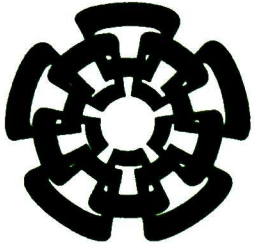




CT-67A-SS1

DOU.-2012

XX(199676.1)



Centro de Investigación y de Estudios Avanzados  
del Instituto Politécnico Nacional  
Unidad Guadalajara

# **Control Robusto de Mecanismos Robóticos Guiados por Visión**

Tesis que presenta:

**Luis Enrique González Jiménez**

para obtener el grado de:

**Doctor en Ciencias**  
en la especialidad de:

**Ingeniería Eléctrica**  
Directores de Tesis



**Dr. Eduardo José Bayro Corrochano**  
**Dr. Alexander Georgievich Loukianov**

CASE NO. CT 00583  
PROGRAM CT-678 SSI  
DATE 22-08-2012  
DOR 2012  
\$

ID: 199510-2001

# **Control Robusto de Mecanismos Robóticos Guiados por Visión**

**Tesis de Doctorado en Ciencias  
Ingeniería Eléctrica**

Por:

**Luis Enrique González Jiménez**

**Maestro en Ciencias en Ingeniería Eléctrica**

**Centro de Investigación y Estudios Avanzados del IPN, Unidad  
Guadalajara 2005-2007**

**Becario de CONACyT (Consejo Nacional de Ciencia y Tecnología),  
expediente no. 172488**

**Directores de Tesis**

**Dr. Eduardo José Bayro Corrochano**

**Dr. Alexander Georgievich Loukianov**

**CINVESTAV del IPN Unidad Guadalajara, Diciembre de 2011.**

# Resumen

La robótica ha adquirido un papel importante en la sociedad en los últimos años. Su utilidad va desde mejorar la eficiencia y seguridad en procesos de manufactura de una gran diversidad de productos, a ayudar a las personas en labores domésticas. Es por eso que los mecanismos robóticos son parte de nuestro entorno actual y su protagonismo promete crecer a pasos agigantados en un futuro cercano. A la par del crecimiento de la robótica, surge la necesidad de esquemas de control y modelado que permitan a un mecanismo robótico realizar su tarea con robustez y precisión. Los esquemas de control deben ser capaces de controlar un sistema altamente no lineal, con incertidumbre en el modelado y perturbaciones externas, propiedades inherentes de un robot en la vida real.

El modelado de los mecanismos robóticos utilizados en esta tesis se desarrolló bajo el esquema del álgebra geométrica conformal. Esta álgebra permite expresar entidades y transformaciones geométricas de una forma compacta y simple. La dinámica, cinemática directa, diferencial e inversa de un robot son ecuaciones obtenidas de las relaciones geométricas que poseen los elementos de dicho mecanismo, lo que perfila al álgebra geométrica conformal como la herramienta por excelencia para obtener este tipo de modelos.

Los modos deslizantes se han establecido como una excelente metodología en el control de plantas con las características antes mencionadas. Entre los algoritmos de modos deslizantes más efectivos se encuentran los modos deslizantes anidados, modos deslizantes integrales y el algoritmo *super-twisting*. En este trabajo se proponen tres controladores continuos y dos discretos basados en estos algoritmos.

Los controladores continuos se diseñan para el modelo dinámico de mecanismos robóticos, y los controladores discretos para su cinemática diferencial. En el primer controlador continuo se utilizan modos deslizantes anidados e integrales para atacar el problema de robustez a perturbaciones *no-matched* y asegurar la existencia de modo deslizante desde el instante inicial. En el segundo se combinan los modos deslizantes anidados con el algoritmo *super-twisting* para darle más suavidad a las señales de control en lazo cerrado. Y en tercer lugar se propone un controlador que

reúne las ventajas de los tres algoritmos para obtener una señal de control suave, con magnitud reducida y que sea robusta a perturbaciones *matched* y *no-matched*.

Por otra parte, los controladores discretos propuestos se basan en modos deslizantes integrales y el algoritmo super-twisting. El primero tiene como objetivo disminuir las ganancias de las señales de control y el segundo en obtener una ley de control continua que evite el efecto *chattering* discreto. Los análisis de estabilidad bajo el enfoque de Lyapunov se presentan para los algoritmos propuestos.

