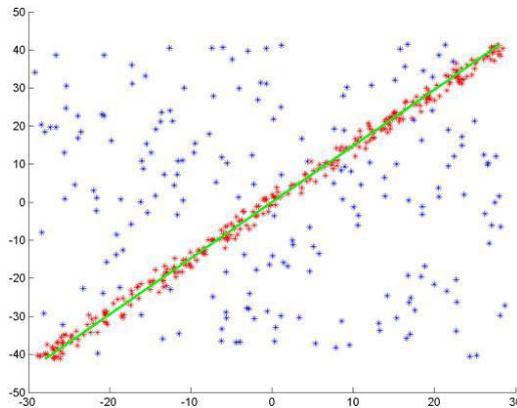


Figura 154. Resultado final. Estimación modelo con algoritmo RANSAC

El método presenta la desventaja de tener un periodo de ejecución variable. Además, utilizar el máximo número de iteraciones para hacer el cálculo no implica que la solución obtenida sea la óptima, puede no ser la mejor, es decir, puede darse el caso de que después de ejecutarse todas las iteraciones el modelo estimado no sea correcto al haber utilizado para el cálculo algunos puntos *outliers*.

#### 4.2.4. Transformada de Hough

La Transformada de Hough es un algoritmo empleado en reconocimiento de patrones en imágenes que permite encontrar ciertas formas dentro de una imagen, tales como: líneas, círculos, elipses, etc. Se trata de una técnica clásica utilizada en visión artificial que permite transformar un elemento en un espacio dado (cartesiano, polar, etc.) en el espacio de Hough.

La versión más simple consiste en encontrar líneas rectas. Su modo de operación es principalmente estadístico y consiste en que para cada punto que se desea averiguar si es parte de una línea se aplica una operación dentro de cierto rango, con lo que se averiguan las posibles líneas de las que puede ser parte el punto. Esto se continúa para todas los puntos en la imagen, al final se determina qué líneas fueron las que más puntos posibles tuvieron y esas son las líneas en la imagen.

La Transformada de Hough para rectas se puede resumir de la siguiente forma:

- Definimos el espacio de parámetros  $\rho$  y  $\phi$ .
- Aplicamos a cada punto de la imagen la transformada de Hough para rectas, es decir, calculamos todas las posibles rectas que pasan por el punto.
- Acumulamos aquellas casillas del espacio de parámetros por las que pasen las rectas.
- Calculamos el máximo del espacio de parámetros, que debe coincidir con los parámetros de la recta detectada.

La Transformada de Hough emplea una representación paramétrica de formas geométricas. Una recta, por ejemplo se representa por un módulo  $\rho$  (perpendicular a la recta y que pasa por el origen (0,0) y un ángulo  $\phi$  (formado por el módulo y el eje positivo de las  $x$ 's). Se representa igual que en la ecuación (163) de la regresión ortonormal, es decir:

$$\rho = x \cdot \cos(\phi) + y \cdot \sin(\phi) \quad (167)$$

El método es robusto frente a ruidos y oclusiones parciales de objetos. Además, otra ventaja de este método es que evita singularidades, como por ejemplo rectas de pendiente infinita. Si se representa  $\phi$  y  $\rho$  en un plano cartesiano, una recta queda determinada mediante un punto con coordenadas  $(\phi(\text{ángulo}), \rho(\text{recta}))$ , mientras que un punto, se representa como una función senoidal. Si por ejemplo tenemos dos puntos, tendremos dos senoides desfasadas  $\alpha$  grados dependiendo de las coordenadas de los puntos. Dichas senoides se irán cruzando cada  $180^\circ$ . La interpretación geométrica de este hecho, es que la función seno de cada punto representa las infinitas rectas que pasan por ellos, cuando dos puntos comparten la misma recta sus representaciones senoidales se cruzan y se obtiene un punto. Cada vez que se da media vuelta ( $\rho = 180^\circ$ ) se vuelve a repetir la misma recta, por lo que volvemos a obtener otro punto, que de hecho es la misma recta.

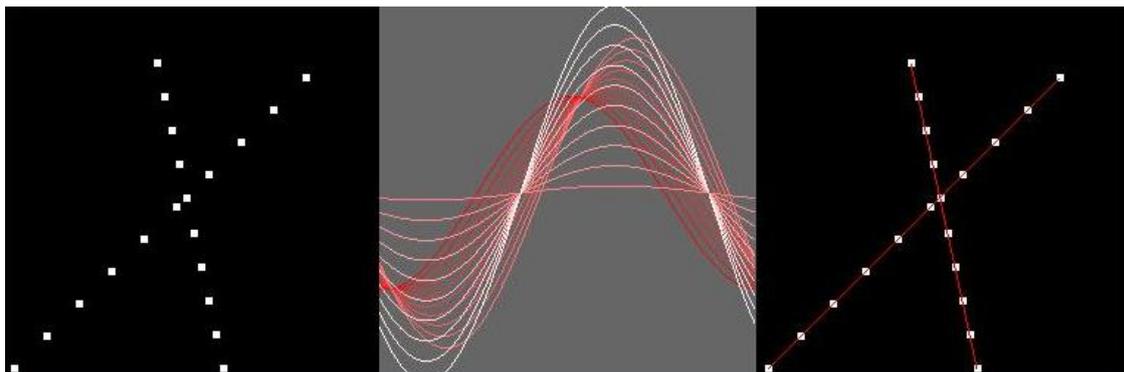


Figura 155. Ejemplo de transformada de Hough para reconocimiento de rectas.

### 4.3. Algoritmo de Detección de Líneas para Sistemas de Visión (con Recursos Ilimitados)

En este apartado se desarrollará un algoritmo genérico de detección y modelado de líneas o caminos, con tal de ofrecer al robot móvil la sensorización y la capacidad de procesamiento necesaria para determinar un punto objetivo al que se quiere llegar cuando éste está siguiendo un camino cualquiera.

Antes de detallar cada una de las etapas se va a presentar un diagrama de flujo que permita ver la relación existente entre las distintas fases del procesado de la imagen. Dicho flujograma aparece en la Figura 156.

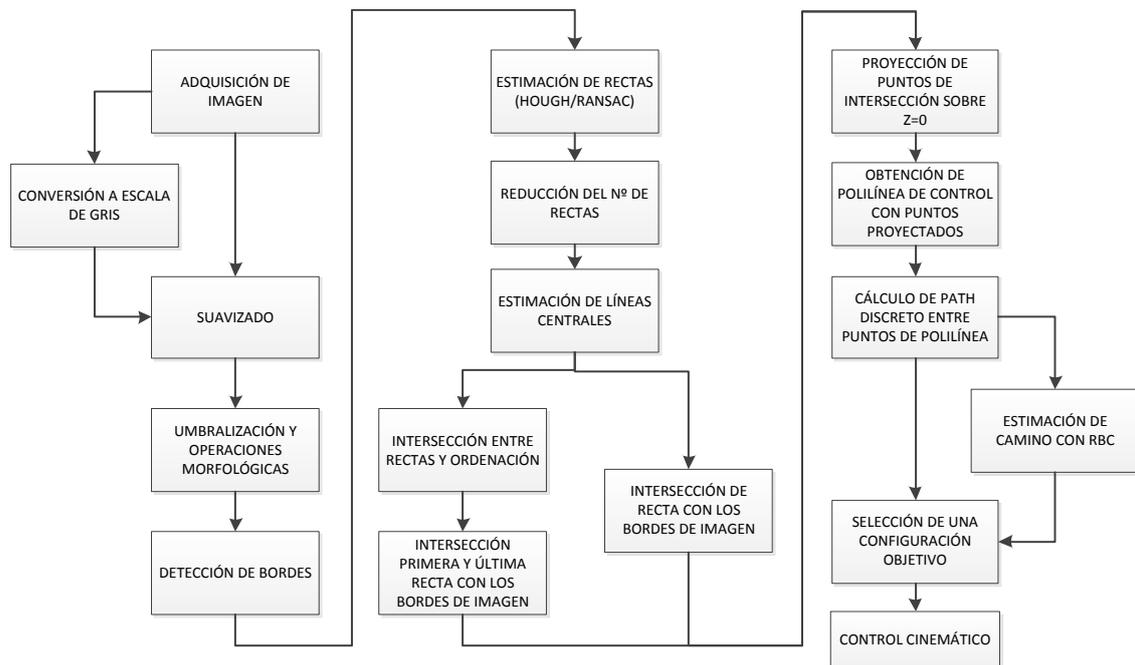


Figura 156. Diagrama de flujo del procesamiento.

En primer lugar se explicará la etapa en la que se pre-procesa la imagen con el objetivo de segmentarla, detectando la región de píxeles que corresponde a la línea buscada. Antes de dicha etapa es necesario adquirir la imagen y en este caso realizar una transformación de color para convertirla a niveles de gris. En la Figura 157 (a) y (b) se observan dos ejemplos de imágenes originales en color, mientras que en la Figura 157 (c) y (d) se observan sus correspondientes imágenes transformadas al espacio de niveles de gris.

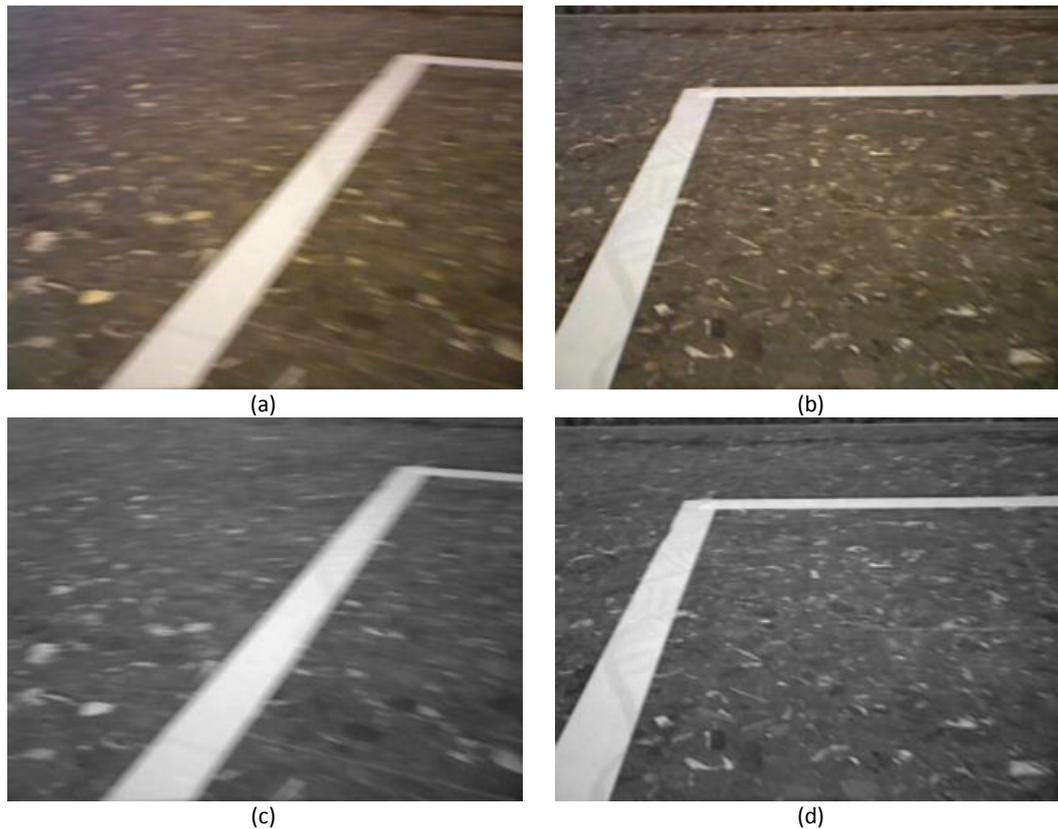


Figura 157. Ejemplos de imágenes de líneas: (a) y (b) originales en color, (c) y (d) transformadas a escala de gris.

La siguiente etapa que se requiere es un suavizado de la imagen, o en algunos casos un realce de la misma. Para ello se utilizan elementos estándar como por ejemplo: filtros de mediana o filtros gaussianos, operaciones de emborronamiento o *blurring*, operaciones morfológicas para imágenes en color, etc. El objetivo es eliminar posibles ruidos, sombras y objetos indeseados que se puedan confundir con la línea buscada, por ejemplo por tener el mismo color. En la Figura 158 se muestran las imágenes anteriores suavizadas mediante un filtro de mediana. Como se observa en la Figura 158 (b) se han eliminado la mayoría de las manchas blancas que aparecían como objetos indeseados de la imagen, en cambio, en la Figura 158 (a) las manchas no se han eliminado totalmente.

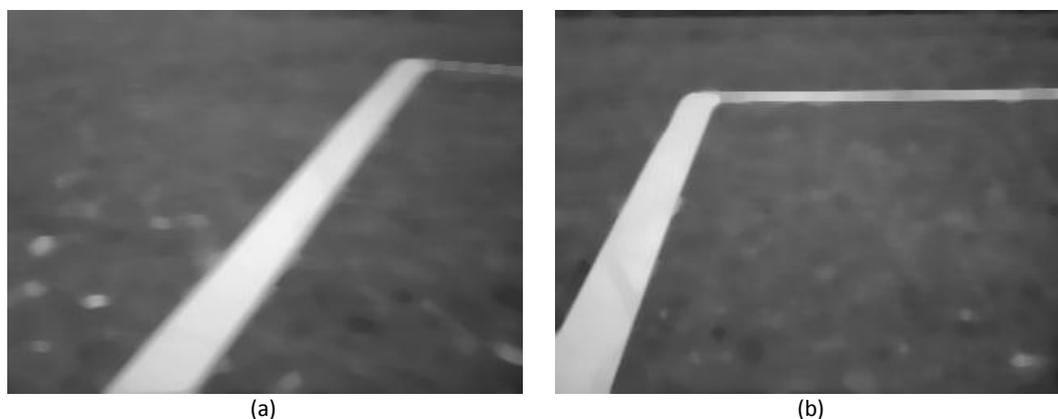


Figura 158. Ejemplos de líneas tras aplicar un suavizado.

Tras la operación anterior y dadas las características de las imágenes con las que se está trabajando, el siguiente paso sería el de umbralizado. En este caso, al tener una línea blanca bien contrastada con respecto al fondo se buscarían los píxeles de mayor nivel de gris descartando los píxeles más oscuros. El procedimiento genérico consiste en fijar uno o varios niveles límite o *threshold* a partir de los cuales se descartan unos píxeles o se consideran como pertenecientes al objeto buscado.

Hay que destacar que no siempre el umbralizado se realizará de esta forma, sino que habrá que adaptarlo al caso concreto en el que se está trabajando. Por ejemplo se podría realizar un umbralizado adaptativo mediante la observación del histograma de la imagen, viendo dónde es más probable que esté la línea buscada por la acumulación de píxeles en dicha zona. Además, existen varios métodos de umbralizado, similares a los filtros electrónicos típicos: paso-bajo, paso-alto, pasa-banda, rechaza-banda, etc. En la Figura 159 aparecen umbralizadas las imágenes de los ejemplos anteriores.



Figura 159. Ejemplos de líneas tras aplicar un umbralizado.

Puede que en algunos casos la umbralización por color no sea suficiente para eliminar los elementos indeseados, por ello, es posible complementarla con técnicas de detección de bordes como por ejemplo el filtro de Sobel o el del gradiente, que convierten la imagen de color a una imagen en niveles de gris en las que se resaltan los bordes de los objetos.

También se puede incluir una etapa con operaciones morfológicas sobre imágenes de blanco y negro, con el objetivo de dejar únicamente los objetos buscados. En el ejemplo que se está mostrando las mejoras no son significativas, pero puede haber casos en los que sí se mejore notablemente.

Las Figuras 160 (a) y (b) muestran el resultado tras una operación de erosión. El objetivo es reducir el tamaño de los objetos eliminando parte de sus bordes, por lo que en caso de haber objetos pequeños indeseados estos pueden desaparecer, quedando aquellos de mayor grosor. Como se observa, en el ejemplo de la Figura 160 (a) sí se eliminan algunos elementos pequeños que aparecían en la imagen. En cambio, en el ejemplo de la Figura 160 (b), al no tener objetos de tamaño reducido no se elimina nada.

La operación complementaria a la erosión es la dilatación, cuyo objetivo es agrandar el tamaño de los objetos añadiendo píxeles en los bordes. Con dicha operación se consigue retornar al tamaño original de la línea para que no haya pérdida de información de su forma original. Las Figuras 160 (c) y (d) muestran los resultados tras la operación de dilatación.

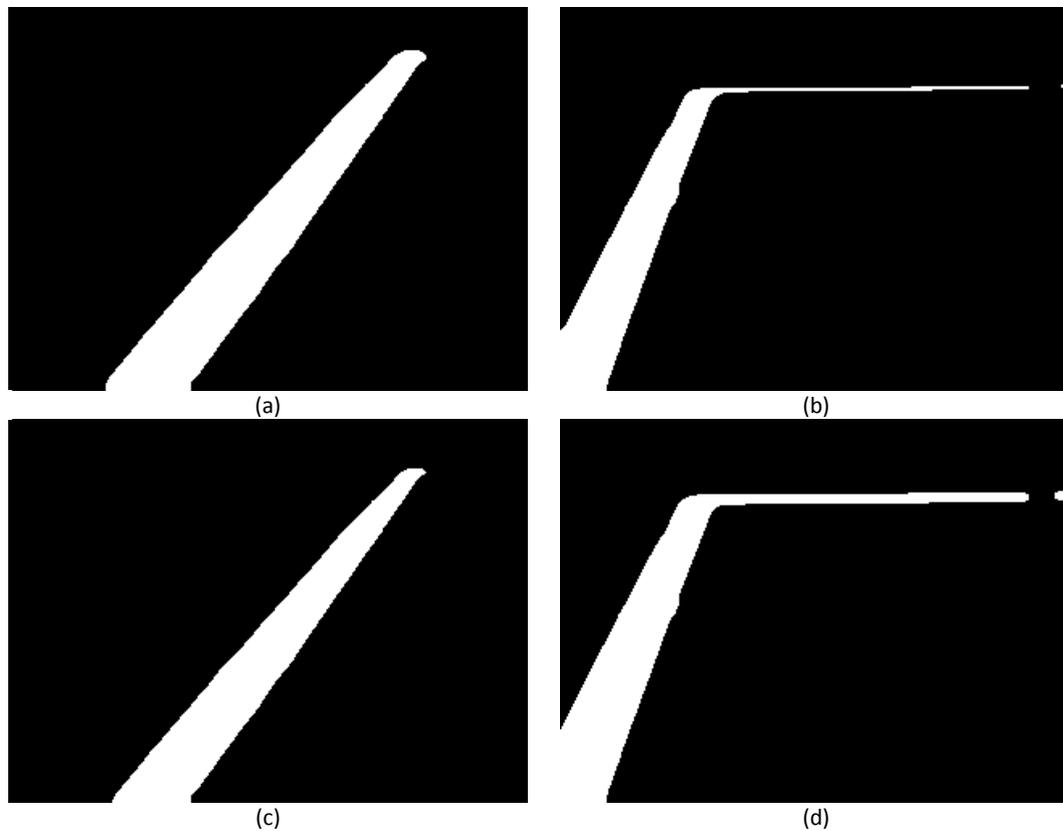


Figura 160. Ejemplos de líneas (a) y (b) tras realizar una erosión, (c) y (d) tras una dilatación.

Una etapa opcional pero interesante es la detección del mínimo rectángulo que inscribe el objeto detectado, con tal de fijar una región de interés o ROI (*Region Of Interest*). Con esta etapa se puede fijar una zona de menor tamaño que la propia imagen y como consecuencia mejorar el procesamiento mediante la reducción del tiempo de cómputo. Las Figuras 161 (a) y (b) muestran el ROI generado para cada ejemplo.

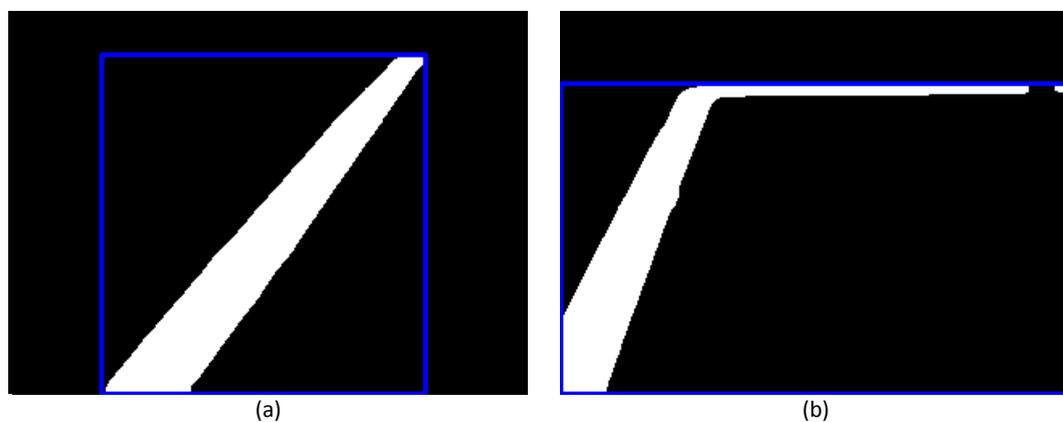


Figura 161. Ejemplos de líneas en las que se ha detectado un ROI (rectángulo azul).

Aunque no se realice la etapa anterior de búsqueda de la región de interés y directamente se utilice la imagen completa, el siguiente proceso a realizar es la detección de bordes. Existen muchas técnicas pero entre todas ellas destaca el algoritmo Canny. En la Figura 162 aparecen las imágenes de blanco y negro con los bordes de la línea detectada.

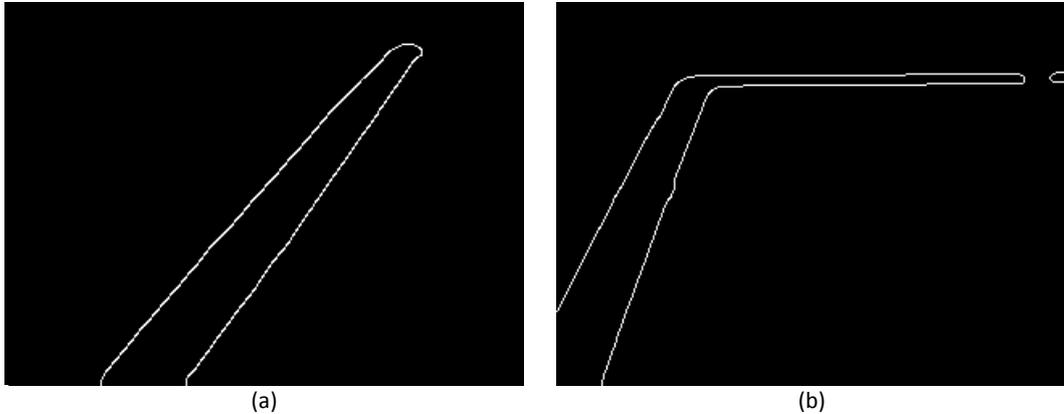


Figura 162. Ejemplos de líneas en tras realizar una detección de bordes.

Tras la detección de los bordes el siguiente paso es determinar la línea central de puntos que sirva para modelar e identificar paramétricamente el camino que se desea seguir. Para ello se propone el uso de la técnica de detección de rectas basada en la Transformada de Hough. Es muy sencillo implementar dicha técnica, pero en la mayor parte de librerías de visión está implementada, lo cual es una ventaja.

El objetivo es por tanto detectar las posibles rectas que superen una serie de requisitos, como por ejemplo que su longitud sea mayor que cierto valor, que tenga un número de píxeles determinado, o que la separación horizontal de sus puntos sea menor que cierto valor. Se trata de un procedimiento empírico que variará según el tamaño de imagen y la forma de la línea buscada. Es evidente que no es lo mismo buscar en un entorno estructurado y conocido a priori que en uno desestructurado y difícil de procesar, como por ejemplo un camino o carretera curvos.

A continuación se va a describir una forma genérica de detección de rectas en la imagen y cómo obtener la secuencia de puntos centrales de la línea. Se detallará el procedimiento seguido para esta aplicación concreta, en la que se tiene una secuencia de líneas rectas pintadas en el suelo, pero podría extrapolarse para casos más genéricos con líneas curvas, como posteriormente se explicará.

Los pasos a seguir en esta fase de procesado son los siguientes:

- Realizar una primera búsqueda de posibles líneas rectas en la imagen de bordes mediante cualquier método de los explicados en el apartado anterior (por su robustez se recomienda el método de la transformada de Hough o el algoritmo de RANSAC).
- Tras obtener una secuencia de rectas detectadas en la imagen se realiza una primera agrupación por similitud de los parámetros de ángulo y distancia ( $\phi$  y  $\rho$ ). Dicha aproximación se hace de modo que se limiten a valores muy pequeños las diferencias entre parámetros. Se realiza de este modo para agrupar las rectas que son prácticamente iguales, por ejemplo las de un mismo borde de la línea. Se calcula la media de sus

parámetros, consiguiendo que no se acumulen más rectas en un lado que en otro y reduciendo el número total de líneas detectas.

- A continuación se realiza otra agrupación menos fina, es decir, aumentando el rango de aceptación de los parámetros de las rectas que se consideran similares. Con esto lo que se consigue es agrupar las rectas de parámetros similares dentro de cierto margen, por ejemplo se agrupan las líneas de cada lado en una única recta, situada justo en el centro de ambas. Esto será así siempre y cuando se haya detectado más de una recta, puesto que si sólo se detecta uno de los bordes no habrá problemas ya que esa será directamente la línea buscada.
- Tras el proceso anterior se obtiene una secuencia de rectas con distintas direcciones y tamaños que modelan de forma muy burda la línea buscada. El problema es no se tiene conocimiento del orden de la secuencia de rectas o de dónde empiezan o acaban, puesto que son un conjunto de parámetros sin significado propio si no se analiza más la situación. Por lo tanto, lo siguiente que se realiza es una ordenación de las rectas detectadas con tal de obtener una secuencia lógica según su posición respecto a la imagen y en consecuencia respecto al robot.
- En caso de detectar sólo una recta no hay problema, por lo que el siguiente paso es detectar su intersección con los bordes de la imagen y ordenar los puntos para que el primero sea el de mayor  $Y$ , ya que es el extremo que más cerca quedará de la posición real del vehículo al proyectarse.
- En caso de detectar dos o más rectas, la ordenación de las mismas se realiza según el siguiente procedimiento:
  - La primera recta será la que tenga uno de sus dos extremos con mayor valor de la coordenada  $Y$ , es decir, aquella recta que empieza en la parte más baja de la imagen y por lo tanto más cerca está del robot.
  - En caso de haber dos líneas con el mismo valor máximo de la coordenada  $Y$ , la primera será la que tenga más centrada la coordenada  $X$  del punto extremo cuyo valor  $Y$  es el máximo.
  - A continuación se realiza una ordenación del resto de rectas. El criterio es agregar de forma correlativa al *path* aquellos puntos que pertenecen al resto de rectas y que están más cerca del extremo opuesto al que sirvió como unión para considerar la última recta como perteneciente al *path*.
  - Después se calculan las intersecciones de forma ordenada entre todas las rectas detectadas. Se obtiene una secuencia de puntos correlativos desde el punto inicial (intersección de la primera recta con la imagen), pasando por los intermedios (intersección entre una recta y la inmediata posterior), hasta el punto final (intersección de la última recta detectada con la imagen).

En la Figura 163 aparecen dos imágenes con los resultados obtenidos tras la identificación de las rectas que modelan el camino, quedando patente que el método propuesto funciona correctamente.

Se observa que en la Figura 163 (a) inicialmente se detectan los dos bordes (líneas verdes), pero que posteriormente se reduce a una única recta central (línea roja). Por lo tanto sus extremos son directamente las intersecciones de la línea con los bordes de la imagen (puntos azules). En cambio, en la Figura 163 (b) se detectan inicialmente tres rectas (líneas verdes), dos bordes de la línea que está más vertical y únicamente uno de la línea horizontal. Posteriormente se estiman las líneas centrales (líneas rojas) y se realiza la ordenación, de modo que la primera es la de mayor  $Y$  y la segunda es la que tiene su punto más cercano al extremo opuesto al inicial de la recta (se observa un ligero gradiente de color de más rojo a menos rojo según el orden de las líneas). Finalmente se calculan las distintas intersecciones entre rectas y entre la primera y última recta con los bordes de la imagen (puntos azules).

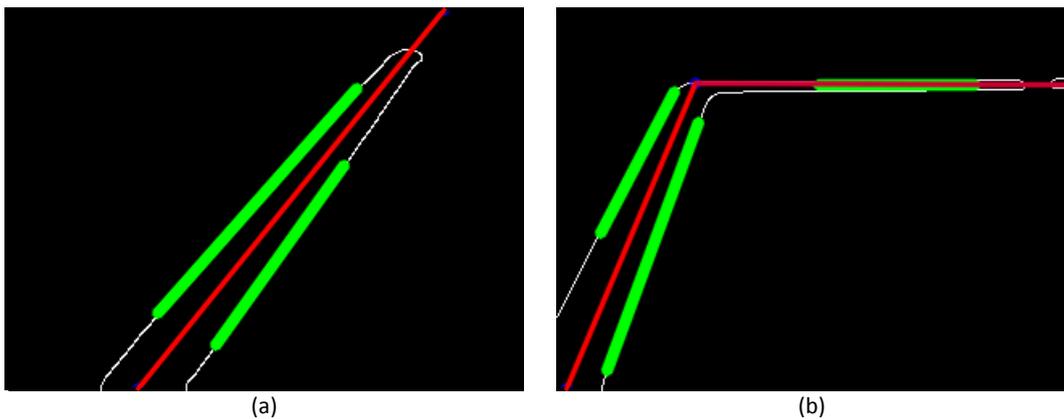


Figura 163. Ejemplos de líneas en las que mediante Hough se han detectado los bordes (líneas verdes), la línea central (rojo) y las intersecciones entre rectas (puntos azules).

Una vez se dispone de un conjunto de puntos ordenado, la siguiente etapa consiste en obtener una primera aproximación del camino a partir de la secuencia de puntos que se tiene. Por lo tanto, el objetivo es, empezando por el primero y acabando por el último, unir los puntos consecutivos mediante una secuencia de puntos intermedios separados una distancia determinada. Esta operación se puede realizar tanto en el plano de la imagen como en el plano de tierra, siendo más lógico hacer lo segundo. Para ello es necesario proyectar los puntos sobre dicho plano  $z = 0$ , pero suponiendo que se tenga el sistema calibrado o se haya obtenido una homografía (ver Anexo 4) este procedimiento es muy sencillo, ya que basta con pasar los puntos de la imagen por la matriz homogénea de transformación. En las gráficas de la Figura 164 aparecen proyectadas sobre el suelo las líneas que se han detectado en las fases previas de procesamiento de la imagen.

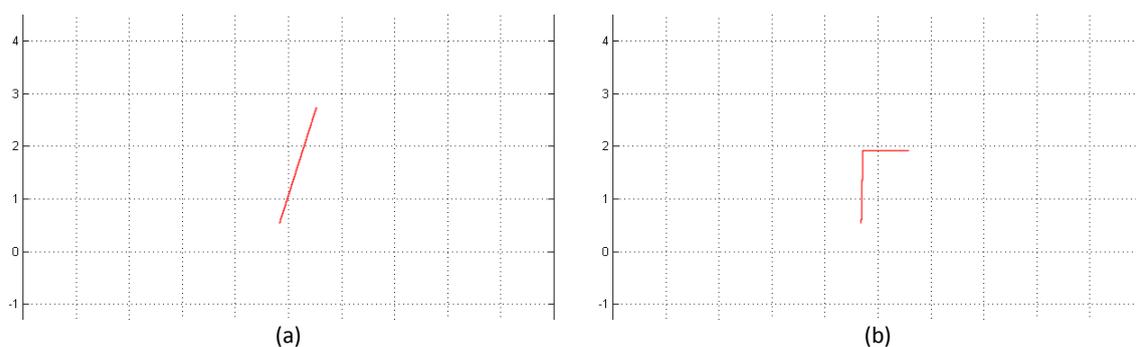


Figura 164. Ejemplos de líneas proyectadas sobre el plano de tierra  $z = 0$ .

Una vez se han proyectado los puntos o vértices de las rectas, como se ha dicho, lo siguiente es estimar la secuencia de puntos que los une para formar un camino completo. El método que se propone consiste en tomar la primera recta y alargarla hasta la altura del robot, de modo que su primer extremo quede justo en la intersección entre la recta y la proyección del robot sobre la misma. Lo siguiente que se debe hacer es incluir la posición del robot en el *path*. Por simplificación se considera  $q_R = (0,0)$ , puesto que tanto la imagen como el control que se realizará posteriormente están referidos al sistema de coordenadas del vehículo. Por lo tanto se dispone de un camino completo que une el robot con el punto de la primera recta que más cercano está de éste y así consecutivamente hasta la última recta.

Para la última fase existen varias posibilidades, cada una con sus ventajas y sus inconvenientes. A continuación se enumerarán las distintas posibilidades, pudiéndose combinar de la forma que se desee.

- Con la secuencia de puntos ya se dispone de un camino, modelado mediante rectas, que puede servir para obtener el punto objetivo que hay que considerar para realizar la etapa de control cinemático de seguimiento de líneas. La selección del punto objetivo se puede hacer básicamente de dos formas:
  - Tomar aquel punto que esté a una distancia  $LA$  determinada respecto al vehículo, es decir, tomar el punto que interseccione con un arco de radio  $LA$  con centro en el origen de coordenadas del robot.
    - Ventaja: El punto objetivo siempre está a una distancia pre-establecida y aceptable para que el robot sea capaz de llegar a su objetivo. Además dicha distancia puede variar en función de la velocidad.
    - Inconvenientes: Puede ocurrir que el arco interseccione con más de una recta según la configuración del camino, o bien que no interseccione con ninguna, en cuyo caso la solución sería el punto más cercano del camino detectado. Además, según qué casos puede ser difícil encontrar el punto de intersección del arco, puesto que como ahora se verá no siempre se modela la línea mediante rectas, sino también mediante curvas.
  - Tomar aquel punto que esté a una distancia  $LA$  determinada pero no respecto al vehículo como una distancia euclídea, sino que dicha distancia se mide de forma lineal con respecto a la longitud del camino, es decir, a lo largo de éste.
    - Ventajas: No hay ningún arco por lo tanto no puede interseccionar con más de una recta ni tampoco no interseccionar con ninguna, ya que la ruta empieza desde la misma posición actual del robot. Además, al no tener que realizar intersecciones siempre es sencillo determinar un punto objetivo, puesto que basta con calcular distancias.
    - Inconveniente: El punto objetivo puede estar a una distancia inaceptable desde el punto de vista de que el robot sea capaz de llegar a su objetivo, ya que en caso de detectarse zonas con curvas muy cercanas y cerradas, el punto objetivo puede caer muy cerca del robot complicando el proceso de control.

- Con la secuencia de puntos se puede realizar una aproximación del modelo del camino mediante curvas de Bezier racionales. Gracias a eso se obtienen trayectorias más suaves, aun cuando las rectas detectadas tengan cambios bruscos de dirección. Además, otra ventaja considerable es que la configuración objetivo ya no es una configuración de tipo pose  $q(x, y, \theta) \equiv q(x, y, \theta, 0)$ , sino que se trata de una postura completa  $q(x, y, \theta, \kappa)$ . Esto se debe a que antes se tenía rectas en las que la curvatura siempre era nula, lo que equivale a no considerarla. Pero con el uso de curvas de Bezier se puede estimar la curvatura del punto objetivo a partir del cálculo de la derivada de la RBC en dicho punto, lo cual aporta grandes ventajas desde el punto de vista de la generación de trayectorias y sobre todo desde el punto de vista del control.

En la Figura 165 se muestran los resultados de la estimación proyectados sobre el plano de la imagen. Como se puede ver aparecen en rojo los puntos del *path* detectado originalmente con las intersecciones de las rectas, lo cual podría ser la solución final. No obstante, también se muestran la aproximación mediante RBC (línea azul) y los puntos de control de la curva (círculos verdes). Como se observa se ha utilizado una RBC de orden cúbico, es decir  $N=3$ , y por eso aparecen cuatro puntos de control. También se ve la ventaja de utilizar RBC puesto que el método es tan versátil que con un mismo elemento se pueden aproximar tanto rectas únicas (ver Figura 165 (a)) como curvas o varias rectas (ver Figura 165 (b)). El hecho de utilizar cúbicas es porque se trata de las curvas de menor orden que son capaces de realizar trayectorias de tipo *S*, es decir, capaces de realizar un cambio de signo de la curvatura.

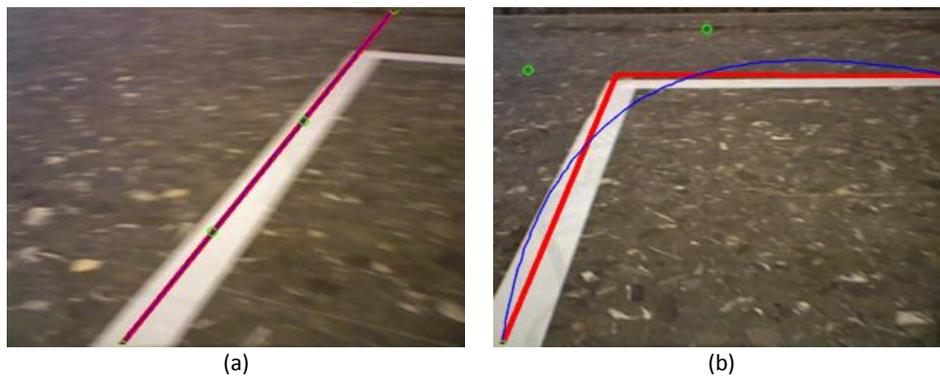


Figura 165. Ejemplos de líneas proyectadas sobre la imagen (rojo) y estimación del camino mediante RBC (azul).

Para el caso de entornos menos estructurados como por ejemplo una carretera, se demuestra mediante una secuencia de imágenes que el método propuesto es igualmente válido. En primer lugar se realiza la captura (ver Figura 166 (a)) y la conversión a niveles de gris, con la posterior fase de suavizado o emborronamiento (ver Figura 166 (b)).



Figura 166. Ejemplo de carril de carretera: (a) imagen original y (b) imagen emborronada en escala de gris.

Posteriormente se realiza un umbralizado y diversas operaciones morfológicas de dilatación y erosionado, dando como resultado la imagen de la Figura 167 (a). A continuación se realiza una detección de bordes mediante el método de Canny (ver Figura 167 (b)).

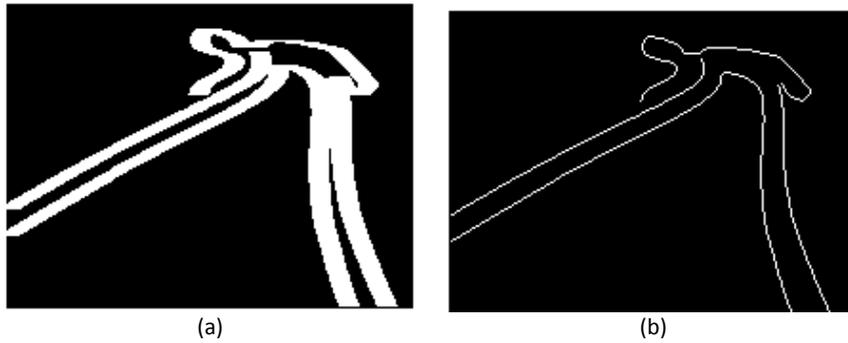


Figura 167. Ejemplo de carril de carretera: (a) imagen tras varias operaciones morfológicas y (b) imagen después del proceso de detección de bordes.

En la Figura 168 (a) se muestra superpuesta con la imagen original la detección del ROI (cuadro azul), junto con el área detectada de la carretera (rojo) con ciertos márgenes de seguridad para reducir el espacio libre de colisión. En la Figura 168 (b) se observa el resultado final obtenido tras haber realizado la proyección inversa sobre el plano de la imagen tanto de los puntos centrales (puntos verdes) detectados sobre el plano de movimiento del vehículo (ver Figura 169), como de la estimación del camino mediante RBC (línea azul y círculos cian).



Figura 168. Ejemplo de carril de carretera: (a) imagen con el ROI (cuadro azul) y la región de interés resaltada en rojo, (b) resultado final con los puntos centrales (verde) y la estimación del camino con la RBC (azul).

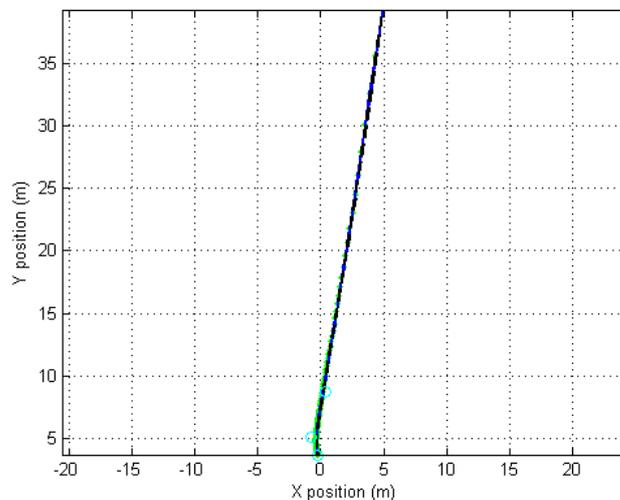


Figura 169. Aproximación de los puntos de la carretera mediante una RBC cúbica.

#### ***4.4. Algoritmo de Detección de Líneas para Sistemas de Visión Embebidos (con Recursos Limitados)***

En esta sección se describe un algoritmo para la detección de líneas y obtención de sus parámetros característicos mediante sistemas embebidos o con recursos limitados. Al igual que en el apartado anterior el fin de dicha detección es realizar un seguimiento de línea con un vehículo auto-guiado o con un robot.

El algoritmo ha sido específicamente diseñado para poder procesar una imagen en escala de grises de  $176 \times 144$  píxeles, que corresponde al modo de baja resolución con formato *YCrCb*, aunque sólo se trabaja con el canal *Y* de intensidad de gris. La CMUcam3 [108] tiene una limitación importante en cuanto a capacidad de RAM disponible, que en la práctica hace que podamos trabajar sólo con tres filas como máximo, condicionando significativamente la implementación del algoritmo realizada. Se parte de las siguientes premisas de partida:

- La orientación de la línea es generalmente vertical y por tanto el algoritmo de detección funcionará mejor cuanto más vertical sea la línea en la imagen. Si bien, en la práctica, el algoritmo ha demostrado poder llegar a detectar líneas prácticamente horizontales, aunque con mayor dificultad.
- El fondo y la línea no tienen por qué estar lo suficientemente contrastados. Generalmente la línea se asume de color negro mientras que el fondo puede ser de cualquier otro color y dado que no se están utilizando filtros de color, el fondo se verá generalmente en gris. Por ello, se utilizará en la etapa de procesado una compensación del fondo para filtrar, en la medida de lo posible, este efecto e indirectamente el problema de la iluminación.
- El fondo de la imagen puede contener suciedad (por ejemplo una mancha de aceite) u objetos que puedan perturbar la percepción de la línea.
- Se pueden disponer en una misma imagen de varias líneas, ya sea por bifurcaciones, tramos curvos, etc.

Para realizar un esbozo de la estructuración del código, en la Figura 170 se observa el diagrama de flujo del bucle principal de la aplicación. Se incluyen tanto funciones que se ejecutan en la cámara, como el proceso de comunicación o funciones que se ejecutan en la aplicación con el PC industrial.

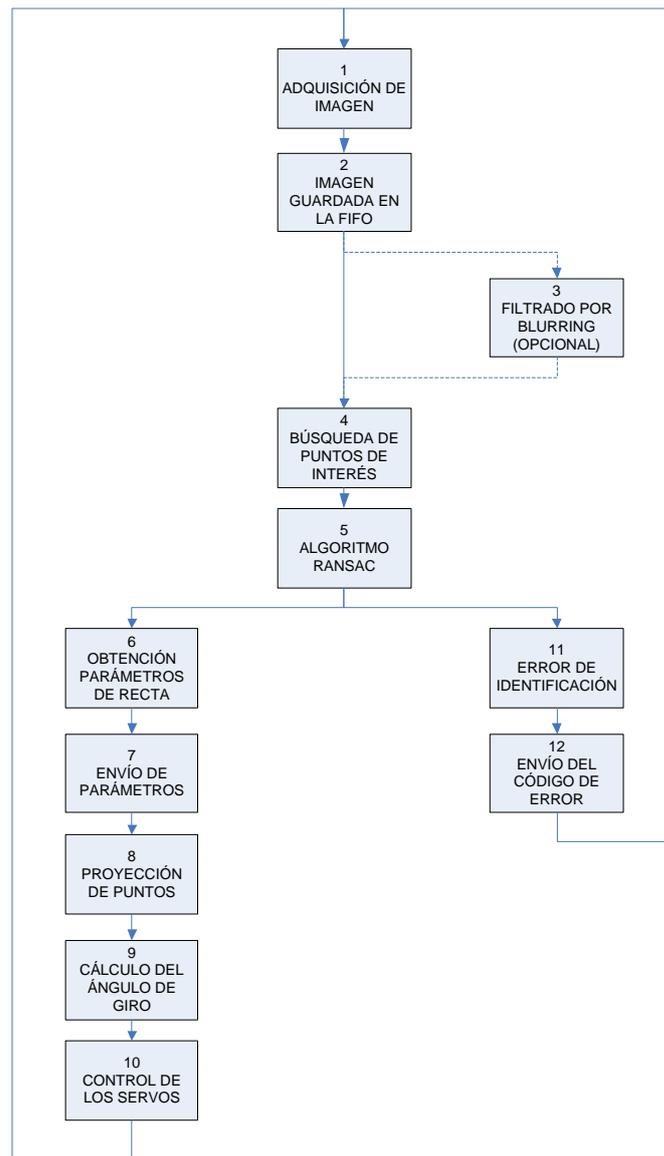


Figura 170. Diagrama de flujo principal de la aplicación.

Se han numerado los bloques con tal de detallar los distintos pasos que se siguen en cada periodo de control durante el proceso de seguimiento o *tracking*. Dichos bloques se explican a continuación:

1. Se da la orden a la cámara para realizar una captura de imagen (la cámara dispone de comandos en lenguaje C que sirven para eso).
2. Se almacena la imagen en la memoria FIFO, de forma automática y totalmente transparente al usuario y a la aplicación. No obstante si se desea y se configura debidamente se pueden almacenar las imágenes en una memoria SD.
3. Dada la posibilidad de que el fondo esté manchado o poco contrastado se tiene la opción de realizar un filtrado con eliminación de fondo que se basa en el principio de emborronamiento de la imagen y que es equivalente a una máscara *min/max*, con conectividad 4.

4. La búsqueda de puntos de interés se basa en la propiedad de la recta que tiene discontinuidad de tipo rampa (descendente y ascendente) y se calcula la primera derivada mediante el gradiente entre píxeles separados  $N$  posiciones. Para ello se implementa una secuencia de instrucciones que determinan si se cumplen las condiciones de que 2 puntos de una misma fila pertenecen a la recta y así dar mayor robustez a la detección que si simplemente se utilizara el gradiente.
5. El algoritmo RANSAC es el encargado de eliminar los puntos detectados que no son *inliers*.
6. Se realiza la estimación de los parámetros de la recta mediante regresión lineal ortogonal.
7. Se envían a la aplicación del PC los parámetros estimados mediante el protocolo de comunicación que se ha establecido.
8. Se realiza la proyección de los puntos detectados y se recalculan los parámetros de la recta ( $\rho$  y  $\phi$ ) de la recta real, no la de la imagen.
9. Se calcula el ángulo que se debe de girar según el algoritmo de persecución pura adaptado para este problema.
10. Se realiza el control de los servos enviando los distintos ángulos a la cámara mediante la comunicación serie.
11. En caso de que el algoritmo RANSAC produzca errores se codifica dicho error.
12. Se envía un telegrama comunicando que ha ocurrido un error y qué código tiene, de modo que se puede corregir.

A continuación se explicarán los bloques más importantes dentro del algoritmo diseñado. Estas funciones son las de filtrado por *blurring*, detección de bordes y algoritmo RANSAC. El motivo es que son las que mayor relación tienen con el procesamiento de la imagen y con la parte de detección de la línea. No obstante los siguientes apartados se dedican para explicar la obtención de los parámetros de la recta en la imagen, así como la proyección de los puntos para obtener los valores reales de la recta. También se explica cómo se realiza el cálculo de la acción de control para el seguimiento basado en el algoritmo de persecución pura y por último se explicará el protocolo de comunicación.

El primer paso del algoritmo corresponde a una sencilla etapa de compensación de fondo en la que se aplica una máscara cuyo objetivo es reemplazar el valor de cada píxel por el valor máximo/mínimo de sus vecinos (conectividad 4). En este sentido primero se aplica una máscara de mínimo y luego de máximo con los píxeles más cercanos para eliminar las motas o pequeñas suciedades del fondo. Posteriormente se aplica una máscara de máximo y una de mínimo con vecinos lo suficientemente alejados (para nuestro caso se han escogido vecinos que están a una distancia de 10 píxeles) y cuyo objetivo es precisamente obtener una imagen uniforme con el color del fondo. Finalmente se resta a la imagen original la imagen con el color de fondo, se multiplica la diferencia de píxeles por una ganancia para resaltar más aún esta diferencia y se invierte la imagen (blancos pasan a negros y viceversa).

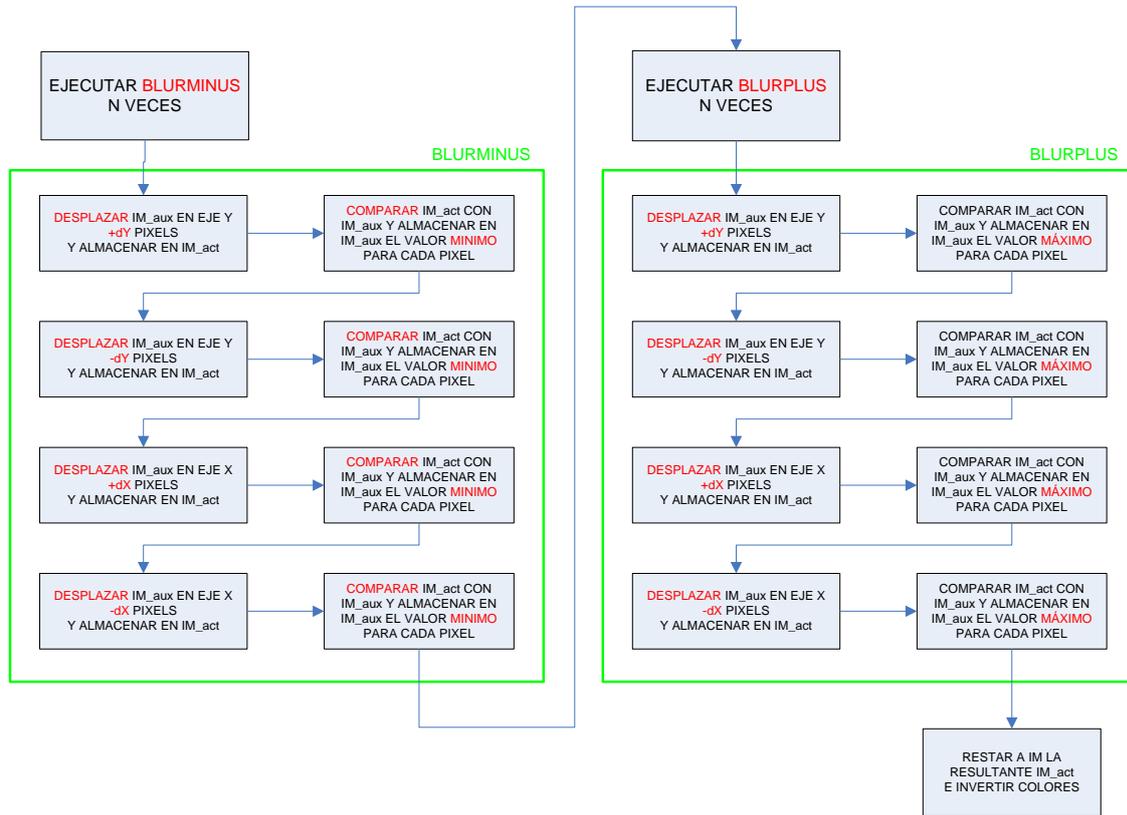


Figura 171. Diagrama de flujo de la función de eliminación de fondo mediante *blurring*.

Se enumeran las distintas simplificaciones en la nomenclatura que se han realizado para hacer más fácil de entender el diagrama de flujo que aparece en la Figura 171.

- **IM**: Imagen Original.
- **IM\_act**: Imagen Actual.
- **IM\_aux**: Imagen Auxiliar.

En la Figura 172 se muestran dos imágenes a las cuales se les ha aplicado la compensación de fondo.

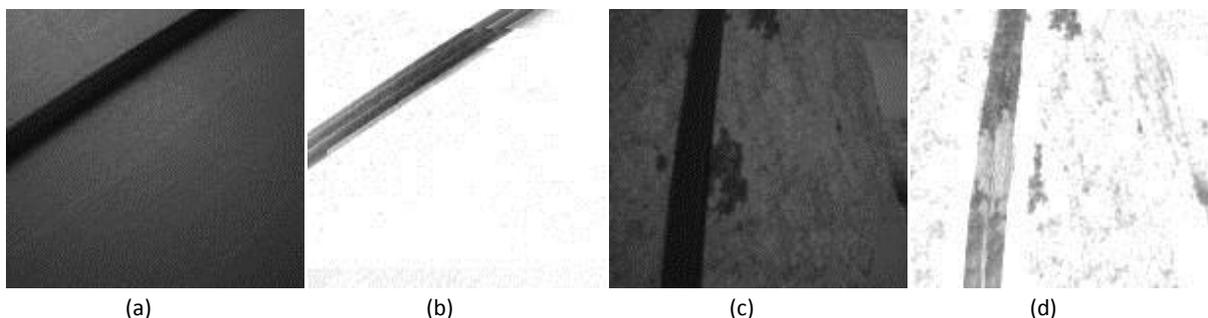


Figura 172. Ejemplos de compensación de fondo. (a) y (c) imágenes originales con fondo gris claro y gris moteado, respectivamente. (b) y (d) resultado de la imagen con el fondo compensado.

Hay que destacar que aunque se ha implementado el filtro anterior y las mejoras en fondos manchados son bastante buenas en cuanto a robustez, el hecho de que el tiempo de cómputo sea

notablemente mayor, hace que se opte por no utilizar normalmente esta fase de pre-procesamiento de la imagen.

A continuación se realiza un umbralizado, en el que se buscan los posibles puntos que pertenecen a la recta a detectar. Para ello, se realiza una búsqueda de cambios de contraste por cada fila. Se trata de un sencillo proceso en el que busca tanto por la izquierda de la imagen como por la derecha, buscando el primer píxel que genera una diferencia relevante en cuanto a su intensidad en escala de grises. Aquella pareja de píxeles que genera un flanco descendente (de blanco a negro) y ascendente (de negro a blanco), es candidata a ser punto del borde de la línea. Si al proyectar los puntos sobre el suelo (plano  $z = 0$ ), la pareja de puntos está a una distancia similar al grosor de la línea, entonces establecemos que son unos buenos candidatos. La matemática de la proyección de puntos de la imagen al plano del suelo se describe con mayor nivel de detalle en el Anexo A.

A continuación se enumeran las distintas simplificaciones en la nomenclatura que se han realizado para hacer más fácil de entender el diagrama de flujo que aparece en la Figura 173.

- **GD**: Gradiente Descendente.
- **GA**: Gradiente Ascendente.
- **N**: Nº de Píxeles anteriores al actual.
- **DP**: Distancia Proyectada.
- **DO**: Distancia Óptima.