Finalmente, se muestran resultados en simulación para todos los controladores propuestos usando los modelos obtenidos por medio del álgebra geométrica conformal. Además, se muestra la implementación en tiempo real de los controles discretos, los cuales se apoyaron en un sistema de visión estéreo y en un algoritmo de segmentación por color para guiar el mecanismo robótico.

Dichos resultados muestran la eficacia de los esquemas planteados en el control de mecanismos robóticos guiados por visión.

# Summary

Robotics has acquired an important role in society in recent years. Its utility ranges from improving the efficiency and safety in manufacturing processes of a wide variety of products to help people in domestic work. That is why robotic mechanisms are part of our environment and its role promises to grow by leaps and bounds in the near future. Along with the growth of robotics, arises the need for modeling and control schemes that allow a robotic mechanism to perform its task with robustness and precision. Control schemes must be able to control a highly nonlinear system with modeling uncertainties and external disturbances, which are inherent properties of a robot in real life.

The modeling of robotic mechanisms used in this thesis was developed under the conformal geometric algebra framework. This algebra allows the representation of geometric entities and transformations in a compact and simple manner. Dynamics, direct kinematics, differential kinematics and inverse kinematics of a robot are obtained from the geometric relationships between its elements, which outlines the conformal geometric algebra as an excellent tool to obtain these models.

Sliding modes have been established as an excellent method to control systems with the aforementioned characteristics. Among the most effective sliding mode algorithms are the nested sliding modes, integral sliding mode and the super-twisting algorithm. In this paper we propose three continuous and two discrete controllers based on these algorithms.

The continuous controllers are designed for the dynamic model of robotic mechanisms, and the discrete controllers for the differential kinematic model. In the first continuous controller we use nested and integral sliding mode to address the problem of robustness to non-matched disturbances and to ensure the existence of sliding mode from the initial instance. In the second controller we combine nested sliding mode and super-twisting algorithm to obtain softer control signals in closed loop. And the third is a controller that combines the advantages of the three algorithms to obtain a smooth control signal, with reduced magnitude and robustness against matched and non-matched disturbances.



On the other hand, discrete controllers are proposed using integral sliding mode and the super-twisting algorithm. The first aims to reduce the magnitude of the control signals and the second to obtain a continuous control law that avoids the discrete chattering effect. The analysis of stability under the Lyapunov approach is presented for the proposed algorithms.

The simulation results for the proposed controllers, using the models obtained through the conformal geometric algebra approach, are presented. Moreover, real time implementation of discrete controls are shown, which were assisted by a stereo vision system and a color segmentation algorithm to guide the robotic mechanism. These results show the effectiveness of the proposed control schemes for the control of robotic mechanisms guided by vision.

# Agradecimientos

*Dedico esta tesis a:*

Mi esposa Ángel María, por su amor y apoyo incondicional que me ha permitido formar mi pequeña familia y lograr muchas otras metas, haciéndolo ver como una increíble aventura.

Mi hijo Luis Ángel, por el nuevo nivel de felicidad que me hizo conocer tan solo con su sonrisa.

Mis padres, Ernesto e Imelda, por ser los mejores padres que alguien pudiera desear tener. Por su apoyo, amor incondicional y el gran ejemplo que han sido en mi vida.

Mis hermanos, Alonso, Kenia y Yesenia, porque siempre han estado ahí y haber hecho de crecer junto a ellos un cúmulo de hermosos recuerdos.

Mi familia política, Rubén, Ana Isabel, Caro, Anita y Benrú, por su apoyo, cariño y por haberme brindado no solo su hogar, sino su amistad.

*Quiero agradecer a:*

Prof. Eduardo Bayro Corrochano y Prof. Alexander Georgievich Loukianov, por brindarme su tiempo, paciencia y conocimiento durante el desarrollo de esta tesis.

Dr. José Javier Ruiz León, Dr. Antonio Ramírez Treviño, Dr. Carlos López Franco y Dra. Nancy Arana Daniel, por su apoyo en la revisión de esta tesis.

Todos mis compañeros del CINVESTAV. Unidad Guadalajara, por su amistad y los grandiosos momentos compartidos.

CINVESTAV, Unidad Guadalajara por el apoyo brindado durante la realización de este trabajo.

CONACyT por el invaluable apoyo económico brindado durante el desarrollo del curso doctoral.



# Índice

<b>1. INTRODUCCIÓN .....</b>	<b>9</b>
<b>1.1. PLANTEAMIENTO DEL PROBLEMA .....</b>	<b>11</b>
<b>1.2. ALGORITMOS DE CONTROL.....</b>	<b>13</b>
<b>1.3. MODELADO MATEMÁTICO.....</b>	<b>15</b>
<b>1.4. VISIÓN COMPUTACIONAL.....</b>	<b>16</b>
<b>1.5. OBJETIVOS.....</b>	<b>17</b>
<b>1.6. ESTRUCTURA DEL DOCUMENTO .....</b>	<b>18</b>
<b>2. ANTECEDENTES.....</b>	<b>19</b>
<b>2.1. ÁLGEBRA GEOMÉTRICA .....</b>	<b>19</b>
2.1.1. EL PRODUCTO CLIFFORD DE VECTORES EN $\mathbb{R}^2$ .....	20
2.1.2. PRODUCTO EXTERIOR .....	21
2.1.3. ÁLGEBRA GEOMÉTRICA <i>N-DIMENSIONAL</i> .....	22
Propiedades de Multivectores .....	23
2.1.4. ÁLGEBRA GEOMÉTRICA CONFORMAL .....	24
Mapeo Conformal .....	25
Espacio de Minkowski $\mathbb{R}^{1,1}$ .....	27
Definición del Álgebra Geométrica Conformal .....	27
2.1.5. ENTIDADES GEOMÉTRICAS CONFORMALES.....	28
Esfera.....	28
Plano.....	30
Círculo .....	31
Línea .....	32
Par de Puntos .....	32
Punto .....	33
2.1.6. TRANSFORMACIONES CONFORMALES.....	34
Reflexión.....	35
Traslación .....	36
Rotación .....	36
2.1.7. ÁNGULOS ENTRE HIPERPLANOS.....	37
Ángulo entre Planos .....	38
Ángulo entre Líneas.....	38
2.1.8. MOVIMIENTOS RÍGIDOS: TRASLADORES, ROTORES Y MOTORES .....	38
Movimiento de Tornillo ( <i>screw motion</i> ).....	39
Propiedades de los Rotores.....	41

Propiedades de los Motores .....	42
<b>2.2. MODELO MATEMÁTICO CLÁSICO.....</b>	<b>43</b>
2.2.1. MODELO CINEMÁTICO.....	44
Cinemática Directa .....	44
Cinemática Inversa .....	45
Cinemática Diferencial .....	46
2.2.2. MODELO DINÁMICO .....	47
Ecuaciones de LaGrange .....	48
Ecuaciones de Newton-Euler .....	51
Dinámica Directa e Inversa.....	53
<b>2.3. FORMA NO-LINEAL CONTROLABLE POR BLOQUES.....</b>	<b>54</b>
<b>2.4. MODOS DESLIZANTES .....</b>	<b>55</b>
2.4.1. PRINCIPIOS DE DISEÑO DEL CONTROL POR MODOS DESLIZANTES.....	55
Control Equivalente.....	56
2.4.2. MODOS DESLIZANTES ANIDADOS (MDA).....	57
Principios de Diseño del Control por MDA.....	57
2.4.3. MODOS DESLIZANTES INTEGRALES (MDI) .....	59
Principios de Diseño de Control por MDI.....	59
2.4.4. ALGORITMO SUPER-TWISTING (ST) .....	61
<b>2.5. VISIÓN COMPUTACIONAL.....</b>	<b>62</b>
2.5.1. CÁMARA <i>PINHOLE</i> .....	62
2.5.2. SEGMENTACIÓN HSV .....	63
2.5.3. SISTEMA DE VISIÓN ESTÉREO .....	65
2.5.4. CALIBRACIÓN MANO-OJO .....	67
<b><u>3. PROPUESTA DE SOLUCIÓN .....</u></b>	<b><u>69</u></b>
<b>3.1. LOCALIZACIÓN DEL OBJETIVO.....</b>	<b>70</b>
3.1.1. CALIBRACIÓN DEL SISTEMA DE VISIÓN ESTÉREO (SVE) .....	70
3.1.2. SEGMENTACIÓN POR COLOR.....	71
3.1.3. TRIANGULACIÓN.....	72
<b>3.2. MODELADO PROPUESTO USANDO ÁLGEBRA GEOMÉTRICA.....</b>	<b>72</b>
3.2.1. MODELO DE CINEMÁTICA DIRECTA.....	72
3.2.2. MODELO DE CINEMÁTICA DIFERENCIAL .....	74
Modelo Cinemático Diferencial Discreto .....	77
3.2.3. MODELO DINÁMICO .....	77
3.2.4. CINEMÁTICA INVERSA.....	81
<b>3.3. SUPOSICIONES DE DISEÑO .....</b>	<b>84</b>
<b>3.4. CONTROL ROBUSTO DE ROBOTS POR MODOS DESLIZANTES.....</b>	<b>85</b>
<b>3.5. CONTROL POR MODOS DESLIZANTES INTEGRALES ANIDADOS (MDIA) .....</b>	<b>85</b>
3.5.1. ALGORITMO MDIA.....	86