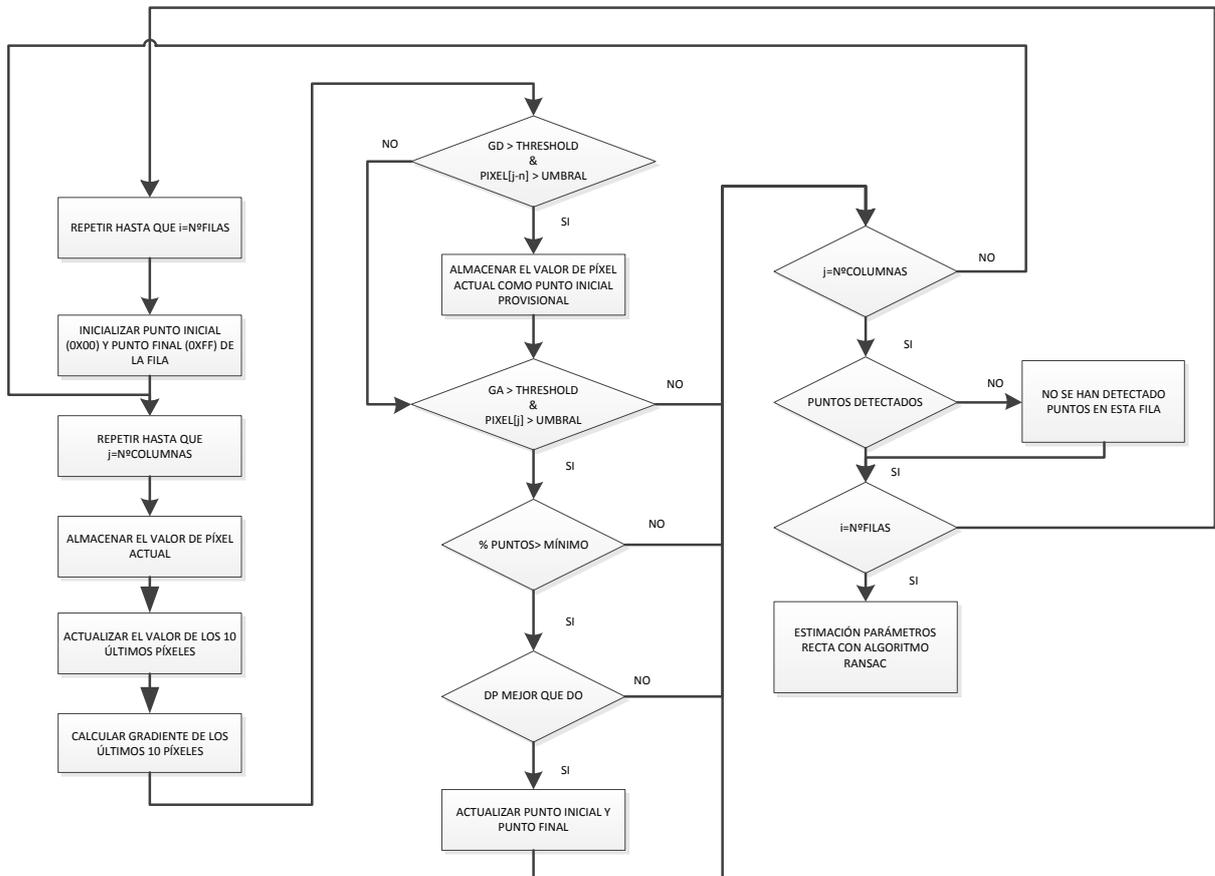


Figura 173. Diagrama de flujo de la función de detección de bordes.

Una vez se han detectado los puntos que son candidatos a pertenecer a la recta que se está buscando es necesario determinar si realmente pertenecen o no. Para ello se utiliza el algoritmo RANSAC, cuyo funcionamiento se ha explicado muy por encima en apartados anteriores. A continuación se muestra el pseudocódigo y una descripción detallada de cada uno de sus pasos:

1. Compensación del fondo de la imagen a partir de operaciones consecutivas de máximos y mínimos con los píxeles vecinos.
2. Para cada fila, determinar los bordes de la línea basándose en la diferencia de intensidad de los píxeles vecinos (búsqueda por la izquierda y la derecha de la imagen).
3. Repetir el proceso los pasos 4 a 12 de forma separada para los bordes de la izquierda y para los bordes de la derecha.
4. Inicializar un contador  $cnt = 0$ , el número de iteraciones a ejecutar  $N = 1$ , el máximo número de iteraciones permitidas  $N_{max}$  y la distancia umbral  $d$  para considerar un punto como *inlier*.
5. **Mientras ( $N > cnt$ )**
6. Seleccionar aleatoriamente dos filas.
7. Calcular los parámetros de la línea  $\rho$  y  $\phi$  que pasen por los puntos seleccionados.
8. Para el resto de los puntos, calcular la distancia ortogonal  $d_p$  a la línea calculada.
9. Clasificar como *Inliers* aquellos puntos cuya distancia  $d_p < d$  y calcular el número de *Inliers*  $N_i$ .
10. Si  $N_i > maxN_i$ , entonces guardar como solución final la solución actual ( $maxN_i = N_i, \rho$  y  $\phi$ ) y recalculer el número de iteraciones que restan  $N = \log(p) / \log(1 - (N_i/N_p)^2)$ , siendo  $p = 0.01$  la probabilidad esperada de tener un espúreo no detectado y  $N_p$  el número total de puntos.
11. Incrementar el contador  $cnt++$  y si  $cnt > N_{max}$ , entonces salir con fallo.
12. **Fin Repetición.**
13. Calcular los parámetros de la línea con Regresión Ortogonal sólo con los puntos *Inliers*.
14. Calcular la línea intermedia.

Tabla 11. Algoritmo utilizado para la obtención de los parámetros característicos que definen la recta detectada.

El hecho de que este procedimiento sea sencillo y proclive a detectar una gran cantidad de píxeles espurios, hace necesario la inclusión de algún sistema capaz de separar los puntos buenos de los que no lo son. Los pasos del 3 al 12 corresponden al algoritmo RANSAC [105], [106], [107], para el filtrado de datos espurios, de modo que se obtiene un subconjunto de puntos que se ajusta al modelo de una recta (*inliers*). Una vez seleccionados los *inliers* el algoritmo finaliza calculando la recta que mejor se ajusta a estos puntos. La recta está definida según el conocido modelo ángulo-distancia:  $\rho = x \cos \phi + y \sin \phi$ .

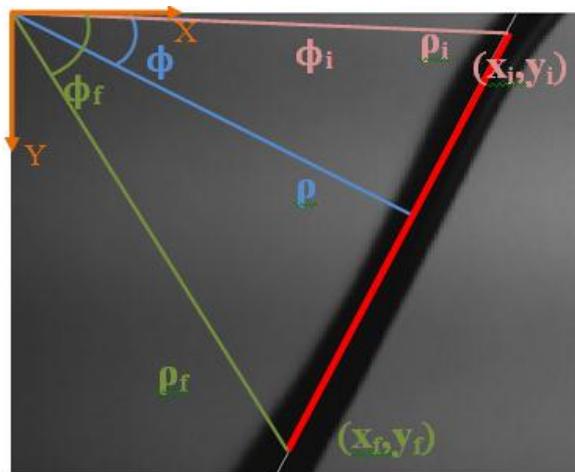


Figura 174. Representación de la recta en el espacio de distancia-ángulo.

Siendo  $(x_i, y_i)$  y  $(x_f, y_f)$  las coordenadas cartesianas del punto inicial y final respecto al sistema de referencia de la imagen;  $\rho_i$  la distancia respecto al sistema de referencia de la cámara del punto inicial;  $\rho_f$  la distancia respecto al sistema de referencia de la cámara del punto final;  $\phi_i$  es el ángulo respecto al eje  $x$  del punto inicial;  $\phi_f$  es el ángulo respecto al eje  $x$  del punto final; y  $\rho$  y  $\phi$  son los parámetros que definen la recta.

Por lo tanto, la recta que mejor se ajusta a todos los *inliers* es aquella que minimiza el siguiente índice:

$$J = \frac{1}{2} \sum_{k=0}^{N-1} (x_k \cdot \cos(\phi) + y_k \cdot \sin(\phi) - \rho)^2 \quad (168)$$

Siendo  $N$  el número total de puntos, y cuya solución es bien conocida [109]:

$$\frac{\partial J}{\partial \rho} = 0 \rightarrow \hat{\rho} = \bar{x} \cdot \cos\phi + \bar{y} \cdot \sin\phi \quad (169)$$

$$\frac{\partial J}{\partial \phi} = 0 \rightarrow \hat{\phi} = \frac{1}{2} \cdot \arctan\left(\frac{-2 \cdot \sigma_{xy}^2}{\sigma_y^2 - \sigma_x^2}\right) \quad (170)$$

A partir de los parámetros característicos de la recta se pueden recalcular, de forma sencilla, los puntos extremos de la recta  $(x_i, y_i)$  y  $(x_f, y_f)$ , teniendo en cuenta las siguientes expresiones:

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \cdot \begin{bmatrix} \rho \\ \rho \cdot \tan(\phi_i - \phi) \end{bmatrix} \quad (171)$$

$$\begin{bmatrix} x_f \\ y_f \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \cdot \begin{bmatrix} \rho \\ \rho \cdot \tan(\phi_f - \phi) \end{bmatrix} \quad (172)$$

En la Figura 175 se muestran algunos ejemplos de identificación con el método propuesto, demostrándose que el algoritmo de detección de líneas es robusto frente a manchas en el suelo, distintas posiciones, orientaciones y anchuras de la línea, etc.

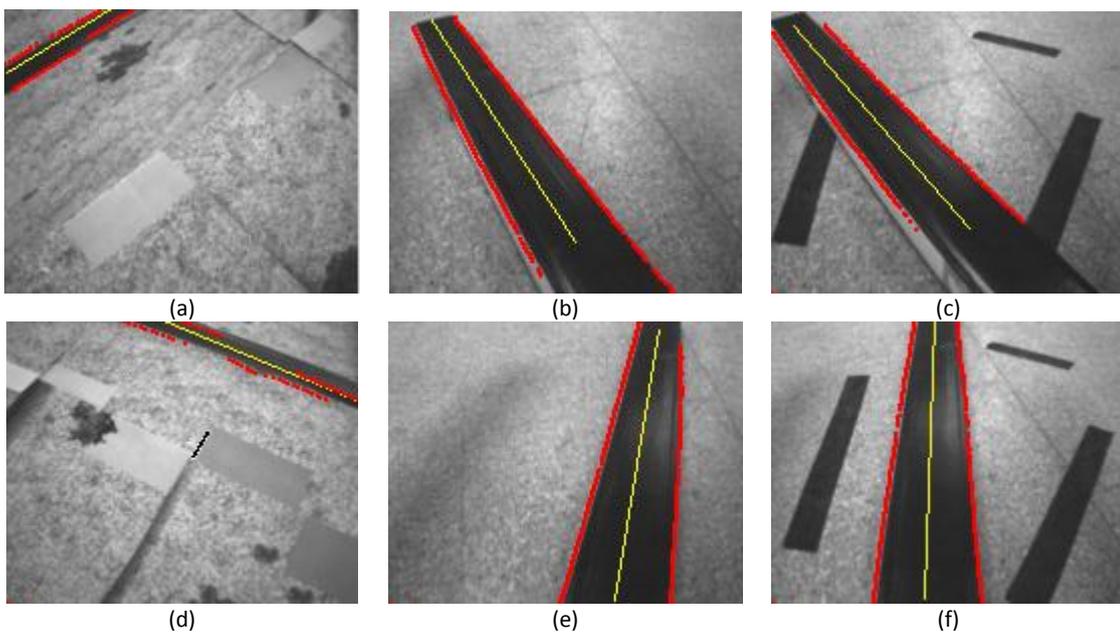


Figura 175. Ejemplos de detección de líneas.

Como se observa en los ejemplos anteriores, el algoritmo es capaz de detectar líneas en situaciones complicadas, como por ejemplo en posiciones muy horizontales, en un extremo de la imagen y con manchas Figura 175 (a) y (d). También es robusto frente a distintas anchuras e inclinaciones, como se ve en la Figura 175 (b) y (e). Finalmente se observa que para imágenes de complejidad similar a las anteriores, pero con manchas en el suelo del mismo color que la línea pero de distinta anchura, el sistema es capaz de detectar perfectamente la línea buscada (ver Figura 175 (c) y (f)).

Una vez obtenidos los puntos extremos de la línea definidos con respecto al sistema de coordenadas de la imagen, hay que proyectarlos sobre el plano  $z = 0$ , que es el plano sobre el que está pintada la línea que debe seguir el vehículo. Dicho procedimiento es similar al realizado en el apartado anterior, salvo que ahora la calibración es para la torre PTU sobre la cual va montada la CMUcam3.

Finalmente, para la obtención del punto objetivo se procede igual que en el método completo, es decir, por geometría se obtiene la intersección de la línea con un arco a una distancia  $LA$  del sistema de coordenadas local del robot.

Tal y como se deduce de lo que se ha comentado anteriormente la cámara con capacidad de procesamiento es un sensor inteligente capaz de detectar la línea y determinar qué puntos son *inliers* y cuáles no. Además se encarga directamente de calcular los parámetros de distancia y ángulo, así como los puntos inicial y final de la recta detectada, todo ello referenciado al sistema de coordenadas de la imagen.

Para el caso particular del sistema embebido, teniendo en cuenta el vehículo móvil sobre el que se va a montar el sistema de visión, en este caso la CMUcam3, es necesario establecer una comunicación entre el ordenador y el sensor inteligente, ya que no será el dispositivo quién controlará el movimiento del vehículo. Esto se debe a que el ordenador actúa en este caso como sistema principal de control, limitando las funciones de la cámara a la detección de la línea y al envío de datos. Por lo tanto, tras procesar los datos recibidos, el ordenador envía una acción de control determinada para posicionar la cámara y en último término el vehículo móvil.

Por las características del hardware utilizado, se ha realizado la comunicación entre ambos dispositivos a través del puerto serie del ordenador. Respecto a la comunicación hay que decir que se ha implementado uno protocolo sencillo que se comenta brevemente en el Anexo 5.

# **CAPÍTULO 5**

## **Implementación de las Técnicas Propuestas**



## 5. Implementación de las Técnicas Propuestas

### 5.1. Sistema de Visión Activa

Con tal de dotar al sistema de adquisición y procesamiento de imagen de un mayor campo de visión, es interesante poder controlar la orientación de la cámara. Con esto se consigue por ejemplo tener más grados de libertad en el problema de seguimiento de líneas, de modo que sea más difícil que el camino desaparezca del campo de visión de la cámara. Es por eso que en los desarrollos realizados se ha decidido utilizar diferentes sistemas de visión activa, como unidades PTU (*Pan-Tilt Unit*) o sistemas PTZ (*Pan-Tilt Zoom*).



Figura 176. (a) Torre PTU para la CMUcam3. (b) Cámara IP con sistema PTZ.

Para la elaboración del sistema de visión activa (o cabeza inteligente), se han tenido en consideración diferentes aspectos, que se muestran a continuación organizados según la fase de un sistema de CV a la que pertenecen.

- Sistema de adquisición de imágenes:

En esta sección se analizarán los elementos encargados de adquirir las imágenes. Aunque a priori hay diferentes tipos de cámaras comerciales con distintas prestaciones, por su versatilidad, sencillez y bajo precio se han utilizado la CMUcam3 y una cámara IP. Además, para dotar de más grados de libertad se opta por utilizar una torreta PTU de modo que la cámara CMUcam3 sea capaz de moverse independientemente del vehículo. Para la cámara IP no es necesario montar ningún sistema puesto que dispone de dos motores que le dan la capacidad de ser un sistema PTZ. Este punto formaría parte de la etapa sensorial de un sistema típico de CV.

- Algoritmo de detección de ruta en la imagen:

Una vez captada la imagen se pasa a la extracción de características de ésta, es decir, el procesado para obtener los parámetros de la ruta a seguir. Se analizan varias posibilidades y algoritmos, para finalmente explicar la solución adoptada. Hay que destacar que al utilizar la CMUcam3 el procesamiento lo hace mayoritariamente el mismo sistema embebido, utilizándose el PC para visualizar las imágenes y realizar el control del vehículo. En cambio, al utilizar la cámara IP, que no dispone de capacidad de procesamiento, se opta por utilizar el ordenador como único elemento de cómputo.

- Representación de la orientación y calibración:

Para efectuar el seguimiento de la ruta es necesario expresar los parámetros de ésta respecto de las coordenadas del vehículo. Aunque existen diferentes técnicas de representación de la orientación, se utiliza la más sencilla e intuitiva para relacionar las coordenadas referidas al elemento sensor con respecto a las del AGV, es decir, las matrices de transformación homogénea o MH.

Para la implementación del control utilizando la carretilla industrial se opta por realizar una calibración mediante los parámetros de distancia y ángulos medidos sobre el mismo vehículo.

En cambio, al utilizar el robot *Pioneer* no se realiza una calibración completa, sino que se obtiene la matriz de transformación homogénea a partir de una Homografía, puesto que es la forma más sencilla de relacionar los puntos situados en dos planos distintos.

- Sistema de comunicación y visualización:

Una vez identificada la información aportada por las imágenes, será necesario transmitirla al exterior, ya sea, hacia el elemento de control del AGV o a un sistema de visualización. Para ello, cuando se utiliza la CMUcam3 se opta por un sencillo protocolo de comunicación serie RS232 que se explica en el Anexo 5. En cambio, al utilizar la cámara IP se opta por utilizar el protocolo TCP/IP con comandos CGI que facilita el mismo fabricante.

## 5.2. Experimentación con CMUcam3 y AGV Industrial

En primer lugar se explicará la implementación realizada sobre el AGV industrial que dispone el IDF de la UPV. Se trata de un vehículo eléctrico de tipo carretilla elevadora modelo *Nichiyu FBT-15P*. Tiene una configuración triciclo, con una rueda directriz trasera y dos ruedas delanteras para la tracción. En la Figura 177 se presenta el vehículo utilizado con los distintos elementos de sensorización que incorpora.

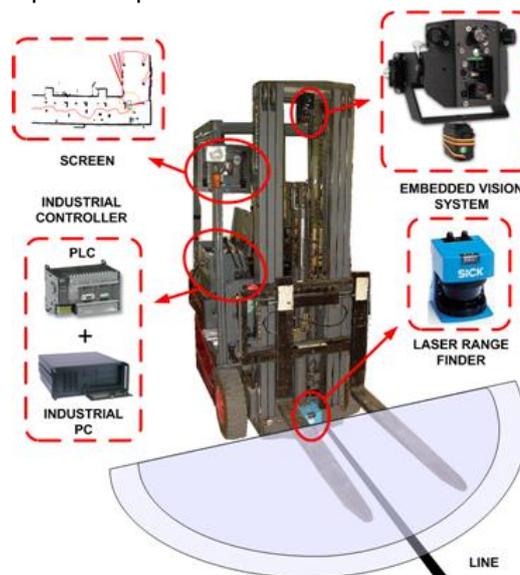


Figura 177. AGV industrial utilizado en la experimentación de seguimiento de líneas.

Antes de proceder a realizar el control sobre el vehículo real hay que probar el algoritmo ante diferentes condiciones, con objeto de comprobar su robustez y precisión y así validar la implementación, igual que se hizo en [110] y posteriormente en [111].

En concreto, se ha probado su funcionamiento para detectar líneas de distintos colores, con mayor y menor contraste y con objetos introducidos a modo de perturbación. En la Figura 22 (a) se muestra un caso en el que la línea es clara frente a un fondo oscuro, en el que se observa que funciona correctamente, tal y como se muestra en la Figura 22 (b).

En las Figura 22 (c), (g), (k) se prueba el algoritmo con imágenes contaminadas por objetos que no pertenecen a la línea. Se observa que el RANSAC, ha sido capaz de filtrar los puntos espurios, dando lugar a una correcta detección de la recta (ver Figura 22 (d), (h), (l)).

Por último, se ha estudiado el efecto de procesar una curva o rectas a tramos (ver Figura 22 (e), (i)). Se observa que en esta ocasión han sido filtrados gran parte de los puntos espurios, obteniendo de nuevo un buen ajuste de la recta.

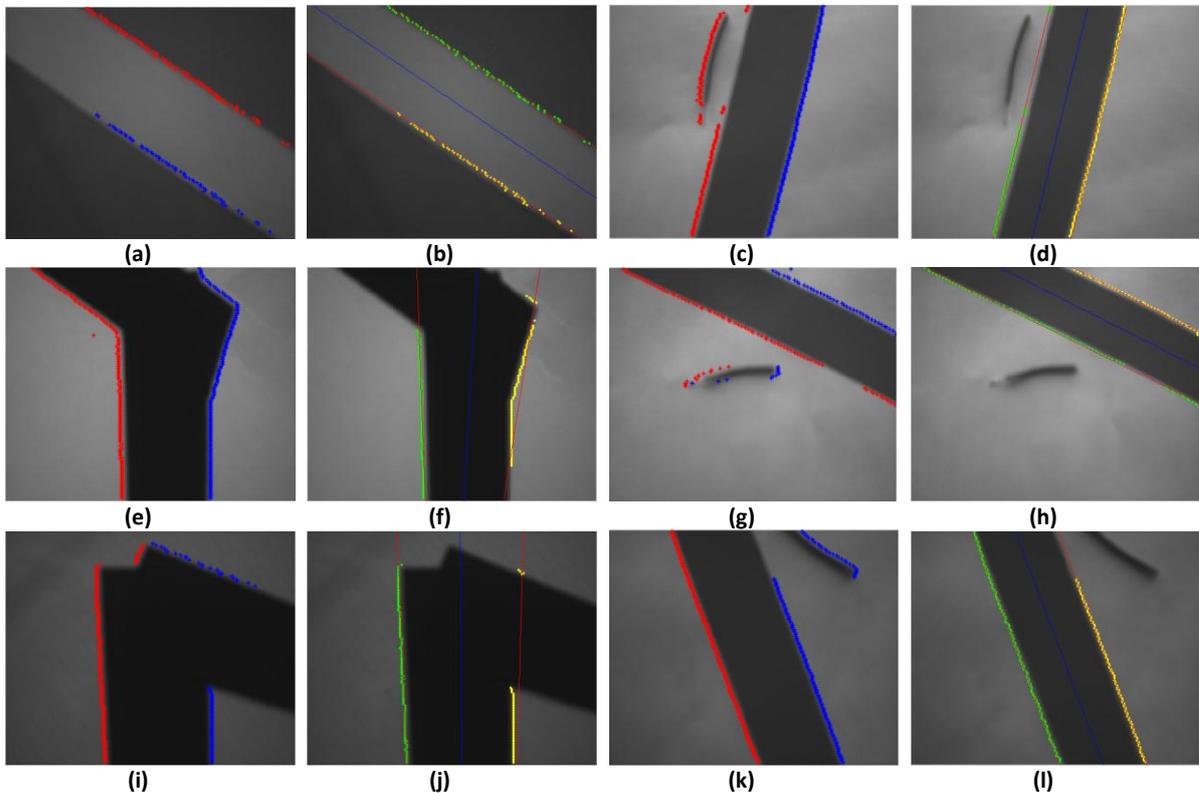


Figura 178. Representación de la recta en el espacio de distancia-ángulo.

Así mismo, se ha realizado un análisis cuantitativo exhaustivo para el cual se ha utilizado un robot industrial sujetando la cámara y una mesa giratoria sincronizada y debidamente calibrada con el robot sobre la cual se ha posicionado una línea con diferentes fondos. Esto nos ha permitido generar una combinación de movimientos tanto desplazamientos del robot ( $R_x$ ) con incrementos de 30 mm, como giros de la mesa giratoria ( $\psi$ ) con incrementos de  $10^\circ$ , conociendo en todo momento la posición (ángulo y distancia) de la línea, esto es sus valores verdaderos.

El objetivo del experimento es analizar la robustez y precisión del algoritmo para diferentes rangos de desplazamientos del robot como de giros de la mesa. Por ello, se ha dividido en tres rangos para los cuales se tienen giros y desplazamientos pequeños ( $\psi \in [-10^\circ, 10^\circ]$  y  $R_x \in [-30mm, 30mm]$ ), medianos ( $\psi \in [-40^\circ, 40^\circ]$  y  $R_x \in [-90mm, 90mm]$ ) y grandes ( $\psi \in [-70^\circ, 70^\circ]$  y  $R_x \in [-150mm, 150mm]$ ).

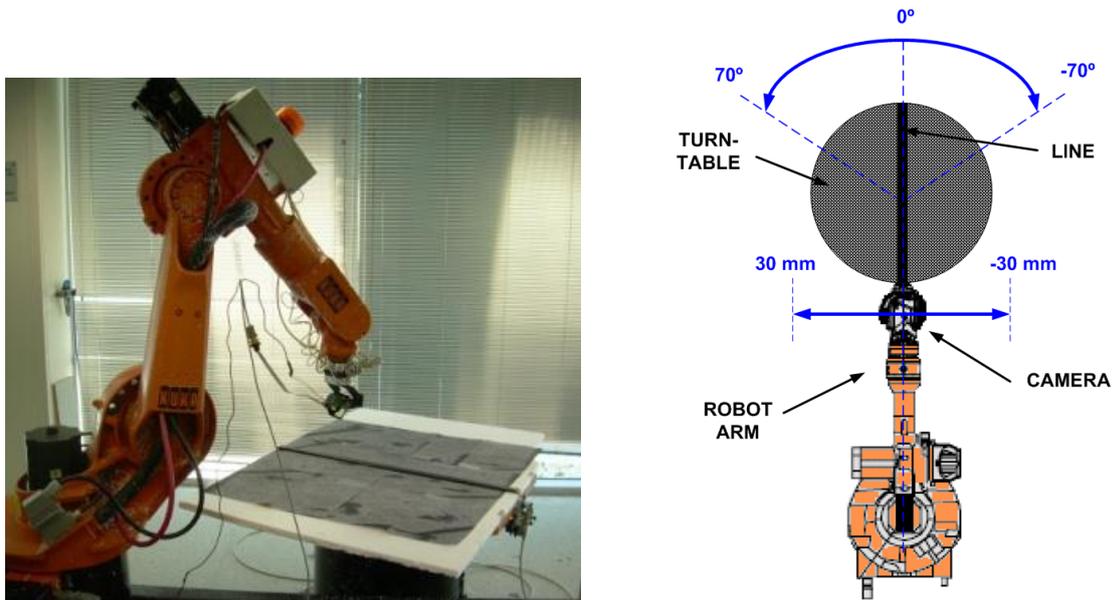


Figura 179. Experimento para la obtención de resultados que sirvan como *ground-truth* al utilizar el sistema de visión CMUcam3 junto con un robot Kuka KR/15 y una mesa giratoria.

Para el análisis de la robustez, se ha considerado un verdadero positivo (*True Positive* o TP) aquellos casos cuyo error de estimación de la línea ( $\epsilon_\rho$  y  $\epsilon_\phi$ ) es  $\epsilon_\rho \leq 60mm$  en distancia y  $\epsilon_\phi \leq 6^\circ$  en ángulo. Por otro lado se ha considerado un falso positivo (*False Positive* o FP) aquellos casos que se ha detectado una línea y cuya estimación ha sido  $\epsilon_\rho > 60mm$  o  $\epsilon_\phi > 6^\circ$ . Finalmente se considera un resultado negativo (*Negative* o N) cuando el algoritmo no ha sido capaz de detectar una línea, generalmente por falta de puntos en los bordes de la línea.

Las Tablas 12 y 13 muestran los resultados de robustez obtenidos para el caso de detección línea sobre fondo gris claro (el caso más favorable) y sobre un fondo gris moteado de diversos tamaños (véase las Figura 180 (a) y (b)) como ejemplo de imágenes utilizadas).

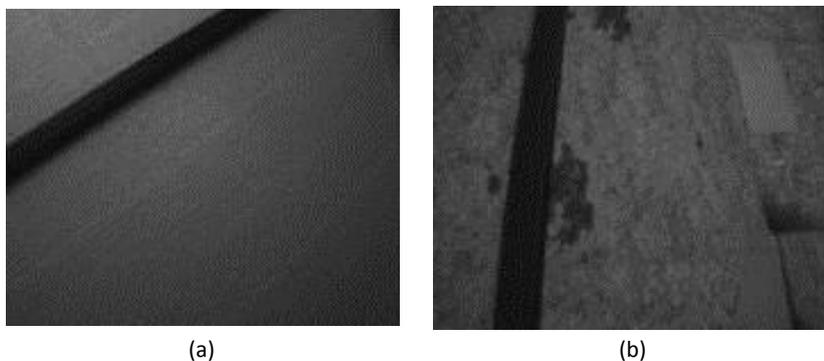


Figura 180. (a) Ejemplo de imagen de línea con fondo gris claro. (b) Ejemplo de imagen con fondo gris moteado.

En las tablas se puede observar que tanto para fondo blanco como para fondo gris moteado el porcentaje de TP es muy elevado, siempre por encima del 98%, incluso para los casos más complejos en los que tanto la inclinación de la línea como su separación hacen especialmente dificultosa la problemática.

Además de considerar experimentos con distintos fondos se han realizado pruebas para verificar la robustez y precisión ante errores de calibración. Se ha tomado como parámetro variable el ángulo  $\beta$  (ver Figura 181), ya que en la aplicación permanece fijo y es el que más afecta en la precisión de los resultados. Por otro lado, el ángulo  $\alpha$  (ver Figura 181) es el otro parámetro angular que interviene, pero éste es variable durante la aplicación, puesto que se trata de la variable de control, y por lo tanto sus errores son compensados por la ley de control de persecución pura, que hará que en cualquier caso el ángulo de giro converja al ángulo real de la línea.

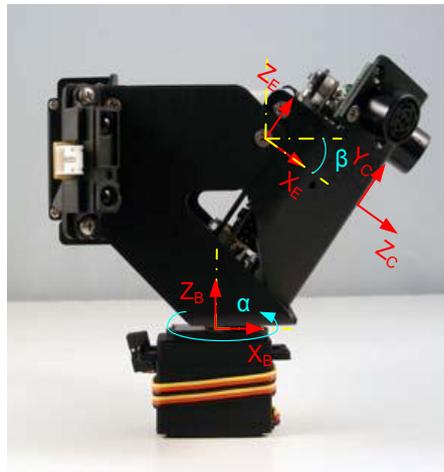


Figura 181. Parámetros angulares para calibración de la PTU

El resto de parámetros de distancia no tiene mucho sentido considerarlos para el análisis frente a errores de calibración, puesto que las distancias son fáciles de medir y en caso de haber errores serían despreciables, ya que estarían en torno a los milímetros, mientras que en la aplicación se trabaja con distancias incluso de metros al aplicar el algoritmo sobre el AGV.

Fondo Gris Claro							
# Imágenes	Giros y Desplazamientos	ROBUSTEZ			PRECISIÓN		
		TP	FP	N	H	M	L
9	Pequeños	100	0	0	100	0	0
63	Medianos	100	0	0	100	0	0
165	Grandes	100	0	0	98.18	1.82	0

Tabla 12. Robustez y precisión del algoritmo sobre fondo gris claro

Fondo Gris Moteado							
# Imágenes	Giros y Desplazamientos	ROBUSTEZ			PRECISIÓN		
		TP	FP	N	H	M	L
9	Pequeños	100	0	0	88.89	11.11	0
63	Medianos	98.41	1.59	0	58.07	40.32	1.61
165	Grandes	98.18	1.82	0	53.70	45.68	0.62

Tabla 13. Robustez y precisión del algoritmo sobre fondo gris moteado

En las tablas anteriores, tal y como ya se ha mencionado antes, se ha realizado un estudio de la robustez y de la precisión del algoritmo para distintos fondos y considerando errores de calibración. A continuación se procede a extraer las conclusiones oportunas al respecto:

- Los mejores resultados tanto de robustez como de precisión son siempre cuando el fondo es gris claro, es decir, cuando hay mayor contraste entre la línea y el fondo y no aparecen manchas que puedan afectar negativamente.
- A medida que aumenta la dificultad del experimento, es decir, cuando los giros y desplazamientos se hacen más grandes, se pierde precisión y también robustez, aunque en el segundo caso se mantiene en niveles muy elevados, en general superiores al 98% de *True Positives*.
- Se consigue detectar la línea el 100% de los experimentos, puesto que no aparecen ningún experimento con valores *Negative*. Esto significa que siempre que hay línea el algoritmo es capaz de detectarla.
- Si se considera la precisión H y M en conjunto, se consigue tener una buena precisión en más del 98% de los casos.
- Los errores de calibración afectan tanto a la precisión como a la robustez, pese a que los niveles de robustez tienen poca sensibilidad ante dichos errores.

Para finalizar este primer estudio cuantitativo se va a realizar un análisis de tiempos, para detallar los resultados obtenidos y así comprobar la versatilidad del algoritmo. Lo único que hay que destacar es que los tiempos de procesamiento y comunicaciones no superan, en el peor de los casos, los 130 milisegundos de media aproximadamente, con 165 pruebas realizadas ante diferentes condiciones.

Si se analiza la etapa del procesamiento exclusivamente se obtiene que los tiempos medios son similares en ambos casos, ya que están en torno a los 40 ms. Concretamente, para el fondo gris claro y uniforme se obtiene un tiempo medio de cómputo de 37.0627 ms, mientras que para el fondo gris oscuro y con manchas se tiene una media de tiempo de 40.4957 ms. El resultado obtenido es lógico debido a que en el caso más complejo, entre otras cosas, el sistema pasa más tiempo en la etapa de eliminación de datos espurios del algoritmo RANSAC, lo cual incrementa el tiempo de procesamiento para cada periodo.

Por lo tanto, cuando se realiza el control para el seguimiento de la línea el algoritmo es capaz de trabajar a una frecuencia en torno a los 8 Hz (125 ms), lo cual es más que aceptable dada la elevada velocidad de convergencia del algoritmo (en iteraciones) y la velocidad del vehículo AGV, que no es muy elevada al tratarse de un vehículo industrial.

Para conocer el potencial del algoritmo de detección implementado y de su elevada robustez se va a mostrar un ejemplo de lo que se considera TP y FP. En la Figura 182 (a) aparece la detección de la línea en una situación complicada, dado que hay otras manchas de color y anchura similar a la línea buscada y también con forma alargada como una recta. Pese a las dificultades, el algoritmo es capaz de detectar correctamente la línea y estimar sus parámetros. En cambio, en la Figura 182 (b) aparece un caso de complejidad similar en el que el algoritmo comete un error por detectar incorrectamente la línea. Esto se debe a la orientación de la línea y

al hecho de que aparezca un objeto similar en color y anchura, que a diferencia de antes sí que acumula un número de puntos mayor que los de la propia recta y por eso se detecta como tal.

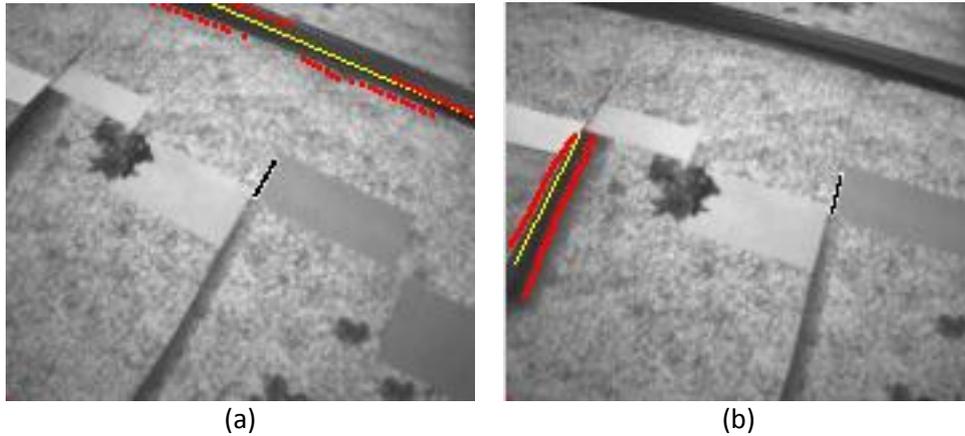


Figura 182. Ejemplo de (a) TP y (b) FP en una situación similar de elevada complejidad.

Además del anterior estudio se ha realizado otro análisis cuantitativo para verificar el correcto funcionamiento de la técnica propuesta. En este caso se busca cuantificar la sensibilidad frente a los errores de calibración en los ángulos de rotación de la torre PTU, tanto de la robustez como de la precisión.

En la gráfica de la Figura 183 el eje  $X$  corresponde al ángulo  $yaw$  y el eje  $Y$  al ángulo  $pitch$ . El valor de robustez se representa mediante colores, en el cual las líneas equipotenciales muestran regiones con el mismo nivel de robustez. Como se puede ver, el algoritmo es muy robusto, ya que más se detectan correctamente más del 95% de los casos TP en una gran región, con errores de  $\pm 4^\circ$  en ambos ángulos.

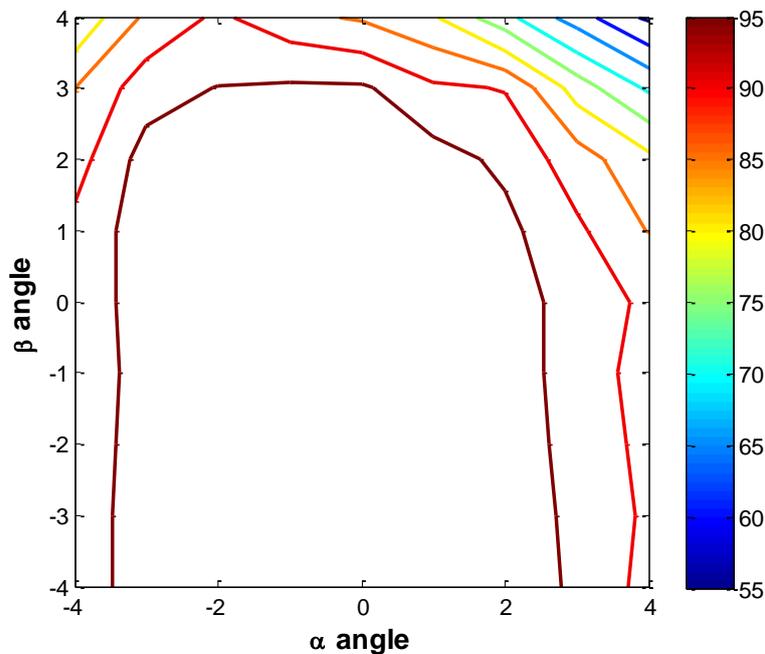


Figura 183. Sensibilidad de la robustez ante errores de calibración.

Como conclusión puede decirse que la robustez se ve afectada de forma simétrica respecto al ángulo *yaw*  $\alpha$ , con unos límites de  $\pm 4^\circ$  sin deteriorarse. Además, se observa que los errores de calibración del ángulo *pitch*  $\beta$  afectan más a la robustez, especialmente para casos en los que la cámara apunta hacia el horizonte, lo cual es lógico debido a la perspectiva de la imagen.

Con respecto a la precisión, en la Figura 184 se observa una gráfica similar a la de robustez. Las líneas equipotenciales representan el porcentaje de detecciones de elevada precisión. En este caso el valor de precisión decrece mucho más rápido y de forma similar en todas las direcciones. Como conclusión puede decirse que los errores de calibración en el ángulo *pitch* afectan más que los del ángulo *yaw*, puesto que el gradiente de degeneración de la precisión es mayor en dicha dirección. Por lo tanto, la correcta calibración es crucial en este tipo de aplicaciones, sobre todo para el ángulo *pitch*  $\beta$ .

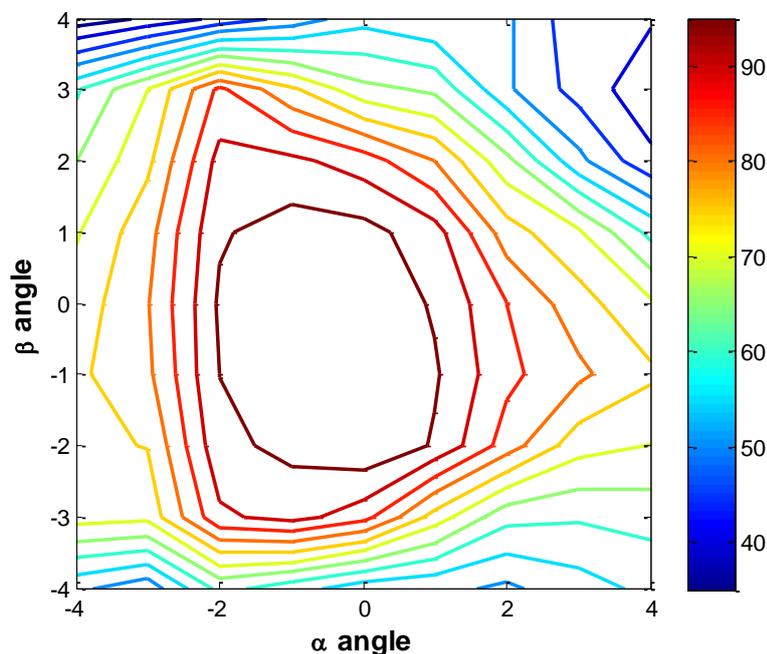


Figura 184. Sensibilidad de la precisión ante errores de calibración.

### 5.2.1. Problema de Seguimiento de Líneas

Una vez se ha demostrado la robustez y precisión del algoritmo de detección ya se puede realizar el control del robot. El problema a resolver se puede separar en dos partes, el seguimiento de la línea por parte del vehículo, que es el objetivo final, y el seguimiento de la misma por parte de la cámara mediante el sistema de visión activa, que es un medio para mejorar el seguimiento de la línea sin que llegue a salirse del campo de visión. Gracias a este elemento se mejoran situaciones anteriores en las que bien por elevadas velocidades del vehículo o por cambios bruscos en la dirección del camino seguido se perdía de vista la línea, provocando inestabilidad al sistema.

Por lo tanto, como solución desde un punto de vista del control, se plantea la inclusión de dos algoritmos de control independientes, uno para el seguimiento del sistema de visión activo (*Line Tracking*) y otro para el vehículo (*Line Following*). Se propone la utilización de técnicas de persecución pura como una primera aproximación al problema.

Para la unidad PTU o cualquier otro sistema de visión activa se propone esta ley de control:

$$\alpha = \tilde{\phi} - \arctan\left(\frac{\tilde{\rho}}{L}\right) \quad (173)$$

Dónde:

- $\alpha$  es el ángulo que se aplica al servo de la PTU o PTZ.
- $\tilde{\phi}$  es el ángulo estimado de la recta respecto al robot.
- $\tilde{\rho}$  es la distancia estimada de la recta respecto al robot.
- $L$  es el parámetro de control y representa una distancia objetivo.

Para el AGV o el robot a controlar se propone una ley de control de persecución pura para realizar el seguimiento con punto descentralizado:

$$\alpha = \tilde{\phi} - \arctan\left(\frac{2 \cdot E \cdot \tilde{\rho}}{L^2}\right) \quad (174)$$

Dónde:

- $\alpha$  es el ángulo que se aplica a la rueda directriz del robot.
- $\tilde{\phi}$  es el ángulo estimado de la recta respecto al robot.
- $\tilde{\rho}$  es la distancia estimada de la recta respecto al robot.
- $L$  es el parámetro de control y representa una distancia objetivo.
- $E$  es el parámetro de distancia del punto descentralizado.

### 5.2.1.1. Tracking de la Línea

Como ya se ha adelantado antes, una vez realizada la detección de la línea y la proyección de los puntos, lo primero que se hace es el seguimiento de la línea con la plataforma PTU, con objeto de tener la línea siempre dentro del campo de visión (FOV), independientemente de la orientación del vehículo. En este sentido, se propone una adaptación del algoritmo de Persecución Pura [102] para el cálculo de  $\alpha$  (guiñada) que se aplica al servo.

A partir de los puntos que definen el segmento de línea  $\tilde{\mathbf{P}}_i$  y  $\tilde{\mathbf{P}}_f$ , obtenemos la distancia de la base del robot a la línea  $\tilde{\rho}$  y el ángulo de la misma  $\tilde{\phi}$ :

$$\begin{aligned} \tilde{\rho} &= \cos(\tilde{\phi}) \cdot \tilde{x} + \sin(\tilde{\phi}) \cdot \tilde{y} \\ \tilde{\phi} &= \arctan\left(\frac{\Delta\tilde{y}}{\Delta\tilde{x}}\right) \end{aligned} \quad (175)$$

Donde  $\Delta\tilde{x} = \tilde{x}_i - \tilde{x}_f$  y  $\Delta\tilde{y} = \tilde{y}_i - \tilde{y}_f$ , siendo  $\tilde{\mathbf{P}}_i = [\tilde{x}_i \ \tilde{y}_i]^T$  y  $\tilde{\mathbf{P}}_f = [\tilde{x}_f \ \tilde{y}_f]^T$ .

La ley de control para el seguimiento de la línea está compuesta por dos términos de corrección: uno relacionado con la propia orientación de la línea y el otro relacionado con la

distancia de separación de la línea. Para el primer caso, se busca que la cámara tenga la misma orientación que la línea, mientras que para el segundo caso, se busca que la cámara apunte hacia un punto de la línea a una distancia  $L$  de la proyección de la cámara sobre la recta (ver ecuación (173)), tal y como se observa en la Figura 185.

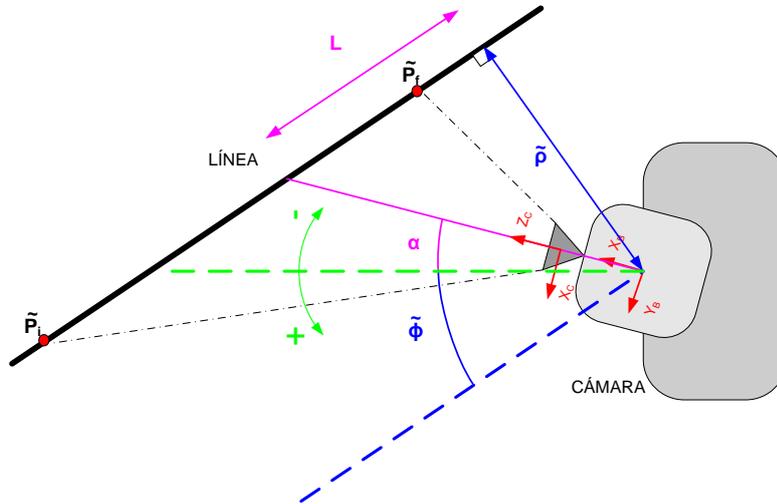


Figura 185. Proceso de detección y seguimientos de línea.

Como se observa en la Figura 186 se realiza un primer experimento en el que se mantiene  $\tilde{\rho} \approx 0$ , es decir, la línea pasa en todo momento por el origen del sistema de coordenadas de la cámara. En este caso, el ángulo girado es exactamente el que se detecta en la imagen (ambas respuestas están superpuestas).

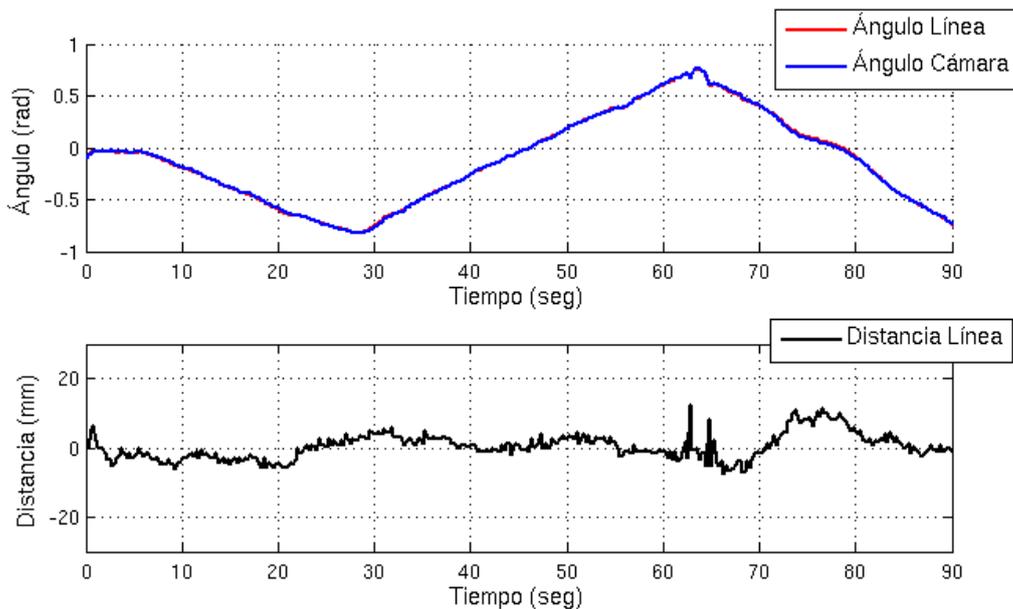


Figura 186. Seguimiento de línea con  $\tilde{\rho} = 0$ .

En la Figura 187, se realiza un segundo experimento, en que se fija el valor del ángulo de la línea para que en todo momento sea  $\Phi \approx 0$ . En este caso, se pretende mostrar que, aunque el ángulo sea siempre nulo, la cámara es capaz de apuntar hacia la línea cuando ésta se desplaza horizontalmente. Se observa que desplazamientos positivos en  $\rho$  (la línea está a la derecha de la cámara) generan ángulos negativos (la cámara gira en el sentido de las agujas del reloj y viceversa).

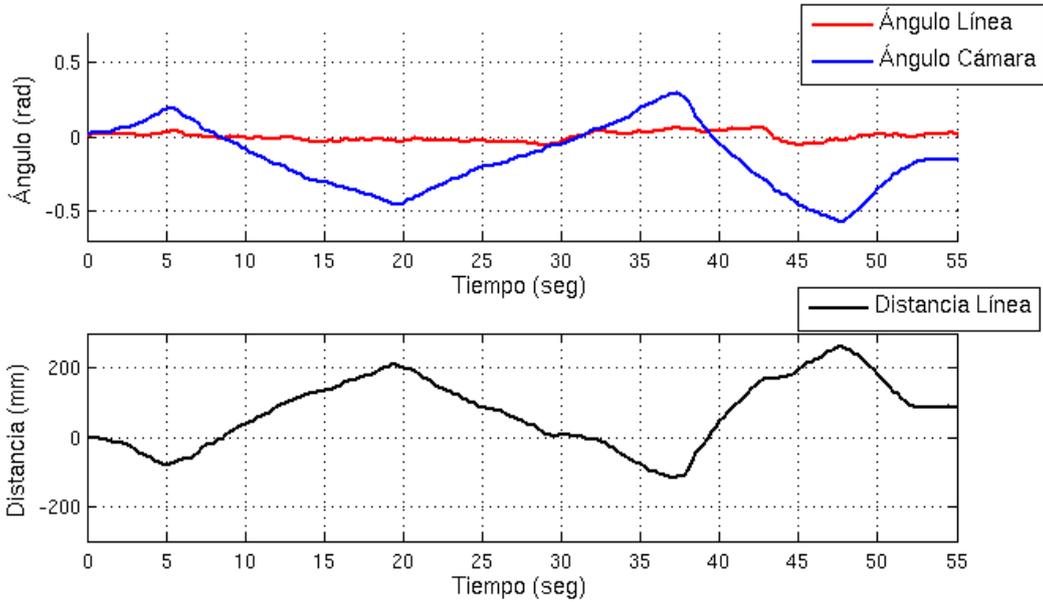


Figura 187. Seguimiento de línea con  $\tilde{\phi} = 0$ .

Finalmente, se ha realizado un tercer experimento en el que se ha generado un movimiento arbitrario entre la línea y la cámara (Figura 188). Se observa que en términos generales la cámara adopta el ángulo que detecta de la línea, en cuyo caso las discrepancias vienen motivadas por la separación de la línea con respecto al origen de la cámara, que no es nula para este experimento.

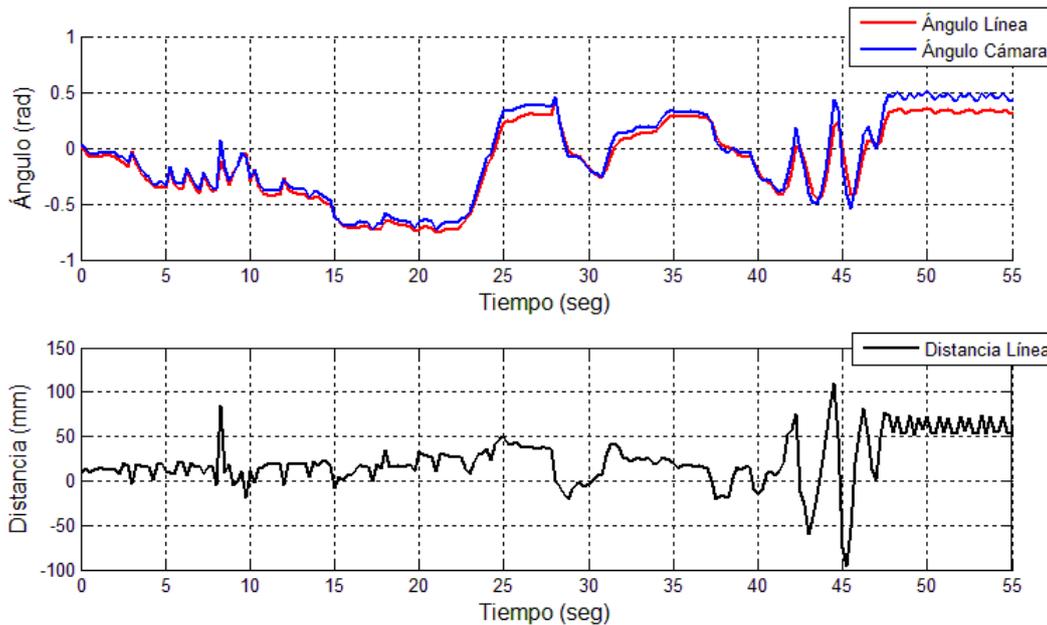


Figura 188. Seguimiento de línea para valores arbitrarios de  $\tilde{\rho}$  y  $\tilde{\phi}$ .

Por el hecho de que mostrar los resultados experimentales en gráficas puede resultar poco ilustrativo, se han realizado varias grabaciones en las que se comprueba el funcionamiento del algoritmo con diferentes modos de funcionamiento. En total hay 3 videos, que se incluyen en el CD que acompaña la presente memoria de Tesina.

- **Video 1:** Detección y *tracking* de línea con transmisión de la imagen con baja resolución (frecuencia de trabajo aproximada de 1Hz).
- **Video 2:** Detección y *tracking* de línea sin transmisión de la imagen en modo manual, con solicitud de captura automática (frecuencia de trabajo aproximada de 5-6Hz).
- **Video 3:** Detección y *tracking* de línea sin transmisión de la imagen en modo automático (frecuencia de trabajo aproximada de 8Hz).

### **5.2.1.2. Seguimiento de la Línea**

Como se ha dicho anteriormente para resolver el problema del seguimiento de línea por parte del AGV se ha implementado una ley de control de tipo persecución pura (ver ecuación (174)). No se han realizado validaciones cuantitativas del correcto funcionamiento del sistema desde un punto de vista de control, como por ejemplo cálculo de la sobreoscilación o el tiempo de convergencia en el seguimiento de la referencia, ni tampoco se han tomado datos de los errores de posición cometidos por el vehículo.

No obstante, sí se han realizado implementaciones reales sobre la carretilla elevadora presentada al principio de este capítulo y de forma cualitativa sí que se ha validado el correcto funcionamiento del algoritmo. Como ejemplo de ello se hace referencia a los vídeos grabados sobre el seguimiento del AGV y que se incluyen en el CD. Dichos vídeos son:

- **Video 4:** Seguimiento de línea sin sistema de visión activa.
- **Video 5:** Seguimiento de línea con sistema de visión activa.
- **Video 6:** Seguimiento de línea con sistema de visión activa para un recorrido largo.

## **5.3. Experimentación con Player/Stage**

Player es un servidor basado en red para el control de robots (entorno multirobot). Al ejecutar Player en tu robot, se dispone de un conjunto de interfaces sencillas y bastante claras para poder adquirir datos de los sensores, así como enviar datos a los actuadores, pero también para incorporar diversos algoritmos que actúan como dispositivos. Todo esto funciona mediante el paso de mensajes a través de TCP/IP con lo que es muy sencillo configurar un entorno en el que se puede leer y escribir de forma remota, por ejemplo para tele-operar un robot o para compartir datos de los sensores, mapas, posiciones, etc. entre robots. Los dispositivos se pueden conectar una vez Player se está ejecutando de forma que sobre la marcha podemos indicarle a Player que disponemos de un nuevo sensor proporcionando una serie de datos que estarán disponibles a partir de ese momento para quien lo requiera.

Player soporta una gran variedad de hardware, tales como los bien conocidos robots de ActivMedia como el Pioneer, pero también una gran variedad de sensores y actuadores: láseres, sónares, cámaras motorizadas (PTU), *bumpers*, etc. Se trata de una arquitectura software basada

en componentes, con lo que es completamente modular y permite a la Comunidad Robótica desarrollar nuevos drivers a medida que se necesiten. En la actualidad, Player funciona fundamentalmente bajo Linux.