Análisis de Estabilidad del controlador por MDIA .....	89
<b>3.6. CONTROL SUPER-TWISTING ANIDADO (STA) .....</b>	<b>91</b>
3.6.1. ALGORITMO STA.....	92
Análisis de Estabilidad del control STA.....	94
<b>3.7. CONTROL SUPER-TWISTING INTEGRAL ANIDADO (STIA).....</b>	<b>96</b>
3.7.1. ALGORITMO STIA.....	96
Análisis de Estabilidad del controlador por STIA.....	99
<b>3.8. CONTROL POR MODOS DESLIZANTES DISCRETOS (MDD).....</b>	<b>100</b>
<b>3.9. CONTROL POR MODOS DESLIZANTES INTEGRALES DISCRETOS (MDID) .....</b>	<b>101</b>
3.9.1. ALGORITMO DE MDID .....	101
Análisis de Estabilidad del control por MDID .....	103
<b>3.10. CONTROL SUPER-TWISTING DISCRETO (STD) .....</b>	<b>104</b>
3.10.1. ALGORITMO STD.....	104
<b><u>4. RESULTADOS EN SIMULACIÓN .....</u></b>	<b><u>107</u></b>
4.1. PARÁMETROS DEL MODELO DINÁMICO.....	107
4.2. CONTROL POR MDIA.....	108
4.3. CONTROL POR STA.....	110
4.4. CONTROL POR STIA.....	112
4.5. ANÁLISIS COMPARATIVO .....	113
4.6. PARÁMETROS DEL MODELO DISCRETO.....	114
4.7. CONTROL POR MDID.....	117
4.8. CONTROL STD.....	119
4.9. ANÁLISIS COMPARATIVO .....	121
<b><u>5. IMPLEMENTACIÓN EN TIEMPO REAL.....</u></b>	<b><u>123</u></b>
5.1. CONTROL POR MDID EN TIEMPO REAL.....	124
5.2. CONTROL STD EN TIEMPO REAL.....	125
5.3. ANÁLISIS COMPARATIVO .....	128
<b><u>6. CONCLUSIONES Y TRABAJO FUTURO.....</u></b>	<b><u>129</u></b>
6.1. CONCLUSIONES.....	129
6.2. TRABAJO FUTURO .....	131
<b><u>7. PUBLICACIONES .....</u></b>	<b><u>133</u></b>
<b><u>8. ANEXOS .....</u></b>	<b><u>135</u></b>

<b>8.1. TEOREMA DE ESTABILIDAD DE LYAPUNOV.....</b>	<b>135</b>
<b>8.2. DISCRETIZACIÓN DE EULER .....</b>	<b>137</b>
<b>8.3. GLOSARIO .....</b>	<b>138</b>
<b><u>BIBLIOGRAFÍA.....</u></b>	<b><u>141</u></b>