Stage es un programa que permite simular entornos multi-robot, diversos tipos de sensores y objetos en un entorno 2D½ basado en un mapa de bits. Esta plataforma ha sido diseñada para soportar la investigación en sistemas autónomos multi-agentes, para simular diversos dispositivos de forma sencilla y computacionalmente muy poco costosa.

Principalmente hay dos maneras de utilizar la plataforma a nivel básico, la creación de mundos y archivos de configuración sobre los que actuaremos, a través de *proxies* con un determinado programa externo; o la creación de mundos y archivos de configuración que cargarán directamente alguna rutina (funciones de control) desde el archivo mundo. El modus operandi será trabajar con un archivo mundo sobre los que actuaremos desde un programa exterior. La siguiente figura resume la estructura de control de la aplicación tanto para el caso que accedamos directamente a hardware como si se ejecuta una simulación. En ambos casos, el código que desarrollemos será idéntico, ya que los drivers proporcionarán los datos a través del proxy.

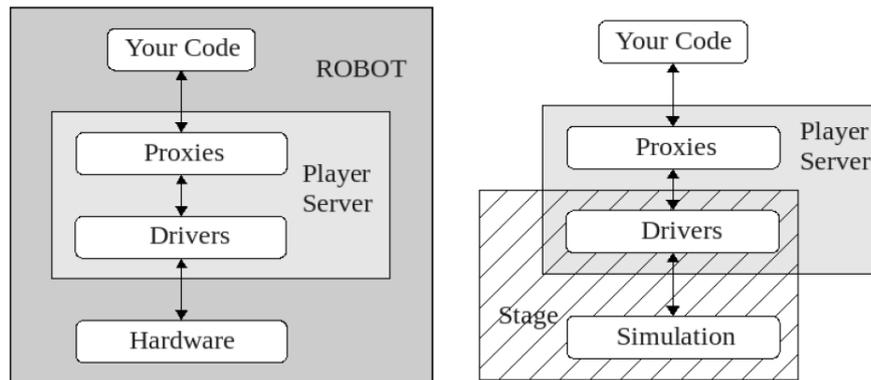


Figura 189. Esquemas de funcionamiento de la plataforma Player para un robot real y para Stage.

Por lo tanto, como conclusión se puede obtener que los algoritmos desarrollados en el simulador para la plataforma Stage se pueden extrapolar de forma directa, sin modificar nada del código, para usarse sobre la plataforma robótica real. Es por ese motivo por el que se decide utilizar el simulador Player/Stage.

La experimentación realizada en Player para la presente Tesina ha consistido en la realización de las simulaciones de visión para la identificación de la línea por parte del robot, la generación de trayectorias de curvatura continua y el control por persecución pura. Además se han realizado los primeros pasos para implementar el control mediante *cc-paths*, tanto en simulación como posteriormente aplicados sobre el robot *Pioneer*.

En la Figura 190 se muestran algunos de los resultados obtenidos en el problema de seguimiento de trayectoria. Como se observa se han generado 2 rutas distintas y el robot es capaz de seguirlas, tal y como se observa en la Figura 190, puesto que en azul aparece la trayectoria planificada y en rojo la traza seguida por el robot.

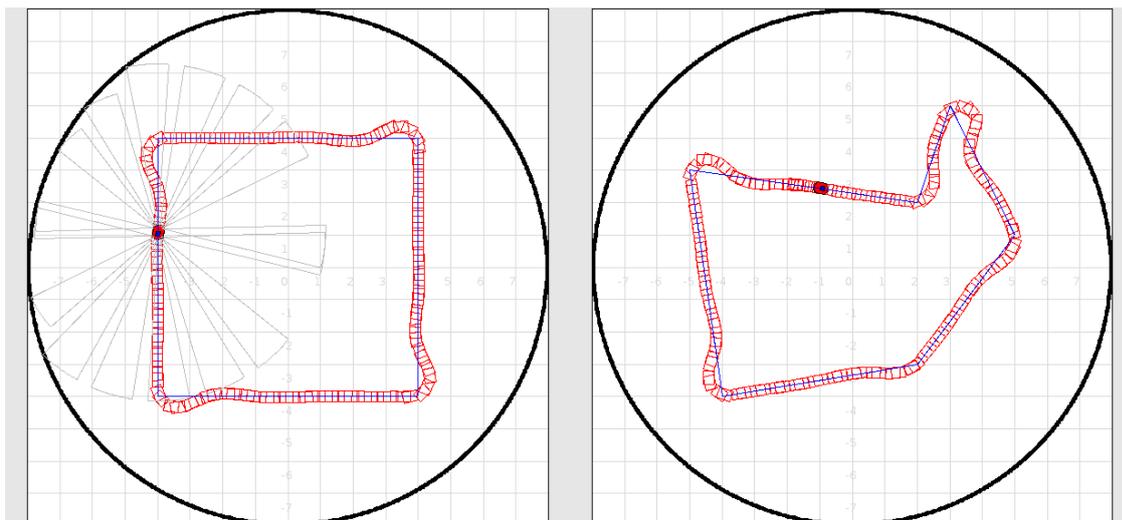


Figura 190. Ejemplos de seguimiento de trayectoria realizados con la plataforma Player/Stage.

Hay que destacar que el control del robot se ha realizado mediante la técnica clásica de persecución pura, aunque el objetivo a corto plazo es implementar las técnicas de control mediante *cc-paths*.

#### 5.4. Experimentación con Player y Robot Pioneer 3DX

De momento no se han realizado implementaciones sobre esta plataforma, pero como ya se ha comentado puede hacerse una extrapolación directa de los algoritmos de simulación sobre el robot *Pioneer*.

Pese a que no se han realizado aun experimentos sobre el robot sí que se ha trabajado con él para adquirir una secuencia de imágenes a modo de ejemplo para resolver la etapa de procesamiento por visión en el problema del seguimiento de líneas. Es por ello que a continuación se muestran algunas imágenes en las que se observa la puesta a punto del robot junto a algunos detalles de los sensores que incorpora.



Figura 191. Robot Pioneer con diversos sensores: cámara, láser, sónar y encoders.

En las siguientes figuras se muestran los sensores ampliados para que puedan verse con mayor nivel de detalle:



Figura 192. Cámara IP de la marca *Edimax* instalada en la parte superior del robot *Pioneer*.



Figura 193. Láser de telemetría de la marca *Hokuyo* instalado en la parte frontal del robot *Pioneer*.



Figura 194. Anillo de sensores de tipo sónar que incorpora el robot *Pioneer*.



Figura 195. Motor y encoder de una de las ruedas del robot *Pioneer*.



# **CAPÍTULO 6**

## **Conclusiones y Trabajos Futuros**



## 6. Conclusiones y Trabajos Futuros

### 6.1. Conclusiones

Una conducción segura y confortable implica tener en cuenta aspectos tales como aceleraciones tangenciales y centrípetas con sus correspondientes *jerks*. Esto es importante, no tan solo para conseguir una conducción suave para pasajeros y mercancías sino también para asegurar la adherencia al firme ante situaciones imprevista y giros acusados. Este tipo de soluciones pueden aplicarse en la industria en aquellas situaciones en las que los materiales deban transportarse grandes distancias, sin necesidad de intervención humana, como por ejemplo en almacenes logísticos. También puede aplicarse en entornos hostiles cuando sea necesario prescindir de la presencia humana, ya sea por el tipo de material transportado o por la peligrosidad de la zona, como por ejemplo debido a la presencia de tuberías de gas, materiales explosivos, etc. Estos aspectos ya considerados en vehículos automóviles así como en el trazado de carreteras no han sido incorporados al guiado automático de vehículos industriales (AGV).

La presente Tesina de Máster ha supuesto el estudio y desarrollo de métodos para la mejora de la seguridad y la confortabilidad en vehículos autónomos en el transporte de pasajeros y/o materiales. Concretamente se ha realizado el desarrollo de un nuevo método para la generación de trayectorias con curvatura continua considerando las restricciones cinemáticas o mecánicas, esto es el máximo ángulo de giro admisible, como restricciones dinámicas, limitando las aceleraciones y *jerks* máximos, con objeto de mejorar la adherencia de las ruedas al firme.

Los desarrollos de esta Tesina han sido aplicados para resolver el problema de seguimiento de líneas, o más genéricamente de caminos, basados en sistemas de visión artificial. Para ello, se han combinado técnicas tradicionales de visión por con las técnicas de generación de trayectorias con curvatura continua. Las técnicas desarrolladas se pueden aplicar a un amplio abanico de situaciones en las que destacan la generación de maniobras especiales, tales como adelantamientos, aparcamientos, transporte de materiales, etc.

Para la realización de la Tesina, se han estudiado numerosas tesis y trabajos extensos de investigación relacionados con la generación de trayectorias de curvatura continua, el control cinemático, algoritmos de visión artificial y aplicaciones industriales. A partir de dichos estudios, se ha podido obtener un amplio conocimiento del estado actual de la investigación en el ámbito de los objetivos y problemas planteados. Si bien comentar, que debido a la limitación en cuanto a extensión de la memoria de la presente Tesina no se comentan todos los desarrollos realizados en algunas de las tesis consultadas [63], [64], [65], [66], [67], [68], [69], [70], [71], [108], [109], [110], [111], [112].

Tras el estudio realizado, se llega a la conclusión de que pese a que se ha trabajado con Clotoides para modelar las carreteras y/o generar las trayectorias mediante un planificador de movimiento, son pocos los trabajos dentro del ámbito de la robótica móvil y por tanto es una línea de investigación prácticamente virgen y con un largo recorrido de cara a futuros desarrollos. En este sentido, no se han encontrado investigaciones en relación a control de robots para seguimiento de líneas utilizando Clotoides en el propio bucle de control cinemático. Por tanto, en vista de la ausencia de técnicas de control que garanticen la continuidad de la curvatura en el

problema del seguimiento de líneas y dado que la utilización de Clotoides para la generación de trayectorias en robótica móvil nos ofrece diversas ventajas con respecto a cualquier otra curva, ya que al ser una trayectoria suave en todo su recorrido se minimiza el deslizamiento de las ruedas y con ello el error de odometría, se decide adoptar esta como la principal línea de trabajo de la presente Tesina y una de las más importantes cara a la futura Tesis Doctoral como continuación lógica del trabajo ya realizado.

Para las implementaciones se ha utilizado un sistema de visión, por ser de bajo coste y muy versátiles, si bien la investigación no requiere necesariamente de la utilización de visión artificial, pudiéndose otro tipo de tecnologías sensoriales.

Como resultado más inmediato de los desarrollos realizados en esta Tesina, se ha conseguido publicar en diversos congresos internacionales además de tener en la actualidad en estado de revisión dos artículos a conferencias de especial relevancia como son el ICRA, dentro del ámbito de la robótica, y el IFAC, dentro del ámbito del control, así como un artículo de carácter de transferencia tecnológica a la revista RIAI en su primer número especial.

## **6.2. Trabajos Futuros**

Quedan pendientes de trabajo futuro de investigación los siguientes aspectos:

- Generalizar las trayectorias con curvatura continua del espacio 2D al espacio 3D. Aspecto totalmente inédito en robótica. Con ello se podrán generar trayectorias para vehículos submarinos y aéreos con restricciones cinemáticas y dinámicas, para aplicación inmediata para un proyecto de Plan Nacional de robótica submarina con el que estoy colaborando.
- Extender el uso de Clotoides de la robótica móvil al ámbito de la robótica de manipuladores industriales, lo cual será de gran utilidad en mecanizados de alta velocidad en células robotizadas.
- Caracterizar de forma natural los parámetros de las curvas basadas en Clotoides que afectan a la máxima brusquedad admisible en la trayectoria.
- Estudiar la generación de trayectorias libres de colisión bajo las premisas establecidas en la Tesina, es decir, teniendo en cuenta tanto las restricciones cinemáticas como dinámicas.
- Análisis de tiempos de computación de cara a la implementación de estos algoritmos en tiempo real.

A nivel de las implementaciones realizadas, quedan pendientes algunas experiencias que por su dedicación requerida han quedado finalmente como trabajos futuros más inmediatos. Así pues se pretende demostrar experimentalmente los beneficios obtenidos con la aplicación de los métodos propuestos. Quedan pendientes muchas posibles implementaciones prácticas cara a caracterizar aspectos tales como la adherencia de las ruedas, la confortabilidad y suavidad de las trayectorias generadas, etc.

En este sentido, se quiere validar experimentalmente las mejoras en la confortabilidad al transportar materiales mediante la medición de las aceleraciones a las que está sometido el vehículo, en las que podría utilizarse a modo de validación de forma visual el transporte de

material delicado, como por ejemplo copas. Otro experimento podría consistir en mover repetidamente al robot a lo largo de una trayectoria predefinida, por ejemplo un cuadrado, en la que la estimación de la posición mediante la odometría diverge de forma distinta frente a la posición real en función del control cinemático aplicado.

También, desde un punto de vista de implementación, se mejorarán algunos aspectos de menor relevancia en el procesamiento del sistema de visión, cara a hacer los algoritmos más robustos y eficientes, así como el uso de técnicas de SLAM para la estimación de la posición y el mapa del entorno. También se estudiará el uso de los presentes métodos como extensión natural de los métodos más usados para la planificación de trayectorias y evitación de obstáculos de forma que se garantice no sólo la ausencia de colisiones, sino que ésta se realiza mediante la generación de trayectorias suaves.

Finalmente, se quiere hacer destacar el uso de novedosas tecnologías, como la *Kinect* de Microsoft. Este sensor, incluye una cámara de color estándar además de una cámara infrarroja que combinada con un emisor de luz estructurada y un avanzado software de procesamiento permite obtener una imagen de profundidad, pudiendo por tanto obtener una nube de puntos 3D de los objetos del entorno a un coste muy inferior a las cámaras 3D basadas en TOF o láseres de telemetría montados en plataformas giratorias. En este sentido, la Comunidad Robótica está experimentando una auténtica revolución tanto por el amplio abanico de aplicaciones en las que puede ser utilizada como la calidad y el tipo de información proporcionada. La utilización de esta tecnología para la generación de trayectorias de curvatura continua en el espacio 3D será sin duda una línea de investigación con unas expectativas muy interesantes, tanto por el interés científico que representa como por su abanico de aplicaciones.



# Bibliografía

- [1] J. C. Latombe. **Robot Motion Planning**. Kluwer Academic Publishers. 1991.
- [2] B. Siciliano and O. Khatib Eds. **Handbook of Robotics**. Springer. 2008
- [3] L. E. Dubins. **On curves of minimal length with constraint on average curvature, and with prescribed initial and terminal positions and tangents**. *American J. Math.* 1957
- [4] J. P. Laumond. **Feasible trajectories for mobile robots with kinematic and environment constraints**. *Int. Conf. Intelligent Autonomous Systems*. 1986
- [5] P. Jacobs and J. Canny. **Planning Smooth Paths for Mobile Robots**. *IEEE Int. Conf. on Robotics and Automation*. 1989
- [6] J. A. Reeds and L. A. Shepp. **Optimal paths for a car that goes both forward and backward**. *Pacific J. Math.* 1990
- [7] J. Iijima, Y. Kanayama and S. Yuta. **A locomotion control system for mobile robots**. *Int. Joint Conf. Artificial Intelligence*. 1981
- [8] Y. Kanayama and N. Miyake. **Trajectory generation of mobile robots**. *Int. Symp. On Robotics Research (ISR)*. 1985
- [9] Y. Kanayama and B. I. Hartman. **Smooth Local Path Planning for Autonomous Vehicles**. *IEEE Int. Conf. on Robotics and Automation*. 1989
- [10] D. Shin and S. Singh. **Path generation for robots vehicles using composite Clothoid segments**. *CMU Technical Report*. 1990
- [11] S. Fleury, P. Souères, J. P. Laumond and R. Chatila. **Primitives for smoothing mobile robot trajectories**. *IEEE Transactions on Robotics and Automation*. 1995
- [12] M. Tounsi and J. F. Le Corre. **Trajectory generation for mobile robots**. *Mathematics and Computers in Simulation*. 1996
- [13] A. Scheuer and T. Fraichard. **Planning Continuous-Curvature Paths for Car-Like Robots**. *IEEE Int. Conf. on Intelligent Robots and Systems*. 1996
- [14] A. Scheuer and T. Fraichard. **Collision-Free and Continuous-Curvature Path Planning for Car-Like Robots**. *IEEE Int. Conf. on Robotics and Automation*. 1997
- [15] A. Scheuer and T. Fraichard. **Continuous-Curvature Path Planning for Car-Like Vehicles**. *IEEE Int. Conf. on Intelligent Robots and Systems*. 1997
- [16] T. Fraichard and J. M. Ahuactzin. **Smooth Path Planning for Cars**. *IEEE Int. Conf. on Robotics and Automation*. 2001
- [17] B. Nagy and A. Kelly. **Trajectory Generation for Car-Like Robots using Cubic Curvature Polynomials**. *Int. Conf. in Field and Service Robotics*. 2001
- [18] D. J. Walton and D. S. Meek. **A Controlled Clothoid Spline**. *Computer and Graphics*. 2005
- [19] A. Piazzi, C. G. Lo Bianco and M. Romano.  **$\eta^3$ -Splines for the Smooth Path Generation of Wheeled Mobile Robots**. *IEEE Transactions on Robotics*. 2007
- [20] W. Yossawee, T. Tsubouchi, S. Sarata and S. Yuta. **Path Generation for Articulated Steering Type Vehicle Using Symmetrical Clothoid**. *IEEE Int. Conf. on Industrial Technology*. 2002
- [21] S. Sarata, W. Yossawee, A. Horiguchi and T. Tsubouchi. **V Shape Path Generation for Loading Operation by Wheel Loader**. *Field and Service Robotics*. 2006
- [22] K. Yang and S. Sukkarieh. **Real-Time Continuous Curvature Path Planning of UAVs in Cluttered Environments**. *Int. Symp. on Mechatronics and its Applications*. 2008

- [23] L. Techy, C. A. Woolsey and K. A. Morgansen. **Planar Path Planning for Flight Vehicles in Wind with Turn Rate and Acceleration Bounds.** *IEEE Int. Conf. on Robotics and Automation.* 2010
- [24] Steven H. Schot. **Jerk. The time rate change of acceleration.** *American Journal of Physics.* 1978
- [25] J. D. Boissonnat, A. Cerezo and J. Leblond. **A note on shortest path in the plane subject to a constraint on the derivative of the curvature.** *INRIA Research Report.* 1994
- [26] A. Marchionna and P. Perco. **A Proposal to Update the Clothoid Parameter Limiting Criteria of the Italian Standard.** *4th Int. SIV Congress.* 2007
- [27] L. Labakhua, U. Nunes, R. Rodrigues and F. S. Leite. **Smooth Trajectory Planning for Fully Automated Passengers Vehicles. Spline and Clothoid Based Methods and Its Simulation.** *Springer. Informatics in Control Automation and Robotics.* 2008
- [28] A. Scheuer and C. Laugier. **Planning Sub-Optimal and Continuous-Curvature Paths for Car-Like Robots.** *IEEE. Int. Conf. on Intelligent Robots and Systems.* 1998
- [29] A. Scheuer and M. Xie. **Continuous-Curvature Trajectory Planning for Manoeuvrable Non-Holonomic Robots.** *IEEE. Int. Conf. on Intelligent Robots and Systems.* 1999
- [30] T. Fraichard, A. Scheuer, and R. Desvigne. **From Reeds and Sheep's to continuous curvature paths.** *IEEE Int. Conf. on Advanced Robotics.* 1999
- [31] K. Jiang, D. Z. Zhang and L. D. Seneviratne. **A parallel Parking System for a Car-like robot with sensor guidance.** *Institution of Mechanical Engineering.* 1998.
- [32] I. E. Paromtchik and C. Laugier. **Motion Generation and Control for Parking an Autonomous Vehicle.** *IEEE Int. Conf. on Robotics and Automation.* 1994
- [33] L. Gracia and J. Tornero. **Geometric Parallel Parking Planner for Car-Like Vehicles.** *Int. Simulation Conference.* 2003
- [34] J. P. Laumond, P. E. Jacobs, M. Taix and R. M. Murray. **A Motion Planner for Nonholonomic Mobile Robots.** *IEEE Transactions on Robotics and Automation.* 1994
- [35] I. Papadimitriou and M. Tomizuka. **Fast Lane Changing Computations using Polynomials.** *American Control Conference.* 2003
- [36] N. Montés and J. Tornero. **Lane changing using s-series Clothoidal approximation and dual-rate based on Bezier points for controlling vehicle.** *Int. Conf. on Systems Theory and Scientific Computation.* 2004
- [37] N. Montés, M. C. Mora and J. Tornero. **Trajectory Generation based on Rational Bezier Curves as Clothoids.** *Intelligent Vehicles Symposium.* 2007
- [38] E. Szádeczky and B. Kiss. **Continuous-Curvature Paths for Mobile Robots.** *Technical Report.* 2008
- [39] Doran K. Wilde. **Computing Clothoid Segments for Trajectory Generation.** *IEEE Int. Conf. on Intelligent Robots and Systems.* 2009
- [40] Ignacio de Corral Manuel de Villena. **Topografía de obras.** *SPUPC* 2001
- [41] **Road Geometry: Highway Link Design** (1993). Departmental Standard TD 9/93, Department of Transport, United Kingdom.
- [42] **Rural Road Design: Guide to the Geometric Design of Rural Roads** (1997). Austroads, Sydney, Australia.
- [43] **Perception of Road-Curve Characteristics.** TNO Report C-18, TNO Institute for Perception, Soesterberg, The Netherlands.
- [44] **Norme funzionali e geometriche per la costruzione delle strade** (2001). D.M. 05/11/2001, Ministry of Infrastructures and Transportation, Rome, Italy.
- [45] Darla L. Weiss. **Dynamic Simulation and Analysis of Roller Coasters.** *Thesis.* 1998
- [46] J. McCrae and K. Singh. **Sketching Piecewise Clothoid Curves.** *Computers & Graphics.* 2009

- [47] I. Bara, J. Lehtinen and J. Popović. **Sketching Clothoid Splines Using Shortest Paths**. *Computer Graphics Forum*. 2010
- [48] J. Boersma. **On a numerical method for the computation of Fresnel integrals**. Technical Report. 1960
- [49] J. Boersma. **Computation of Fresnel Integrals**. *Mathematics of Computation*. 1960
- [50] O. L. Fleckner. **A Method for the Computation of the Fresnel Integrals and Related Functions**. *Mathematics of Computation*. 1968
- [51] W. J. Cody. **Chebyshev Approximations for the Fresnel Integrals**. *Mathematics of Computation*. 1968
- [52] M.A. Heald. **Rational approximation for the Fresnel Integrals**. *Mathematics of Computation*. 1985
- [53] Klaus D. Mielenz. **Computation of Fresnel Integrals**. *Journal of Research of the National Institute of Standards and Technology*. 1997
- [54] Klaus D. Mielenz. **Computation of Fresnel Integrals II**. *Journal of Research of the National Institute of Standards and Technology*. 2000
- [55] L. Z. Wang, K. T. Miura, E. Nakamae, T. Yamamoto and T. J. Wang. **An approximation approach of the Clothoid curve defined in the interval  $[0, \pi/2]$  and its offset by free-form curves**. *Computer-Aided Design*. 2001
- [56] D. Khosla. **Accurate Estimation of Forward Path Geometry using Two-Clothoid Road Model**. *IEEE Intelligent Vehicle Symposium*. 2002
- [57] D. A. Schwartz. **Clothoid Road Geometry Unsuitable for Sensor Fusion. Clothoid Parameter Sloshing**. *IEEE Intelligent Vehicles Symposium*. 2003
- [58] J. S. Reyes, J. M. Chacón. **Polynomial Approximation to Clothoids via S-Power Series**. *Computer-Aided Design*. 2003
- [59] D. S. Meek and D. J. Walton. **An Arc Spline Approximation to a Clothoid**. *Journal of Computational and Applied Mathematics*. 2004
- [60] N. Montés, A. Herraiz, L. Armesto and J. Tornero. **Real-time Clothoid Approximation by Rational Bezier Curves**. *IEEE Int. Conf. on Robotics and Automation*. 2008
- [61] D. Tilbury, R. Murray and S. Sastry. **Trajectory Generation for the N-trailer Problem Using Goursat Normal Form**. *IEEE Conference on Decision and Control*. 1993
- [62] V. P. Kostov and E. D. Kostova. **The Planar Motion with Bounded Derivative of the Curvature and its Suboptimal Paths**. *Acta Math. Univ. Comenianae*. 1995
- [63] V. P. Kostov and E. D. Kostova. **Some properties of Clotoids**. *INRIA Research Report*. 1995
- [64] E. D. Kostova and V. P. Kostov. **Irregularity of optimal trajectories in a control problem for a car-like robot**. *INRIA Research Report*. 1998
- [65] A. Sheuer. **Planification Courbure Continue pour Robot Mobile**. *Thesis*. 1998
- [66] Matthew J. Barton. **Controller Development and Implementation for Path Planning and Following in an Autonomous Urban Vehicle**. *Thesis*. 2001
- [67] Kenneth R. Meidenbauer. **An Investigation of the Clothoid Steering Model for Autonomous Vehicles**. *Thesis*. 2007
- [68] Joshua Henrie. **Planning Continuous Curvature Paths using Constructive Polylines**. *Thesis*. 2008
- [69] James Palmer McCrae. **Sketch-Based Path Design**. *Thesis*. 2008
- [70] T. Hessburg and M. Tomizuka. **Fuzzy Logic Control for Lateral Vehicle Guidance**. *IEEE Control Systems Magazine*. 1994
- [71] D. Pomerleau. **RALPH Rapidly Adapting Lateral Position Handler**. *Intelligent Vehicle Symposium*. 1996
- [72] H. Cramer, U. Scheunert, G. Wanielik. **A New Approach for Tracking Lanes by Fusing Image Measurements**

- with map data.** *IEEE Intelligent Vehicles Symposium.* 2004
- [73] S. Shimizu, H. Fujiyoshi, H. Sakai and T. Kanade. **Recognition of Road Contours Based on Extraction of 3D Positions of Delineators.** *IEEE Int. Conf. on Intelligent Transportation Systems.* 2007
- [74] V. Neagoie and C. Tudoran. **Road Following for Autonomous Vehicle Navigation using a Concurrent Neural Classifier.** *Int. Symp. Soft Computing in Industry.* 2008
- [75] C. Wang, Z. Hu and K. Uchimura. **A Precise Road Network Modeling and Map Matching for Vehicle Navigation.** *IEEE Int. Conf. on Intelligent Transportation Systems.* 2008
- [76] Y. Ma, J. Košecká, and S. S. Sastry. **Vision Guided Navigation for a Nonholonomic Mobile Robot.** *IEEE Transactions on Robotics and Automation.* 1999
- [77] K. A. Redmill S. Upadhyaya A. Krishnamurthy Ü. Özgüner. **A Lane Tracking System for Intelligent Vehicle Applications.** *IEEE Int. Conf. on Intelligent Transportation Systems.* 2001
- [78] L. Armesto, L. Gracia and J. Tornero. **Multi-Rate Line Tracking for Mobile Robots Based on Artificial Vision and Odometry.** *Industrial Simulation Conference.* 2002
- [79] C. Corridori and M. Zanin. **High Curvature Two-Clothoid Road Model Estimation.** *Intelligent Transportation Systems Conference.* 2004
- [80] S. Chaib, M. S. Netto and S. Mammar.  **$H_\infty$ , Adaptive, PID and Fuzzy control: a comparison of controllers for vehicle lane keeping.** *IEEE Intelligent Vehicles Symposium.* 2004
- [81] L. Beji and Y. Bestaoui. **Motion Generation and Adaptive Control Method of Automated Guided Vehicles in Road Following.** *IEEE Transactions on Intelligent Transportation Systems.* 2005
- [82] T. Bakker, K. van Asselt, J. Bontsema, J. Müller and G. van Straten. **A Path Following Algorithm for Mobile Robots.** *Springer Auton Robot.* 2010
- [83] A. Broggi, M. Bertozzi, A. Fascioli, C. G. Lo Bianco and A. Piazzini. **The ARGO Autonomous Vision and Control Systems.** *Int. Journal of Intelligent Control and Systems.* 1999
- [84] M. A. Sotelo, J. Nuevo, L. M. Bergasa, M. Ocaña, I. Parra and D. Fernández. **Road Vehicle Recognition in Monocular Images.** *IEEE Int. Symposium on Industrial Electronics.* 2005
- [85] S. Nedeveschi, R. Schmidt, T. Graf, R. Danescu, D. Frentiu, T. Marita, F. Oniga and C. Pocol. **3D Lane Detection System Based on Stereovision.** *IEEE Int. Conf. on Intelligent Transportation Systems.* 2004
- [86] R. Danescu, S. Nedeveschi and T. B. To. **A Stereovision-Based Lane Detector for Marked and Non-Marked Urban Roads.** *Int. Conf. on Intelligent Computer Communications and Processing.* 2007
- [87] R. Danescu, S. Nedeveschi, M.M. Meinecke, T.B. To. **Lane Geometry Estimation in Urban Environments Using a Stereovision System.** *Intelligent Transportation Systems Conference.* 2007
- [88] C. Gackstatter, S. Thomas, P. Heinemann, G. Klinker. **Stable Road Lane Model Based on Clothoids.** *Int. Forum on Advanced Microsystems and Automotive Applications.* 2010
- [89] R. A. Gupta, W. Snyder and W. S. Pitts. **Concurrent visual multiple lane detection for autonomous vehicles.** *IEEE Int. Conf. on Robotics and Automation.* 2010
- [90] A. S. Huang and S. Teller. **Probabilistic Lane Estimation using Basis Curves.** *Robotics: Science and Systems Conference.* 2010
- [91] C. Corridori, D. Giordani, P. Lombardi, S. Messelodi, C. M. Modena, M. Zanin. **An In-Vehicle Vision System for Dangerous Situation Detection.** 2005
- [92] L. B. Cremean and R. M. Murray. **Model-Based Estimation of Off-Highway Road Geometry.** *IEEE Int. Conf. on Robotics and Automation.* 2006
- [93] A. S. Huang, D. Moore, M. Antone, E. Olsony and S. Teller. **Multi-Sensor Lane Finding in Urban Road Networks.** *Robotics: Science and Systems Conference.* 2008
- [94] M. Manz, F. von Hundelshausen and H. J. Wuensche. **A Hybrid Estimation Approach for Autonomous Dirt**

- Road Following using Multiple Clothoid Segments.** *IEEE Int. Conf. on Robotics and Automation.* 2010
- [95] C. J. Taylor, J. Koščeká, R. Blasi and J. Malik. **A Comparative Study of Vision-Based Lateral Control Strategies for Autonomous Highway Driving.** *Int. Journal of Robotics Research.* 1999
- [96] M. Bertozzi, A. Broggi, M. Celliario, A. Fascioli, P. Lombardi and M. Porta. **Artificial Vision in Road Vehicles.** *Proceedings of the IEEE.* 2002
- [97] V. Kastinaki, M. Zervakis, K. Kalaitzakis. **A Survey of video processing for traffic applications.** *Elsevier. Image and Vision Computing.* 2003
- [98] J. Zhou, W. F. Bischof, T. Caelli. **Road tracking in aerial images based on human-computer interaction and Bayesian filtering.** *Elsevier. Photogrammetry and Remote Sensing.* 2006
- [99] J. C. McCall and M. M. Trivedi. **Vision Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation.** *IEEE Transactions on Intelligent Transportation Systems.* 2006
- [100] G. Campion, G. Bastin and B. A. Novel. **Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots.** *IEEE Transactions on Robotics and Automation.* 1996
- [101] B. A. Novel, G. Campion and G. Bastin. **Control of Nonholonomic Wheeled Mobile Robots by State Feedback Linearization.** *Int. Journal of Robotics Research.* 1995
- [102] A. Ollero. **Robótica. Manipuladores y Robots Móviles.** *Marcombo Boixareu.* 2001
- [103] V. Girbés, L. Armesto and J. Tornero. **On Generating Continuous-Curvature Paths for Line Following Problem with Curvature and Sharpness Constraints.** *IEEE. Int. Conf. on Robotics and Automation.* 2011
- [104] V. Girbés, L. Armesto and J. Tornero. **Continuous-Curvature Control of Mobile Robots with Constrained Kinematics.** *IFAC World Congress.* 2011
- [105] M. Fischler and R. Bolles. **Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography.** *Graphics and Image Processing.* 1981.
- [106] O. Chum and J. Matas. **Randomized RANSAC with  $T_{d,d}$  Test.** *Image and Vision Computing.* 2004.
- [107] J. Matas and O. Chum. **Randomized RANSAC with Sequential Probability Ratio Test.** *Int. Conf. of Computer Vision.* 2005
- [108] Web de CMUcam: [www.cmucam.org](http://www.cmucam.org)
- [109] G. Araujo and M. Aldon. **Line Extraction in 2D Range Images for Mobile Robotics.** *Journal of Intelligent and Robotic Systems.* 2004.
- [110] V. Girbés, E. Solanes, E. Martínez, L. Armesto and J. Tornero. **Proyecto PISALA: Sensorización Inteligente para el Seguimiento de Líneas por Visión Artificial.** *Robot'09 Workshop.* 2009
- [111] V. Girbés, L. Armesto and J. Tornero. **PISALA project. Intelligent Sensorization for Line Tracking with Artificial Vision.** *Int. Symposium on Robotics.* 2010
- [112] M. B. Duinkerken, J. A. Ottjes and G. Lodewijks. **Comparison of routing strategies for AGV systems using simulation.** *Proc. of Winter Simulation.* 2006
- [113] S. Thrun, W. Burgard and D. Fox. **Probabilistic Robotics.** *MIT Press.* 2005
- [114] K. R. S. Kodagoda, W. S. Wijesoma and E. K. Teoh. **Fuzzy speed and steering control of an AGV.** *IEEE Transactions on Control Systems Technology.* 2002
- [115] W. S. Wijesoma, K. R. S. Kodagoda and E. K. Teoh. **Stable Fuzzy State Space Controller for Lateral Control of an AGV.** *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology.* 2002
- [116] M. Brady, H. Durrant-Whyte, H. Hu, J. Leonard, P. Probert and B. S. Y. Rao. **Sensor-based control of AGVs.** *Computing & Engineering Journal.* 1990
- [117] G. Garibotto, S. Masciangelo, P. Bassino, C. Coelho, A. Pavan and M. Marson. **Industrial exploitation of computer vision in logistic automation: autonomous control of an intelligent forklift truck.** *IEEE. Int. Conf. on Robotics and Automation.* 1998

- [118] J. Pagés, X. Armangué, J. Salvi, J. Freixenet and J. Martí. **A computer vision system for autonomous forklift vehicles in industrial environments.** *Mediterranean Conf. on Control and Automation*. 2001
- [119] M. S. J. Yoder. **Automatic pallet engagement by a vision guided forklift.** *IEEE Int. Conf. on Robotics and Automation*. 2005
- [120] M. Mora, V. Suesta, L. Armesto and J. Tornero. **Factory management and transport automation.** *IEEE Conf. on Emerging Technologies and Factory Automation*. 2003
- [121] L. Armesto, M. Mora and J. Tornero. **Supervisión, teleoperación y navegación de vehículos industriales y su integración en el sistema de gestión.** *Revista Iberoamericana de Automática e Informática Industrial*. 2005
- [122] L. Armesto and J. Tornero. **AutoTrans: Management and transport automation in warehouses.** *Industrial Simulation Conference*. 2005
- [123] [www.vision.caltech.edu/bouguetj/calib\\_doc](http://www.vision.caltech.edu/bouguetj/calib_doc)
- [124] G. Bradski and A. Kaebler. **Learning OpenCV.** *O'Reilly Media*. 2008

# **ANEXOS**



# **ANEXO 1.**

## **Métodos de Generación de Trayectorias**



## Mapas de Carreteras y Grafos de Visibilidad

Como principales características destacan las siguientes:

- Identifican un conjunto de rutas libres del espacio.
- Suelen trabajar con objetos poligonales.
- Los nodos del mapa son los vértices de los objetos.

Como ya se ha comentado se asume que existe un mapa topológico, por lo tanto, esta técnica requiere conocimiento previo del entorno, lo cual permite realizar un procesado *off-line* para extraer las principales características.

Entre sus principales usos destaca su inclusión en los sistemas GPS comerciales y en algunos almacenes de logística para el control de circulación de AGVs, tal y como se observa en la Figura 196.

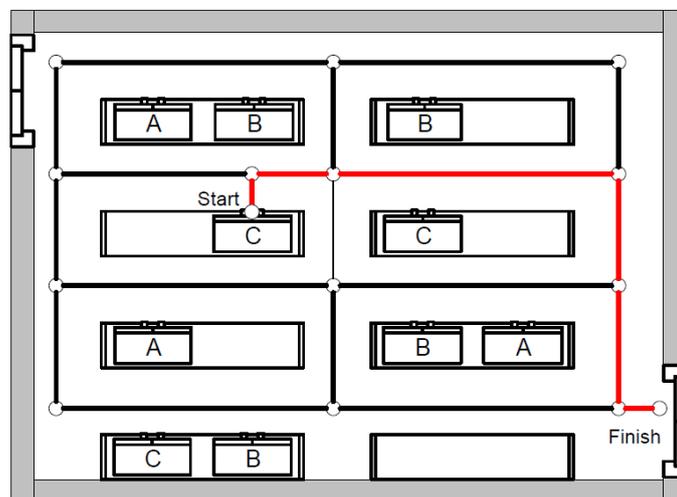


Figura 196. Ejemplo de mapa de carreteras en entorno industrial.

## Descomposición Celular

El objetivo de esta técnica es dividir el espacio mediante discretización, de modo que se obtengan elementos más pequeños que se puedan identificar fácilmente. Las celdas discriminan entre el espacio libre y el espacio ocupado.

### *Descomposición Regular*

Como principales características destacan las siguientes:

- El espacio se descompone de manera regular.
- Es una técnica muy costosa en cuanto a recursos de memoria y tiempo de cómputo.

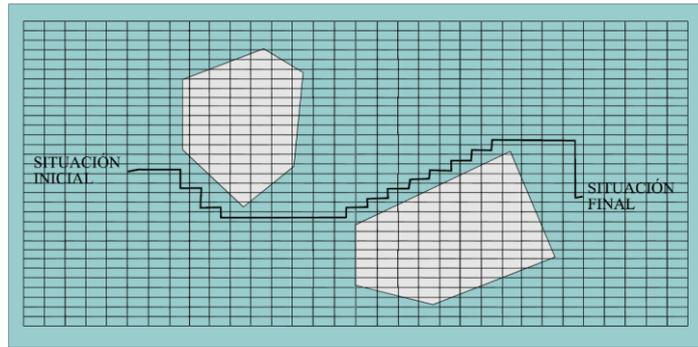


Figura 197. Descomposición celular regular.

### ***Descomposición Trapezoidal***

Como principales características destacan las siguientes:

- El espacio se descompone de manera irregular.
- Se pueden realizar barridos horizontales, verticales, etc.

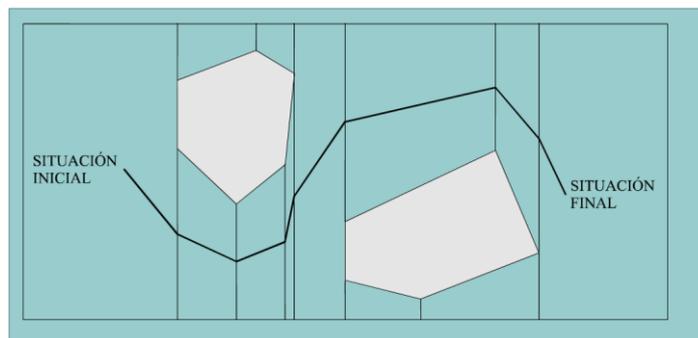


Figura 198. Descomposición celular trapezoidal.

### ***Descomposición por Árboles Octales (Oct-trees/Quad-trees)***

Como principales características destacan las siguientes:

- Se trata de una representación eficiente del espacio ocupado y libre.
- Para el caso 3D se utiliza la representación Oct-tree.
- Su manejo requiere de estructuras arbóreas.

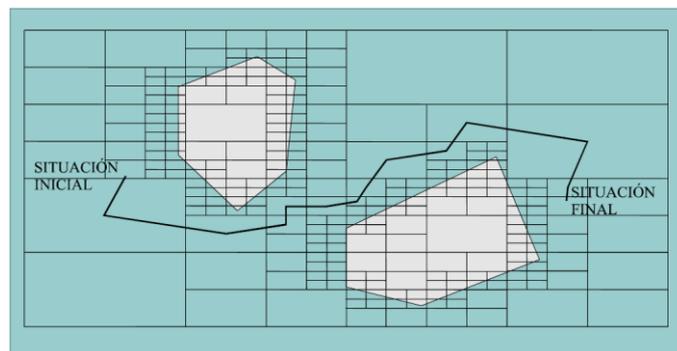


Figura 199. Descomposición celular por árboles octales.

## Grafos de Visibilidad

El objetivo es unir todos los vértices que sean visibles entre sí, de modo que se tenga el camino más corto entre ellos. El procedimiento para generar un grafo de visibilidad es un método iterativo que se encarga de realizar todas las conexiones entre los vértices hasta completar el grafo.

- Unir todos los vértices de los objetos visibles desde el  $q_{start}$ .
- Unir todos los vértices de los objetos visibles desde el  $q_{goal}$ .
- Para cada vértice, unir dicho vértice con los demás vértices visibles (Repetir hasta completar).

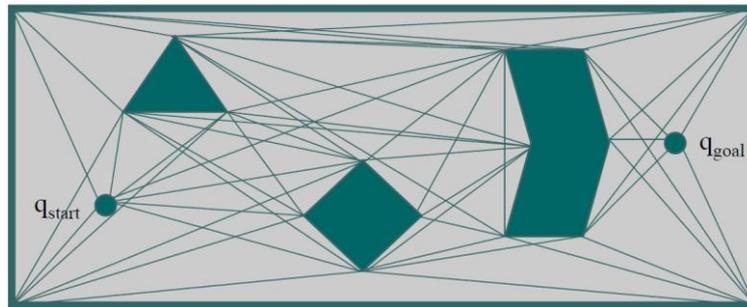


Figura 200. Ejemplo de Grafo de Visibilidad.

## Diagrama de Voronoi

El Diagrama Generalizado de Voronoi (GVD) es el lugar geométrico cuya distancia es la misma para los objetos más cercanos. El objetivo es generar el espacio de soluciones más seguro, es decir, aquél que esté más alejado de los obstáculos.

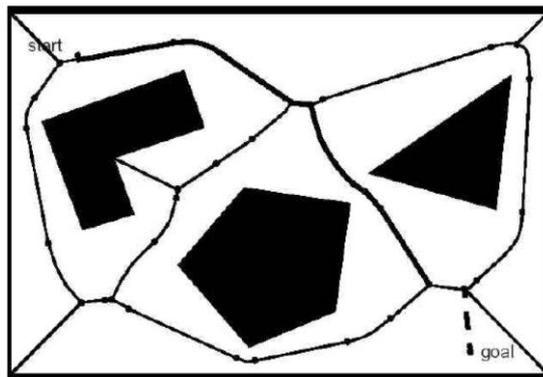


Figura 201. Ejemplo de Diagrama de Voronoi

## Campos Potenciales Artificiales

El robot móvil se comporta como un cuerpo bajo la influencia de un campo potencial artificial  $U(q)$  generado por la configuración del punto destino y los obstáculos, donde  $U(q)$  se define (aunque no es la única aproximación) como la suma algebraica de:

- Un *potencial atractivo*, que atrae el robot hacia la configuración final.
- Un *potencial repulsivo*, que tiende a alejarlo de los obstáculos.

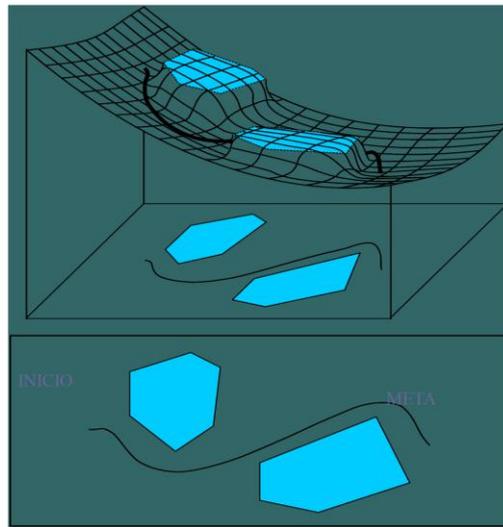


Figura 202. Ejemplo de Campo de Potencial Artificial

Como principales características de este método de planificación de movimientos destacan las siguientes:

- No Requiere Conocimiento Previo del Entorno.
- Aplicable a los Espacios Cartesianos y de Configuraciones.
- Fácil Extensión a Espacios Multi-dimensionales.
- Método Local o Global.
- Procesado *on-line* en Tiempo Real.

## **ANEXO 2.**

# **Control de Robots Móviles**



## ***Modelado Cinemático***

Para poder hacer un control sobre un robot móvil es necesario disponer de la información cinemática: posición, orientación y velocidades; así como información sobre el entorno que lo rodea. También es interesante conocer cuál es la siguiente configuración que se desea alcanzar para que sirva como referencia al sistema de control del vehículo.

Hay que decir que un modelo cinemático es un conjunto de ecuaciones que definen el comportamiento teórico del robot, de modo que conociendo unas variables de entrada se obtengan otras variables, en este caso de salida. Existen básicamente dos tipos de modelos cinemáticos:

- **Modelo directo:** a partir de las variables de velocidad y posición de los motores se puede determinar la posición global del robot en el entorno en el que se encuentre.
- **Modelo inverso:** a partir de la posición en el espacio cartesiano que se desee para el robot se determinan los valores de las acciones de control que deben de tener los distintos motores para llevar al robot a dicha posición.

De forma genérica, según la distribución de sus ruedas y en cuáles de ellas se disponen los motores o actuadores encargados del movimiento se pueden obtener distintos modelos cinemáticos de robots [100]. Por eso en primer lugar se van a detallar las principales configuraciones que suelen tener la mayoría de los robots y vehículos autónomos: la configuración triciclo y la diferencial.

A continuación, antes de realizar la descripción de los distintos modelos cinemáticos, se muestra una lista con todas las variables que intervienen en los modelos.

- $X, Y$ : Sistema de coordenadas global (mundo).
- $x_r, y_r$ : Sistema de coordenadas local (robot).
- $x, y$ : Posición Cartesiana del robot con respecto al sistema global.
- $x', y'$ : Velocidad Cartesiana del robot con respecto al sistema global.
- $\theta$ : Orientación del robot.
- $\omega$ : Velocidad angular del robot.
- $v_i, v_d$ : Velocidades de las ruedas fijas.
- $s_i, s_d$ : Desplazamiento de las ruedas fijas.
- $c_i, c_d$ : Cuentas de las ruedas fijas.
- $v_t$ : Velocidad de la rueda orientable.
- $\alpha$ : Orientación de la rueda orientable.

## Configuración Diferencial

Dicha configuración es aquella en la que el robot dispone de dos ruedas fijas que tienen movimiento independiente y que están situadas en un mismo eje de rotación, que está en el medio del eje de tracción. Además de las ruedas motrices puede estar formado también por una tercera rueda con movimiento libre (rueda castor o loca), en general situada en la parte delantera de la base del robot, para darle mayor estabilidad al movimiento. En cualquier caso no dispone de ninguna rueda orientable, es decir, no hay ningún motor o servomotor capaz de rotar una de las ruedas sobre un eje perpendicular al plano de movimiento del robot. En la Figura 203 se muestra la configuración de un robot diferencial.

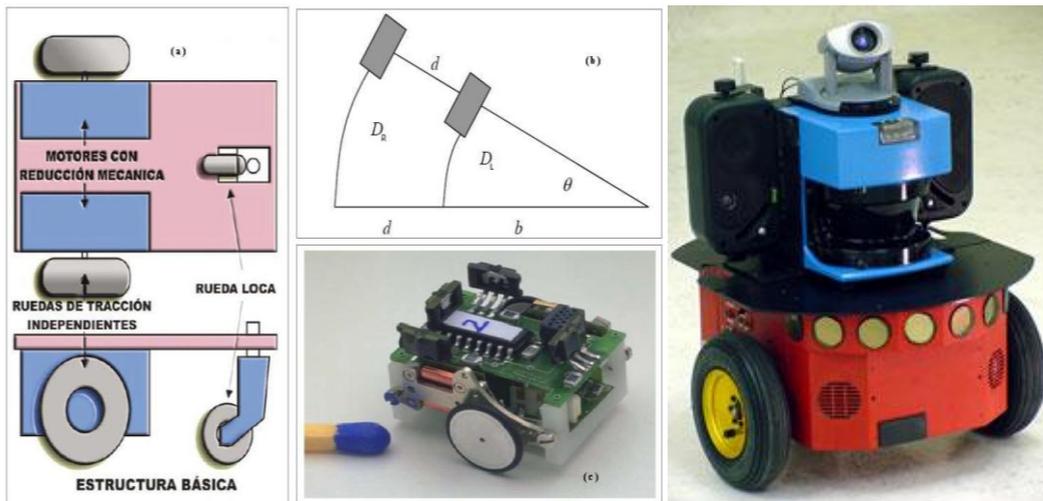


Figura 203. Varios modelos de robots con configuración diferencial.

## Modelo Cinemático Directo

A continuación se muestra una imagen en la que se pueden ver las distintas variables que intervienen en el modelo para la configuración diferencial.

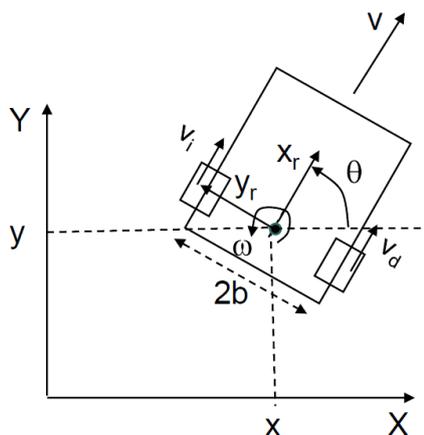


Figura 204. Modelo de la configuración diferencial.

Suponiendo que  $2b$  es el valor de la separación entre las ruedas de tracción, se llega a las siguientes expresiones matemáticas:

$$\begin{aligned} \Delta s_{i/d} &= \frac{r \cdot \Delta c_{i/d}}{PPV} & v_{i/d} &= \frac{\Delta s_{i/d}}{\Delta T} \\ \Delta s &= \frac{\Delta s_d + \Delta s_i}{2} & \Delta \theta &= \frac{\Delta s_d - \Delta s_i}{2b} \\ v &= \frac{\Delta s}{\Delta T} & \omega &= \frac{\Delta \theta}{\Delta T} \end{aligned} \quad (176)$$

Finalmente se obtienen las siguientes expresiones matriciales:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \frac{1}{2b} \begin{bmatrix} b \cdot \cos \theta & b \cdot \cos \theta \\ b \cdot \sin \theta & b \cdot \sin \theta \\ -1 & 1 \end{bmatrix} \quad (177)$$

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} = \frac{1}{2b} \begin{bmatrix} b \cdot \cos \theta & b \cdot \cos \theta \\ b \cdot \sin \theta & b \cdot \sin \theta \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta s_i \\ \Delta s_d \end{bmatrix} = \begin{bmatrix} \Delta s \cdot \cos \theta \\ \Delta s \cdot \sin \theta \\ \Delta \theta \end{bmatrix} \quad (178)$$

### **Modelo Cinemático Inverso**

A partir del modelo cinemático directo se puede obtener el modelo inverso. Simplemente hay que calcular la pseudoinversa. Las expresiones que se obtienen son las siguientes:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} \quad (179)$$

$$\begin{bmatrix} v_i \\ v_d \end{bmatrix} = \begin{bmatrix} 1 & -b \\ 1 & b \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (180)$$

$$\begin{bmatrix} v_i \\ v_d \end{bmatrix} = \begin{bmatrix} x' \cdot \cos \theta + y' \cdot \sin \theta - b \cdot \theta' \\ x' \cdot \cos \theta + y' \cdot \sin \theta + b \cdot \theta' \end{bmatrix} \quad (181)$$

Esta configuración tiene como principal inconveniente que si  $\theta' = 0$ , entonces  $\omega = 0$  y por tanto el robot no gira. Por ejemplo, si queremos seguir referencias rectilíneas ( $\theta'_{ref} = 0$ ) el robot no es capaz de corregir su posición, salvo que incluyamos un término integral que modifique la  $\theta'$  aplicada.

Dados los problemas que aparecen en el modelo inverso de la configuración diferencial se hace una ligera modificación de modo que en vez de tomar como referencia el punto medio situado en el eje de las dos ruedas motrices, se toma un punto situado cierta distancia  $e$  respecto de dicho eje. Se obtiene así un modelo más robusto que el anterior denominado modelo cinemático diferencial con punto descentrado. El control para esta configuración es mucho más estable, evitándose las oscilaciones producidas por intentar controlar el punto situado sobre el eje de giro del robot, el cual, ante cambios grandes de la referencia tiende a desestabilizar al sistema.

## Configuración Diferencial con Punto Descentralizado

Las variables que intervienen en este modelo cinemático son las mismas que las del caso anterior. Tal y como se ha adelantado anteriormente, la única diferencia radica en que ahora el control no se efectúa sobre el eje del robot sino sobre un punto perpendicular a dicho eje y separado cierta distancia  $e$ . Dicha distancia es un nuevo parámetro que se incorpora al modelo y que se mantiene siempre con un valor constante.

### Modelo Cinemático Directo

A continuación se muestra la imagen que representa el modelo cinemático y se incluyen las ecuaciones que forman parte del mismo. Dichas ecuaciones son las que servirán para futuras estimaciones de posición y velocidad utilizadas en el control cinemático del robot.

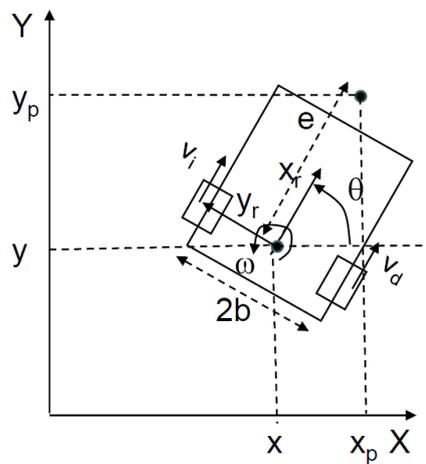


Figura 205. Modelo de la configuración diferencial con punto descentralizado.

Ecuaciones para la obtención del modelo cinemático directo:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} x + e \cdot \cos\theta \\ y + e \cdot \sin\theta \end{bmatrix} \quad (182)$$

$$\begin{bmatrix} x_p' \\ y_p' \end{bmatrix} = \begin{bmatrix} 1 & 0 & -e \cdot \sin\theta \\ 0 & 1 & e \cdot \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} \quad (183)$$

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} = \frac{1}{2b} \begin{bmatrix} b \cdot \cos\theta & b \cdot \cos\theta \\ b \cdot \sin\theta & b \cdot \sin\theta \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_i \\ v_d \end{bmatrix} \quad (184)$$

Finalmente se incluyen las ecuaciones completas del modelo cinemático directo:

$$p' = J(p) \cdot q' \quad (185)$$

$$\begin{bmatrix} x_p' \\ y_p' \end{bmatrix} = \frac{1}{2b} \begin{bmatrix} b \cdot \cos\theta + e \cdot \sin\theta & b \cdot \cos\theta - e \cdot \sin\theta \\ b \cdot \sin\theta - e \cdot \cos\theta & b \cdot \sin\theta + e \cdot \cos\theta \end{bmatrix} \cdot \begin{bmatrix} v_i \\ v_d \end{bmatrix} \quad (186)$$

## Modelo Cinemático Inverso

Análogamente a los realizado con el modelo cinemático de la configuración diferencial, para la configuración con punto descentralizado se obtiene el modelo cinemático inverso. La ecuación matricial que se obtiene es la siguiente:

$$q' = J^{-1}(p) \cdot p' \quad (187)$$

$$\begin{bmatrix} v_i \\ v_d \end{bmatrix} = \frac{1}{e} \begin{bmatrix} e \cdot \cos\theta + b \cdot \sin\theta & e \cdot \sin\theta - b \cdot \cos\theta \\ e \cdot \cos\theta - b \cdot \sin\theta & e \cdot \sin\theta + b \cdot \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x'_p \\ y'_p \end{bmatrix} \quad (188)$$

## Configuración Triciclo

Dispone de dos ruedas traseras y de una rueda delantera (o trasera) orientable. Sólo se puede actuar sobre dos variables, habitualmente la orientación de la rueda delantera y su velocidad de avance, de modo que en general las ruedas traseras son libres. No obstante, en muchos casos esto puede variar, por ejemplo, en el modelo en configuración triciclo utilizado en la experimentación de la presente Tesina, el vehículo tiene la rueda trasera orientable y dispone de dos ruedas delanteras para la tracción (ver Figura 206).



Figura 206. Carretilla elevadora con configuración triciclo.

La configuración triciclo además tiene mejores características en cuanto a adherencia (menor deslizamiento) que la configuración diferencial. Además permite sensorizar más ruedas (sensorización redundante), con lo que se pueden realizar estimaciones más precisas.

### Modelo Cinemático Directo

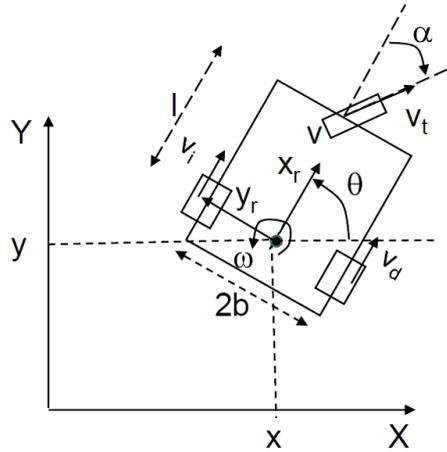


Figura 207. Modelo de la configuración triciclo.

Suponiendo que  $2b$  es el valor de la separación entre las ruedas de tracción y que  $l$  es la separación entre los ejes trasero y delantero, se llega a las siguientes expresiones matemáticas:

$$\begin{aligned} \Delta s_t &= \frac{r \cdot \Delta c_t}{PPV} & v_t &= \frac{\Delta s_t}{\Delta T} \\ \Delta s &= -\cos \alpha \frac{\Delta s_t}{\Delta T} & \Delta \theta &= -\frac{\sin \alpha}{l} \frac{\Delta s_t}{\Delta T} \\ v &= \frac{\Delta s}{\Delta T} & \omega &= \frac{\Delta \theta}{\Delta T} \end{aligned} \quad (189)$$

Finalmente se obtienen las siguientes expresiones matriciales:

$$\begin{aligned} \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} &= \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \\ \begin{bmatrix} v \\ \omega \end{bmatrix} &= \begin{bmatrix} -\cos \alpha \\ -\frac{\sin \alpha}{l} \end{bmatrix} \cdot v_t \end{aligned} \quad (190)$$

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos \theta \cdot \cos \alpha \\ \sin \theta \cdot \sin \alpha \\ -\frac{\sin \alpha}{l} \end{bmatrix} \cdot v_t \quad (191)$$

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} \cos \theta \cdot \cos \alpha \\ \sin \theta \cdot \sin \alpha \\ -\frac{\sin \alpha}{l} \end{bmatrix} \cdot \Delta s_t \quad (192)$$

## Modelo Cinemático Inverso

A partir del modelo cinemático directo se puede obtener el modelo inverso. Simplemente hay que calcular la pseudoinversa. Las expresiones que se obtienen son las siguientes:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} \quad (193)$$

$$\begin{bmatrix} v_t \\ \alpha \end{bmatrix} = \begin{bmatrix} \sqrt{v^2 + \omega^2 \cdot l^2} \\ -\arctan\left(\frac{l \cdot \omega}{v}\right) \end{bmatrix} = \begin{bmatrix} \sqrt{(x' \cdot \cos\theta + y' \cdot \sin\theta)^2 + \omega^2 \cdot l^2} \\ -\arctan\left(\frac{l \cdot \theta'}{x' \cdot \cos\theta + y' \cdot \sin\theta}\right) \end{bmatrix} \quad (194)$$

## Control de Movimiento de un Robot Móvil

El control cinemático del movimiento de un vehículo es el objetivo fundamental de la robótica móvil. Mediante las ecuaciones de los modelos cinemáticos directo e inverso, cada iteración o periodo de control se hace una estimación de la posición del robot y se determinan las acciones de control necesarias para conseguir los objetivos fijados.

Son muchas las estrategias de control que puede tener un robot móvil, aunque fundamentalmente se puede dividir en dos:

- **Control cinemático:** utiliza la posición del robot y en general se pretende seguir una trayectoria o un camino, la cual puede estar pre-programada o bien se detecta el camino a seguir por cualquier método o mediante cualquier tipo de sensor. Para dicho control se usan generalmente los sensores propioceptivos, ya que lo que se busca es llegar al objetivo suponiendo que el entorno está completamente libre de obstáculos, aunque también se usarán sensores estereoceptivos en los casos en los que se requiera información del entorno para estimar la posición o bien información de la trayectoria a seguir por ser un elemento externo al propio robot. Es por eso que con la simple estimación de la posición mediante los encoders del robot se puede realizar este tipo de control, aunque es recomendable utilizar técnicas de SLAM para realizar una localización más exacta.
- **Técnicas de control basadas en comportamiento:** mediante el uso de sensores estereoceptivos y según la función programada para el robot se consigue interactuar con el entorno. Dentro de este grupo destacan los comportamientos basados en reglas, la evitación de obstáculos o algunos métodos en los que es necesario el conocimiento a priori del entorno.

## Estimación de la Posición de un Robot Móvil

### Estimación mediante Odometría

Una vez obtenidas las ecuaciones del modelo se puede obtener la evolución en el tiempo de las variables de posición y orientación mediante la integración de dichas ecuaciones. A diferencia de otros sistemas de control, en los que se utilizan computadores de gran potencia, en los sistemas empujados la capacidad de cómputo es una de las mayores limitaciones. Como principal consecuencia de ello no se puede hacer una integración precisa ni mucho menos exacta. Lo que en realidad se hace es una aproximación mediante un algoritmo recursivo discreto que calcula la integral de dichas variables. Es evidente que el algoritmo tiene errores, pero puede considerarse válido para periodos de control muy pequeños.

A partir de las velocidades cartesianas y angular se tiene lo siguiente:

$$\begin{bmatrix} x'(t) \\ y'(t) \\ \theta'(t) \end{bmatrix} = \begin{bmatrix} v(t) \cdot \cos\theta(t) \\ v(t) \cdot \sin\theta(t) \\ \omega(t) \end{bmatrix} \quad (195)$$

Si se realiza la integral de la ecuación diferencial, cuya expresión recursiva es muy extensa. Por eso, cuando el incremento de tiempo es lo suficientemente pequeño, la integral se puede aproximar a la siguiente expresión:

$$\begin{bmatrix} x(t + \Delta t) \\ y(t + \Delta t) \\ \theta(t + \Delta t) \end{bmatrix} = \begin{bmatrix} x(t) + v(t) \cdot \Delta t \cdot \cos\theta(t) \\ y(t) + v(t) \cdot \Delta t \cdot \sin\theta(t) \\ \theta(t) + \omega(t) \cdot \Delta t \end{bmatrix} \quad (196)$$

Por lo tanto se llega a que en discreto la posición del robot se puede calcular a partir de una de estas dos expresiones matriciales:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + v_k \cdot T \cdot \cos\theta_k \\ y_k + v_k \cdot T \cdot \sin\theta_k \\ \theta_k + \omega_k \cdot T \end{bmatrix} \quad (197)$$

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \Delta s_k \cdot \cos\theta_k \\ y_k + \Delta s_k \cdot \sin\theta_k \\ \theta_k + \Delta\theta_k \end{bmatrix} \quad (198)$$

Cualquiera de las dos opciones es válida, aunque dado que se tienen codificadores de posición para las ruedas se opta por utilizar la segunda opción puesto que no estima la velocidad sino la diferencia de posición o incremento de espacio.

## Estimación mediante Técnicas de Scan Matching

El *Scan Matching* lo que pretende es calcular el desplazamiento que ha realizado el vehículo a partir de la estimación de la rotación y la translación relativa entre un escaneo y el siguiente (ver Figura 208). Generalmente se realiza mediante sensores láser, debido a su precisión, facilidad de uso y rapidez de medida, aunque pueden utilizarse otros tipos de sensores de rango.

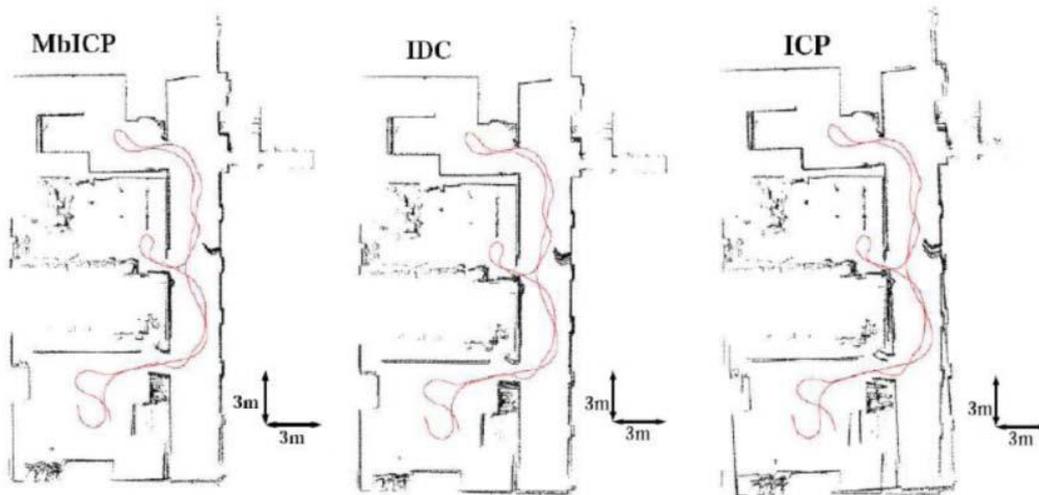


Figura 208. Resultados de diferentes algoritmos de *Scan Matching*: MbICP (izq.), IDC (centro) e ICP (dcha.).

Las mejoras que aportan las técnicas de *Scan Matching* en cuanto a precisión de la estimación son enormes, tal y como se observa en la Figura 209.

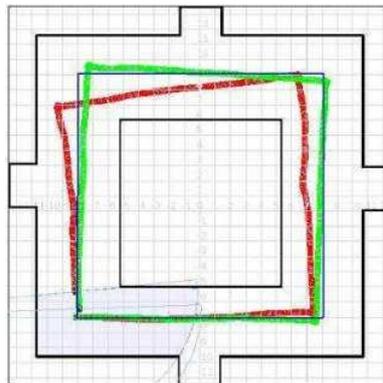


Figura 209. Estimación de la posición del robot: *Scan Matching* (verde) y odometría (rojo).



# **ANEXO 3.**

## **Soluciones Industriales para AGVs**



## ***Definición de AGV***

Como uno de los elementos principales utilizados para la implementación de los desarrollos de la presente Tesina es un AGV con diferentes dispositivos de sensorización y de control, en este anexo se ha creído conveniente incluir una breve explicación de los denominados AGV, así como de las soluciones tecnológicas existentes para su uso a nivel industrial

Un AGV (*Automatic Guided Vehicle*, en sus siglas en inglés) es un robot capaz de moverse de forma autónoma en un determinado entorno, realizando diversas tareas como el transporte o la manipulación de objetos y sin chocar con los elementos que lo rodean. Por lo tanto se puede decir que en general un vehículo auto-guiado consiste básicamente en una plataforma con ruedas, que dispone de un microprocesador incorporado y una fuente de alimentación propias, que le permiten moverse por sí mismo y recibir órdenes de una unidad de control o de otros AGV.

La aparición en el mercado del primer AGV data de los años 50, de la mano de la *Barrett Electronics of Northbrook*, empresa de Illinois, que desarrolló un modelo que tenía como única misión el transporte de mercancías pesadas y consistía en un tractor adaptado que seguía un conductor situado en el suelo, sustituyendo el sistema empleado basado en raíl. Actualmente los AGV incorporan toda una gama de herramientas, desde pallets para apilar contenedores hasta brazos robots, que les permiten realizar todo tipo de funciones de manipulación de objetos.

Hoy en día existen un gran número de vehículos auto-guiados con diferentes características, en función del entorno donde vaya a trabajar el sistema y las tareas que deba llevar a cabo. Sin embargo, las siguientes partes son comunes a todos los AGV:

- **Unidad principal de control** se compone de 3 partes:
  - **Unidad de control de conducción:** Encargada de los convertidores que alimentan los motores del sistema locomotor.
  - **Unidad de control:** Encargada de las tareas que debe llevar a cabo el robot, como la carga de mercancías o la manipulación de objetos mediante el brazo robot.
  - **Unidad de seguridad:** Encargada de evitar las colisiones con los elementos del entorno, sobre todo con otros objetos móviles. Debe de garantizar que ni los objetos manipulados ni los operarios de planta sufren ningún daño. Es una de las partes fundamentales que, junto al comportamiento predictivo, dota al sistema de mayor seguridad frente a los vehículos de control manual. Los elementos de seguridad básicos de los AGV se observan en la Figura 210.

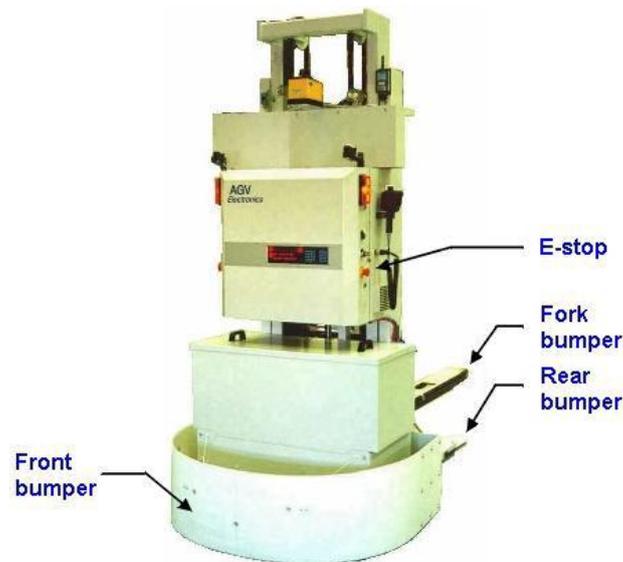


Figura 210. Sistemas de seguridad de un AGV

- **Fuente de alimentación:** Suele consistir en baterías eléctricas, cerradas y recargables que dotan al equipo con tanta autonomía de movimiento como capacidad tengan éstas.
- **Sistema locomotor:** Encargado del movimiento del dispositivo. El mecanismo utilizado para el desplazamiento dependerá del entorno de trabajo, y de las tareas a realizar. Los más utilizados son los basados en ruedas y orugas.
- **Dispositivo de sensorización:** Encargado de alertar a la unidad de seguridad de los elementos del entorno, así como de informar a la unidad de control para que se lleven a cabo las tareas de localización o del vehículo y el cálculo de la trayectoria a seguir.

En las líneas complejas de fabricación flexible, en donde el recorrido de las piezas entre máquinas y entre éstas y los almacenes es aleatorio y las distancias son considerables, el sistema de transporte flexible más utilizado es el que utiliza vehículos guiados automáticamente o AGVs.

Estos vehículos que se extienden desde repartidores con capacidad de carga de unos pocos kilogramos hasta transportadores de más de 100 toneladas pueden variar su ambiente de trabajo desde oficinas alfombradas hasta los más hostiles sectores fabriles. Éstos factores sumados a la flexibilidad que permiten han ampliado el campo de aplicación de los AGV de gran modo en los últimos diez años, extendiéndolo a las más diversas áreas como industrias químicas, de cerámicos, metalmecánicas, alimenticias, textiles, depósitos de todo tipo, empresas de servicios como correos, y otros sitios como aeropuertos y hoteles.

Como se ha comentado en la primera parte de este Anexo, los vehículos guiados automáticamente pueden definirse como vehículos autopropulsados, capaces de seguir automáticamente una trayectoria variable según un patrón flexible, es decir fácilmente modificable.

Este sistema de transporte puede desglosarse en tres partes:

- El vehículo.
- El sistema de tráfico.
- El sistema de gestión.

El sistema de tráfico está formado por:

- Los circuitos o caminos a recorrer.
- Los puntos de carga y descarga.
- Los puntos de comunicación.
- Indicadores de tráfico.

## Tecnologías tradicionales

En la actualidad existen diversas soluciones comerciales con AGVs y según el método utilizado para el guiado se pueden clasificar de dos maneras: de camino fijo o de camino libre. La determinación de cuál de los dos elegir dependerá de la Tabla 14.

CARACTERÍSTICA	SISTEMAS FIJOS	SISTEMAS FLEXIBLES
Costo	Menor	Mayor
Flexibilidad	Escasa	Alta
Cambios futuros	Altos costos por modificación y/o ampliación	Posibilidad de cambios de disposición acorde a requisitos de capacidad

**Tabla 14. Características para la selección del método de guiado de un AGV.**

Los primeros AGVs utilizaban un cable de guiado inductivo o una línea óptica visible, pintada sobre el suelo. El cable inductivo de guía es todavía uno de los sistemas más empleados en AGVs. No obstante, muchas compañías tratan de cambiar el guiado mediante cable inductivo por los sistemas más modernos de navegación, como pueden ser:

- Guías de imanes empotrados
- Laser de balizamiento ubicados en zonas específicas del almacén
- Microondas
- Sensores ultrasónicos
- Gyro inerciales
- Sistemas de guiado por cámara

## Filo-Guiado

El sistema de guiado magnético por cable se basa en un conductor eléctrico enterrado (unos 20 mm), y mediante una corriente eléctrica que fluye por este, se crea un campo electromagnético alrededor del mismo. El campo producido es más fuerte (intenso) cerca del conductor y se reduce a medida que se aleja del conductor.

Un campo electromagnético que atraviese una espiral inducirá un voltaje eléctrico a través de ésta, que puede detectarse a través de la terminación de la espiral y es proporcional a la intensidad del campo electromagnético.

La antena orientable contiene dos espirales en donde se induce el campo y la señal que se le envíe al motor del AGV dependerá de la ubicación de dicha antena. El AGV tendrá como mínimo una antena para orientarse y otra de cruz para detectar los cables-guías que estén perpendiculares entre sí. Los cables de cruz se usan para actualizar la posición exacta del AGV. Muchos AGV también tendrán antenas adicionales de guía para trasladarse marcha atrás.

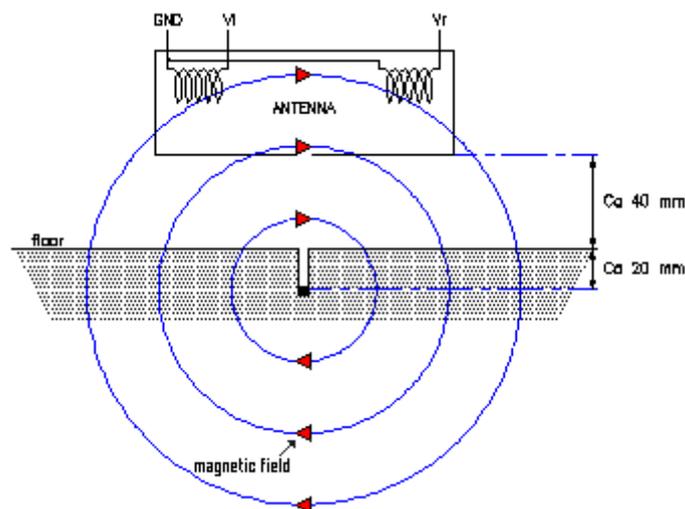


Figura 211. Esquema de funcionamiento del sistema de guiado por cable magnético.

La corriente que circula por el conductor es de muy baja intensidad, unos 400 mA a unos 40 Voltios y a baja frecuencia. El campo magnético generado interactúa con dos bobinas situadas en la parte delantera del vehículo, que en función de la diferencia de las señales en cada bobina autocorrigue su trayectoria. Cada tramo del circuito tiene una frecuencia distinta y esto permite al vehículo cambiar de trayectoria.

## ***Guiado Óptico***

El AGV sigue una línea marcada en el suelo con una sustancia fluorescente que, al ser activada mediante una luz ultravioleta, es detectada por dos células fotoeléctricas en la base del vehículo. Al variar la frecuencia de la luz ultravioleta, la señal detectada es distinta y permite pasar de un tramo del circuito a otro tramo. Si una célula se sale fuera de la línea, no percibe señal y el vehículo corrige su posición hasta que las dos células estén activas.

Este sistema, muy fácil, barato de instalar y modificar, tiene el inconveniente de precisar unas condiciones de suelo y de ambiente muy apropiados que en general no son las de una industria.

## ***Guiado por Laser de Balizamiento***

El explorador de Laser se monta en una posición alta sobre el AGV y deber tener visibilidad a todos los puntos de referencia. Estos puntos de referencia se encuentran recubiertos por una cinta reflectora. Se debe tener un mínimo de tres puntos de referencia durante el viaje.

El explorador de laser se basa en coordenadas  $x - y$  con una exactitud de  $\pm 2 \text{ mm}$  de tolerancia que se conecta directo a la placa de control del AGV. Dicho explorador de laser usa un haz de laser emitido por un transmisor de pulso que repasa el área de circunvalación mediante un espejo interno que rota a  $10 \text{ rev/seg}$ . El mismo haz es reflejado por los puntos de referencia, esto determinará las posiciones de referencia fija a través del método de triangulación realizado por un microprocesador interno del explorador, logrando detectar la posición del vehículo en todo momento.

La programación de trayectoria es fácil, ya que se define y modifica utilizando generalmente AutoCAD, permitiendo al usuario modificar o agregar una línea de trayectoria libremente.

Entre las principales ventajas destacan las siguientes:

- No requiere ninguna calibración.
- Fácil instalación de los reflectores.
- Alta tolerancia para el bloqueo de reflectores con ningún efecto sobre la trayectoria.
- El vehículo podrá conducirse donde quiera mientras puede actualizar su posición absoluta a través de los reflectores.

La principal desventaja de este sistema de guiado es:

- 15-30 % más caro que los vehículos de filo-guiado.

## ***Guiado Magnético***

El imán-guía orientado es un sistema de navegación que posee sensores de imanes ubicados en posiciones especiales, para encontrar imanes pequeños instalados a lo largo de la trayectoria guía en el piso. Los sensores guías informarán de las posibles desviaciones y actualizará la trayectoria del AGV.

Los imanes se instalan cada 5 o 10 metros sobre la trayectoria a recorrer, en hoyos perforados en el suelo y cubiertos con *epoxy*, dejando la superficie plana y lisa. Éstos son capaces de detectar imanes pequeños enterrados en el piso con alta exactitud, dando las coordenadas en x-y de los mismos. Además, el sensor guía se encuentra en el eje único, lo que dará la dirección al vehículo y asiste el voltaje proporcional al motor para lograr el desplazamiento del mismo.

## **Nuevas Tecnologías**

El mayor inconveniente que presentan los AGVs tradicionales es su poca flexibilidad para adaptarse a los cambios del entorno, el coste asociado a la infraestructura que requieren, en ocasiones no pueden re-planificar su ruta, etc.

Con objeto de aumentar el grado de autonomía de los AGV, los investigadores han desarrollado en las últimas décadas numerosas herramientas para resolver los problemas que se presentan tradicionalmente en la robótica móvil, tales como, simulación [112], la localización y la construcción de mapas (SLAM) [113], navegación y planificación de trayectorias [114], [115], [116], coordinación entre robots, etc. En este sentido, se requiere cada vez más sensores con mayor capacidad de procesamiento y abstracción, como por ejemplo los sistemas de visión, que están adquiriendo una importancia cada vez mayor debido a la continua mejora de la tecnología. Tómese como ejemplo la aplicación ROBOLIFT© [117], en la que se utiliza un sistema de visión para detectar marcas artificiales en el suelo. En [118] y [119], se resuelve el problema maniobras de carga y descarga con pallets, utilizando para ello un sistema de visión.

En [120], [121], [122], se desarrollaron varias aplicaciones industriales con AGVs, en las que fundamentalmente se realizaron diversas propuestas de automatización, definiendo arquitecturas hardware y software, orientadas a la tele-operación de vehículos, la navegación autónoma o el seguimiento de línea por visión artificial.

A nivel más industrial, los nuevos sistemas de orientación de AGVs se llaman Sistemas Libres de navegación de gama. Sin embargo todos los sistemas de orientación de AGV requieren de referencias instaladas a lo largo o alrededor de una trayectoria de guía, ninguno es 100% libre. Estos sistemas tienen la desventaja de que el software puede ser muy complejo y para su programación y mantenimiento se requieren expertos.

## Navegación inercial

Uno de estos sistemas de guiado está basado en navegación inercial, en la que acelerómetros y giróscopos, junto con *transponders* embebidos en el suelo permiten localizar la posición del vehículo.

Por lo tanto, el sistema de guía inercial confía en la retroalimentación para estimar la distancia viajada. El AGV va midiendo la distancia recorrida y controla la posición, esto permite salirse de la referencia pero conociendo en todo momento la localización del vehículo, de modo que es sencillo volver al recorrido de guía.

## Visión Artificial

Otro tipo de guiado es aquel en el que se realiza un seguimiento de línea por visión artificial, en cuyo caso, el objetivo es que el AGV siga un circuito pintado en el suelo utilizando un sistema de visión.

En la Figura 212 se muestran dos AGVs pertenecientes al Instituto de Diseño y Fabricación (IDF) de la UPV, para los que se han implantado sistemas de visión para realizar el guiado del vehículo por una planta industrial.

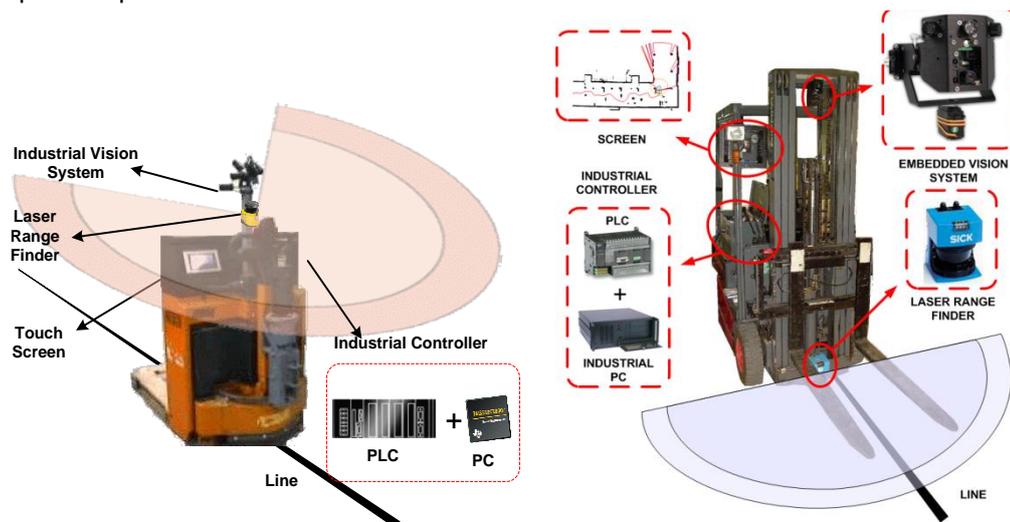


Figura 212. AGVs guiados mediante visión.



# **ANEXO 4.**

## **Proceso de Calibración**



## Proceso de Calibración

Para la calibración de la cámara CMUcam3, utilizada en los experimentos de control realizados con la carretilla industrial con función AGV, se ha utilizado el *Matlab Calibration Toolbox* [123], obteniendo tanto los parámetros intrínsecos como los extrínsecos, esto es la transformación homogénea capaz de expresar las coordenadas de los puntos en el sistema de coordenadas de la cámara a puntos referidos al sistema de coordenadas de la imagen. Despreciando posibles aberraciones de la óptica, la matriz de parámetros intrínsecos, según el modelo *pin-hole*, es  $K = \begin{bmatrix} A & 0 & U_0 \\ 0 & B & V_0 \\ 0 & 0 & 1 \end{bmatrix}$ , donde  $A = \frac{S_U \cdot f}{CCD_U}$  y  $B = \frac{S_V \cdot f}{CCD_V}$ , siendo  $S_U$  y  $S_V$  el tamaño de la imagen,  $CCD_U$  y  $CCD_V$  el tamaño del CCD (*Charge-Coupled Device*) y  $f$  la distancia focal.

Una vez conocidos los parámetros intrínsecos es necesario obtener los parámetros extrínsecos, es decir, los que permiten referir el sistema de coordenadas del centro focal con respecto a la base de la cámara. Para determinar las transformaciones homogéneas necesarias hay que definir los sistemas de coordenadas que intervienen en la transformación. A continuación se muestran dos esquemas (Figuras 213 (a) y (b)), en las que se puede observar cómo se distribuyen los SC, así como los parámetros que se van a considerar:

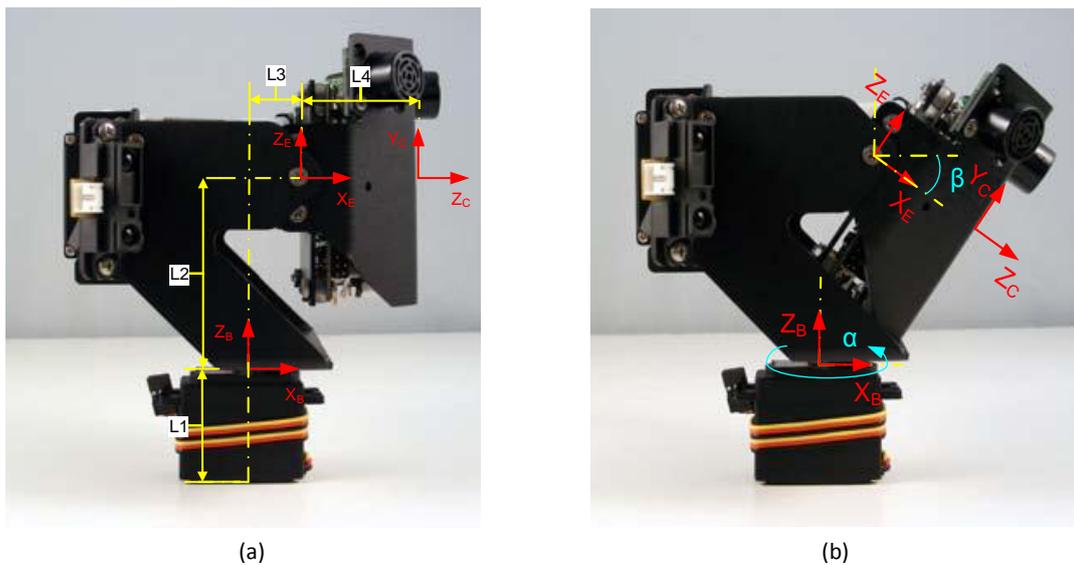


Figura 213. Sistemas de coordenadas de la torreta de la cámara.

En la Figura 213 (a) se muestra de forma detallada el sistema de coordenadas de la base que corresponde con el giro vertical del primero de los dos servomotores que controlan la orientación de la cámara. Además hay otro sistema de coordenadas situado en el segundo eje de giro y finalmente otro situado en el foco de la cámara. En la Figura 213 (b) aparecen los parámetros de giro ( $\alpha$  y  $\beta$ ), que en este caso son variables en función de las posiciones de los servos.

A continuación se muestran las distintas matrices de transformación homogéneas (MH) utilizadas:

$${}^W\mathbf{T}_B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{Desplazamiento en } Z \quad (199)$$

$${}^W\mathbf{R}_B = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{Giro de } \alpha \text{ en } Z \quad (200)$$

Las siguientes matrices permiten referir el centro focal de la cámara con respecto a la base de la torreta:

$${}^B\mathbf{T}_E = \begin{bmatrix} 1 & 0 & 0 & L3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{Desplazamientos en } X \text{ y } Z \quad (201)$$

$${}^E\mathbf{T}_C = \begin{bmatrix} 1 & 0 & 0 & L4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{Desplazamiento en } X \quad (202)$$

$${}^E\mathbf{R}_{1,C} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{Giro de } \pi/2 \text{ en } Z \quad (203)$$

$${}^E\mathbf{R}_{2,C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\beta) & -\sin(\beta) & 0 \\ 0 & \sin(\beta) & -\cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{Giro de } \beta \text{ en } X \quad (204)$$

Hay que destacar que en todo momento se están aplicando transformaciones respecto al sistema de coordenadas local y por lo tanto cada nueva matriz de transformación que se aplica debe de post-multiplicar al conjunto de matrices que se tiene hasta ese momento.

La matriz de transformación homogénea (MH) final que modela los parámetros extrínsecos de la cámara es la siguiente:

$$\mathbf{H}(\alpha, \beta) = {}^W\mathbf{H}_C = {}^W\mathbf{T}_B \cdot {}^W\mathbf{R}_B \cdot {}^B\mathbf{T}_E \cdot {}^E\mathbf{T}_C \cdot {}^E\mathbf{R}_{1,C} \cdot {}^E\mathbf{R}_{2,C} \quad (205)$$

Siendo  $\mathbf{T}$  y  $\mathbf{R}$  matrices de traslación y rotación homogéneas, respectivamente.

Por tanto, la ecuación que modela la MH directa para obtener las coordenadas de los píxeles a partir de los puntos del espacio es:

$$\mathbf{P}_{2D} = \mathbf{K} \cdot \mathbf{H}(\alpha, \beta) \cdot \mathbf{P}_{3D} \quad (206)$$

Siendo  $\mathbf{P}_{2D} \in \mathbb{R}^3$  el punto homogéneo respecto al sistema de coordenadas de la imagen,  $\mathbf{P}_{3D} \in \mathbb{R}^4$  el punto homogéneo en el espacio 3D respecto al sistema de referencia global o en este caso al SC del robot,  $\mathbf{K} \in \mathbb{R}^{3 \times 3}$  la matriz de parámetros intrínsecos y  $\mathbf{H} \in \mathbb{R}^{4 \times 4}$  la matriz homogénea de los parámetros extrínsecos de la cámara.

El modelo de proyección inversa que permite obtener puntos sobre el plano  $z = 0$ , requiere eliminar la cuarta fila y la tercera columna de  $\mathbf{H}$ . Obteniendo, por tanto, el siguiente modelo de proyección inversa:

$$\tilde{\mathbf{P}}_{3D} = \mathbf{K}^{-1} \cdot \tilde{\mathbf{H}}(\alpha, \beta)^{-1} \cdot \mathbf{P}_{2D} \quad (207)$$

Siendo:

$$\tilde{\mathbf{H}}(\alpha, \beta) = \begin{bmatrix} -\sin(\alpha) & -\cos(\alpha) \cdot \cos(\beta) & (L3 + L4) \cdot \cos(\alpha) \\ \cos(\alpha) & -\sin(\alpha) \cdot \cos(\beta) & (L3 + L4) \cdot \sin(\alpha) \\ 0 & \sin(\beta) & L1 + L2 \end{bmatrix} \quad (208)$$

Y siendo  $\tilde{\mathbf{P}}_{3D} = [X \ Y \ S]^T \in \mathbb{R}^3$  el punto homogéneo en el plano  $z = 0$ . Por tanto, para obtener las coordenadas que realmente se proyectan se debe normalizar el resultado obtenido. Para ello basta con dividir las coordenadas obtenidas por el valor  $S$ , que es el valor correspondiente al escalado. El resultado es el siguiente:

$$\tilde{\mathbf{P}}_{3DP} = [X_F \ Y_F \ 1]^T = \left[ \frac{X}{S} \ \frac{Y}{S} \ 1 \right]^T \quad (209)$$

## Cálculo de una Homografía

Al realizar la suposición de que el vehículo autónomo se desplaza por el plano  $z = 0$ , se está simplificando notablemente el problema de navegación. Eso implica que en todo momento los planos de imagen y de tierra están en la misma posición relativa. Como consecuencia y dado que la proyección perspectiva es invertible, es posible realizar la estimación de correspondencias entre los puntos de la imagen y los del mundo y viceversa.

Existe todo un campo en visión por computador dedicado a las proyecciones perspectivas, sobre todo desde el punto de vista de visión 3D. No obstante, aquí sólo se va a hacer referencia a la técnica del *Inverse Perspective Mapping* (IPM), que consiste básicamente en realizar la transformación inversa a la perspectiva producida por la cámara respecto al plano de desplazamiento del vehículo. Esto se consigue gracias al cálculo de los parámetros de la Homografía entre dichos planos, que no es más que una matriz homogénea de  $3 \times 3$ .

Simplemente se mencionará la función de la librería OpenCV que se ha utilizado para el cálculo de la homografía que permite realizar la inversión de la perspectiva y por tanto permiten conocer la correspondencia exacta entre los puntos detectados en la imagen y sus homólogos en el mundo. Dicha función es "*cvGetPerspectiveTransform*" y su funcionamiento puede consultarse en [124].

En la Figura 214 aparece un ejemplo de un proceso de inversión de la proyección tras el cálculo de una homografía, ya que en (a) aparece la imagen con los puntos seleccionados y en (b) aparecen los puntos mapeados sobre el plano  $z = 0$ .



Figura 214. Ejemplo de transformación mediante el método IPM.

Además del anterior ejemplo también se muestra la inversión de la proyección realizada tras obtener la matriz homogénea de transformación para la cámara IP montada sobre el robot *Pioneer 3DX*.

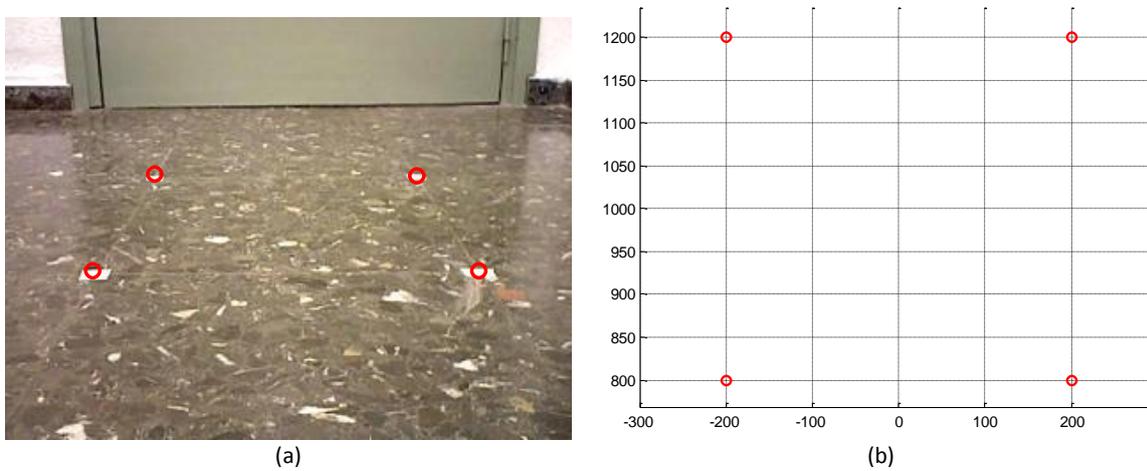


Figura 215. Transformación mediante el método IPM para la cámara IP montada sobre el robot *Pioneer 3DX*.

## **ANEXO 5.**

# **Protocolo de Comunicación Serie**



## Protocolo de Comunicación Serie

Por las características del hardware embebido utilizado, es decir, por las limitaciones de procesamiento de la CMUcam3, es necesario que ésta se comuniquen con un ordenador. Para solucionar dicho problema se ha realizado la comunicación entre ambos dispositivos a través del puerto serie del ordenador. Respecto a la comunicación hay que decir que se ha implementado un protocolo sencillo en el que el telegrama sigue la siguiente estructura:

STX (1 byte)	LEN (2 bytes)	CMD (1 byte)	DATA (LEN-1 bytes)	TIME (4 bytes)	CRC (1 byte)
-----------------	------------------	-----------------	-----------------------	-------------------	-----------------

Figura 216. Campos del telegrama de comunicación

Como en todos los protocolos en primer lugar se envía una cabecera para reconocer el inicio de la trama. En este caso el carácter de inicio de telegrama es el STX (0x02 ASCII).

Después se envían 2 bytes (LEN) en los que se indica el número de caracteres que se van a enviar, de modo que se incluyan tanto el byte CMD, en el que se indica el código del comando que se va a enviar, como los datos de la trama (DATA). En cualquier caso, en el campo LEN no se considera el byte de codificación CRC. Cuando el mensaje lo envía la cámara, además de lo anterior en el campo TIME del telegrama se almacena también el tiempo de procesamiento en una variable de tipo *float* (4 bytes). Finalmente, se realiza una XOR de todos los bytes (excepto el primero que al ser siempre el mismo no se considera) y el resultado se almacena en la última posición de la trama formando el denominado CRC que sirve para determinar si el mensaje es correcto o no.

Al tratarse de un procedimiento común a todos los comandos, a continuación se muestra el proceso de comunicación del protocolo, donde se aprecia qué secuencia se sigue durante el intercambio de telegramas:

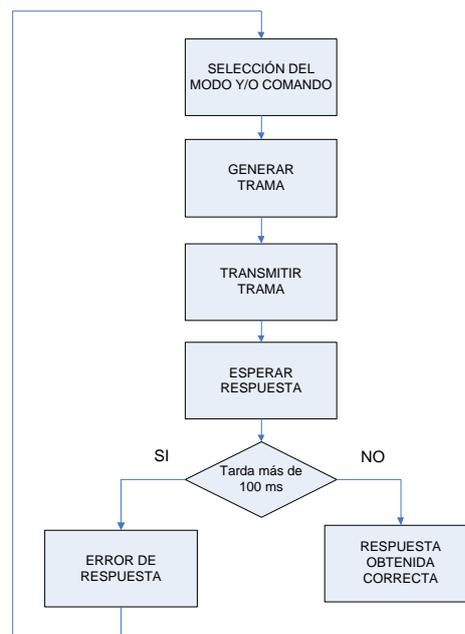


Figura 217. Proceso de comunicación del protocolo

A continuación se muestran unos diagramas con el proceso de formación de los telegramas de comunicación.

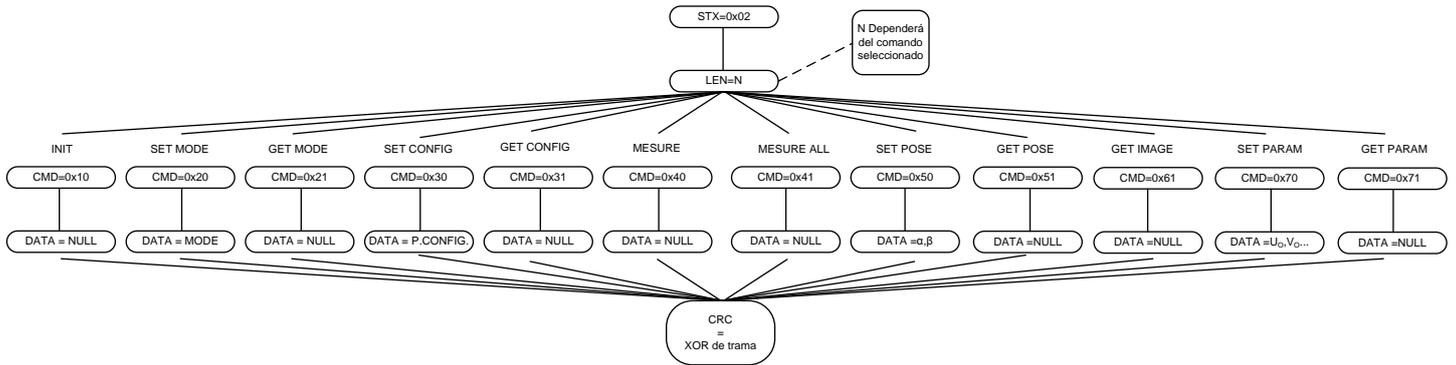


Figura 218. Proceso de formación telegrama de comunicación desde PC

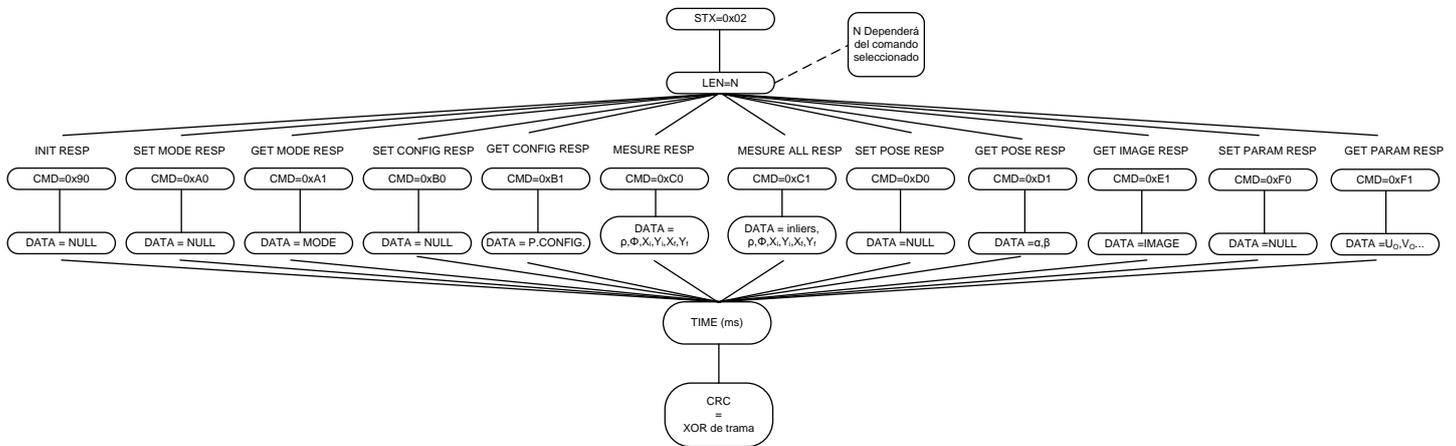


Figura 219. Proceso de formación telegrama de comunicación desde CMUcam3

Hay que destacar que el protocolo dispone de un total de 12 comandos, llamadas desde la aplicación del ordenador, así como 12 respuestas desde la CMUcam3:

• **Comandos de llamada.**

- INIT\_CMUCAM (CMD=0x10): Inicialización de los parámetros de la cámara a sus valores por defecto y posicionamiento de la misma a los ángulos iniciales.
- SET\_MODE\_CMUCAM (CMD=0x20): Cambio del modo de funcionamiento (Automático o Manual).
- GET\_MODE\_CMUCAM (CMD=0x21): Lectura del modo de funcionamiento.
- SET\_CONFIG\_CMUCAM (CMD=0x30): Cambio de los parámetros de configuración de la cámara (*Threshold, Contrast, MinInliers, Umbral, etc.*).
- GET\_CONFIG\_CMUCAM (CMD=0x31): Lectura de los parámetros de configuración de la cámara (*Threshold, Contrast, MinInliers, Umbral, etc.*).
- MESURE\_CMUCAM (CMD=0x40): Solicitud de los parámetros de la recta, sin transmisión de los puntos *inliers*.
- MESURE\_ALL\_CMUCAM (CMD=0x41): Solicitud de los parámetros de la recta, con transmisión de los puntos *inliers*.
- SET\_POSE\_CMUCAM (CMD=0x50): Cambio de posición de los servos.

- GET\_POSE\_CMUCAM (CMD=0x51): Lectura de posición de los servos.
  - GET\_IMAGE\_CMUCAM (CMD=0x61): Solicitud de transmisión de la imagen capturada.
  - SET\_PARAM\_CMUCAM (CMD=0x70): Cambio de los parámetros intrínsecos de la cámara ( $\alpha$ ,  $\beta$ ,  $u_o$ ,  $v_o$ ).
  - GET\_PARAM\_CMUCAM (CMD=0x71): Lectura de los parámetros intrínsecos de la cámara ( $\alpha$ ,  $\beta$ ,  $u_o$ ,  $v_o$ ).
- **Comandos de respuesta.**
    - INIT\_RESP\_CMUCAM (CMD=0x90): Respuesta de la cámara si todo es correcto al recibir el comando INIT\_CMUCAM.
    - SET\_MODE\_RESP\_CMUCAM (CMD=0xA0): Respuesta de la cámara si todo es correcto al recibir el comando SET\_MODE\_CMUCAM.
    - GET\_MODE\_RESP\_CMUCAM (CMD=0xA1): Transmisión del modo de funcionamiento en el que está trabajando la cámara.
    - SET\_CONFIG\_RESP\_CMUCAM (CMD=0xB0): Respuesta de la cámara si todo es correcto al recibir el comando SET\_CONFIG\_CMUCAM.
    - GET\_CONFIG\_RESP\_CMUCAM (CMD=0xB1): Transmisión de los parámetros actuales de configuración almacenados en la cámara.
    - MEASURE\_RESP\_CMUCAM (CMD=0xC0): Transmisión de los parámetros de la línea detectada ( $\rho, \Phi, X_i, Y_i, X_f, Y_f$ ).
    - MEASURE\_ALL\_RESP\_CMUCAM (CMD=0xC1): Transmisión de los parámetros de la línea detectada ( $\rho, \Phi, X_i, Y_i, X_f, Y_f$ ) y también los puntos *inliers*.
    - SET\_POSE\_RESP\_CMUCAM (CMD=0xD0): Respuesta de la cámara si todo es correcto al recibir el comando SET\_POSE\_CMUCAM.
    - GET\_POSE\_RESP\_CMUCAM (CMD=0xD1): Transmisión de los parámetros de posición de la cámara (parámetros extrínsecos).
    - GET\_IMAGE\_RESP\_CMUCAM (CMD=0xE1): Transmisión de los píxeles de la imagen almacenada en una cadena de caracteres del tamaño total de la imagen ( $Size = \frac{176 \times 143}{sample}$ , donde *sample* es el valor de muestreo de la imagen que se asigna junto con los parámetros de configuración y hace que se tomen los píxeles saltados. Si *sample* = 1, entonces se transmite toda la imagen, y si *sample* = 4, se transmite 1 píxel de cada 4).
    - SET\_PARAM\_RESP\_CMUCAM (CMD=0xF0): Respuesta de la cámara si todo es correcto al recibir el comando SET\_PARAM\_CMUCAM.
    - GET\_PARAM\_RESP\_CMUCAM (CMD=0xF1): Transmisión de los parámetros intrínsecos de la cámara.
    - ERROR\_CMUCAM3 (CMD=0x00): En caso de producirse algún error la cámara codifica el tipo de error y lo transmite a la aplicación del PC.

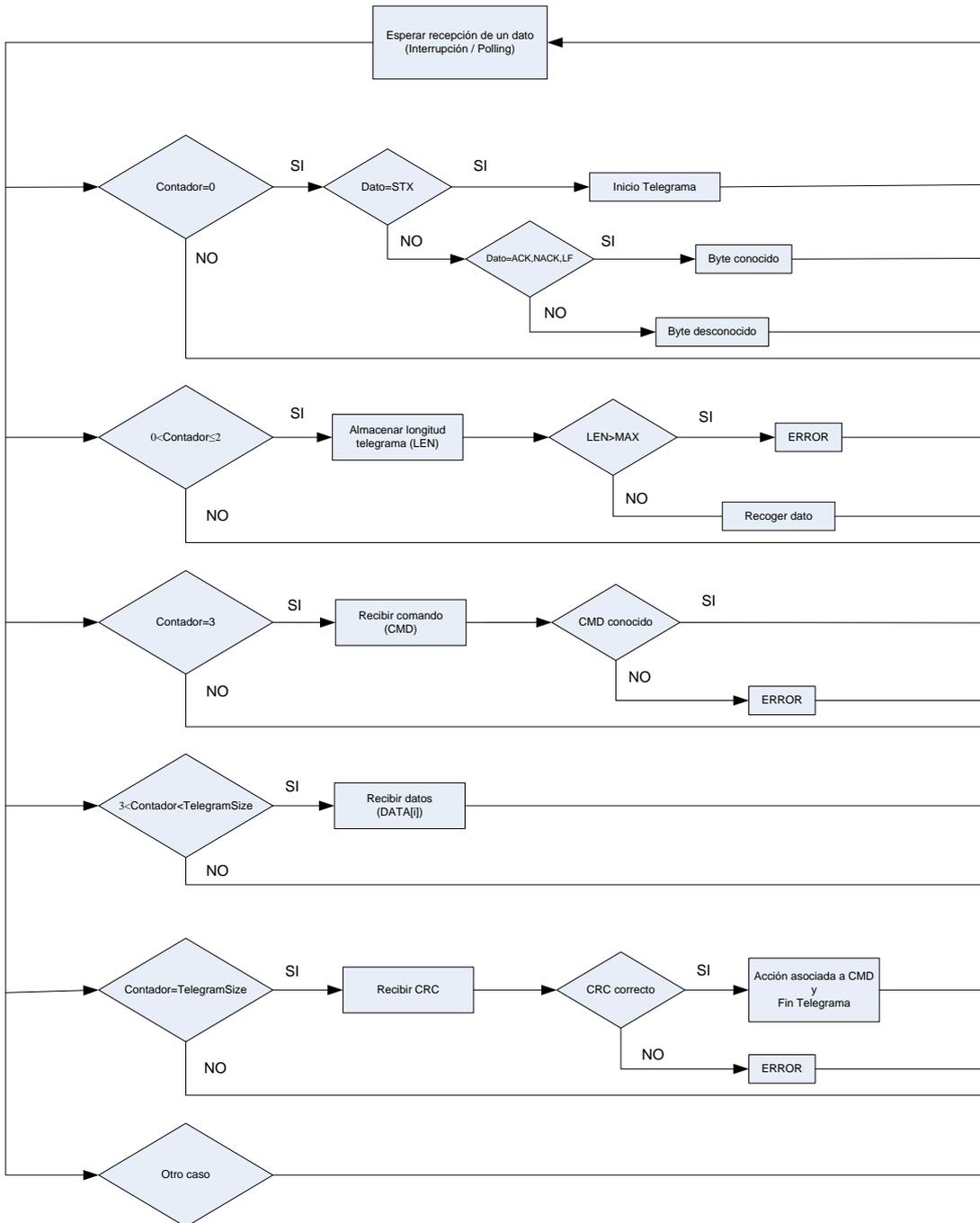


Figura 220. Proceso de decodificación del telegrama

El esquema de la Figura 220 representa detalladamente el diagrama de flujo del protocolo, en el que inicialmente hay una etapa de espera de recepción de datos. Según si el telegrama lo recibe la aplicación del PC o de la CMUcam3 se hace la espera de una forma u otra. Por lo tanto, el inicio del proceso de comunicación al recibir un byte puede ser por interrupción (en la aplicación del ordenador) o mediante *polling* (en la cámara). El resto del flujograma muestra cómo se va decodificando el mensaje y verificando los posibles errores para ejecutar la acción que vaya asociada a cada caso.

Hay que decir que de entre todos los comandos destacan los que intervienen directamente en el proceso de seguimiento, que son los de requerimiento de medida y el posicionamiento de los servos. Hay que decir que debido a las limitaciones del hardware se dispone de 2 tipos de medida, la diferencia entre ambos comandos radica en que en uno se envían únicamente los parámetros de la línea detectada (ángulo, distancia, punto inicial y punto final), mientras que en la otra opción además de eso se envían también los *inliers* detectados para su posterior visualización, lo cual ralentiza la comunicación. Es evidente que en el primer caso no se pueden visualizar los resultados obtenidos al no transmitirse los puntos *inliers*, pero el control es más eficaz puesto que el tiempo requerido en la comunicación es inferior y por tanto se puede realizar el control con un periodo de muestreo mucho menor. En cambio, en el segundo caso se puede ver gráficamente qué está haciendo el algoritmo, pero en consecuencia el control es más lento.

Una vez se han recibido en la aplicación del ordenador los datos de la recta detectada (parámetros, puntos inicial y final, *inliers*, etc.), éste se encarga de gestionar su visualización. Para ello se ha realizado un programa que es capaz de mostrar por la pantalla en tiempo real los puntos *inliers* de la recta, además de dibujar la recta detectada en la imagen y en su caso la imagen almacenada en la CMUcam3. En la Figura 221 se muestra un ejemplo de la imagen que aparece en la pantalla del ordenador cuando se ejecuta la aplicación.

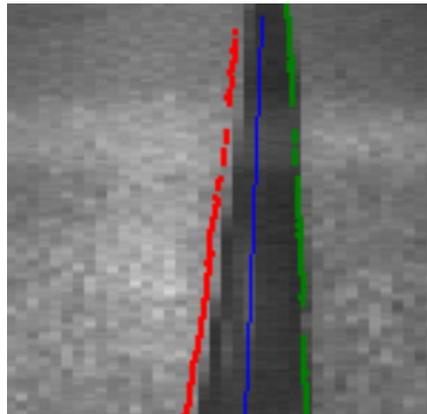


Figura 221. Aplicación del ordenador para visualización de la línea detectada



# **ANEXO 6.**

## **Artículos Publicados**