# Lista de Figuras

Figura 1.	Sistemas robóticos asistidos por visión computacional. ....	11
Figura 2.	Bivector representado como un plano orientado. ....	21
Figura 3.	Sentido del bivector. ....	21
Figura 4.	Proyección estereográfica de un punto en un círculo unitario. ....	25
Figura 5.	Vectores base y líneas nulas del plano de Minkowski. ....	27
Figura 6.	El modelo homogéneo para la proyección estereográfica en el caso 1D. ....	28
Figura 7.	Esfera construida con centro en $p_e$ y radio $\rho$ . ....	29
Figura 8.	Plano definido por 3 puntos que yacen en él y el punto en el infinito. ....	30
Figura 9.	Círculo formado por tres puntos o la intersección de dos esferas. ....	31
Figura 10.	Línea mapeada en la esfera unitaria. ....	32
Figura 11.	El par de puntos como la intersección de una línea y una esfera. ....	33
Figura 12.	Punto generado por su proyección estereográfica en la esfera unitaria. ....	34
Figura 13.	Reflexión de un punto $x$ respecto a un plano $\pi$ . ....	35
Figura 14.	Reflexión entre planos paralelos, la traslación. ....	36
Figura 15.	Reflexión entre planos no paralelos, la rotación. ....	37
Figura 16.	Uniones: Rotativa (izquierda) y Prismática (derecha). ....	43
Figura 17.	Manipulador serial de 6 grados de libertad. ....	44
Figura 18.	Ejes de rotación para un manipulador de 6 grados de libertad. ....	46
Figura 19.	Eslabón como un sistema de partículas y su centro de masas. ....	48
Figura 20.	Caracterización del $i^{\text{ésimo}}$ eslabón para las ecuaciones de Newton-Euler. ....	52
Figura 21.	Cámara <i>pinhole</i> ....	63
Figura 22.	Segmentación HSV de un objeto. ....	65
Figura 23.	Sistemas de Visión Estéreo. ....	65
Figura 24.	Triangulación: Ideal (Izquierda) y Real (Derecha). ....	66



Figura 25. Calibración Mano-Ojo.....	67
Figura 26. Manipulador Asistido por Visión y el Objetivo. ....	69
Figura 27. Calibración Mano-Ojo de un SVE.....	70
Figura 28. Segmentación en HSV para un SVE.....	71
Figura 29. Manipulador serial AdeptSix600 y sus ejes de rotación.....	74
Figura 30. PTU y su esquema geométrico. ....	81
Figura 31. Proyección del sistema robótico en el plano $\pi_1$ .....	83
Figura 32. Proyección del sistema robótico en el plano $\pi_2$ .....	83
Figura 33. Función sigmoidea para varios valores de pendiente. ....	84
Figura 34. Manipulador de 2 grados de libertad y sus parámetros.....	108
Figura 35. MDIA: Respuesta de seguimiento para la 1 <sup>era</sup> unión (izq.) y la 2 <sup>da</sup> (der.).....	109
Figura 36. MDIA: Variables de error (izq.) y retrato de fase de errores (der.).....	109
Figura 37. MDIA: Señales de control para la primera unión (izq.) y para segunda (der.).....	110
Figura 38. STA: Respuesta de seguimiento para la 1 <sup>era</sup> unión (izq.) y la 2 <sup>da</sup> (der.).....	110
Figura 39. STA: Variables de error (izq.) y retrato de fase de errores de seguimiento (der.).....	111
Figura 40. STA: Señales de control (izq.) y velocidades angulares (der.) de las uniones.....	111
Figura 41. STIA: Respuesta de seguimiento para la 1 <sup>ra</sup> unión (izq.) y la 2 <sup>da</sup> (der.).....	112
Figura 42. STIA: Variables de error (izq.) y retrato de fase de errores (der.).....	112
Figura 43. STIA: Señales de control (izq.) y superficies deslizantes (der.).....	113
Figura 44. Tabla comparativa entre los controladores MDIA, STA y STIA en simulación.....	114
Figura 45. Unidad <i>Pan-Tilt</i> con sus ejes de rotación (izq.) y su esquema geométrico (der.).....	114
Figura 46. Secuencia de imágenes para obtener la orientación de referencia. ....	116
Figura 47. Componentes euclidianas de la orientación de referencia obtenida.....	116
Figura 48. Orientación y referencia para MDID (izq.) y PIDD (der.).....	118
Figura 49. Variables de error para MDID (izq.) y PIDD (der.).....	118

Figura 50. Señales de control para MDID (izq.) y PIDD (der.) .....	119
Figura 51. STD: Componentes del vector de orientación $A_k$ y su referencia $A_{k,ref}$ (izq.) y las superficies deslizantes (der.).....	120
Figura 52. STD: Señales de control (izq.) y posiciones angulares (der.) .....	120
Figura 53. Tabla comparativa entre los controladores PIDD, MDID y STD en simulación.....	121
Figura 54. Sistemas de coordenadas relacionados por la calibración Mano-Ojo en el sistema robótico para MDID (izq.) y STD (der.).....	123
Figura 55. Posiciones angulares (izq.) y variables de error (der.) en tiempo real. ....	124
Figura 56. Superficies deslizantes (izq.) y señales de control (der.) en tiempo real. ....	125
Figura 57. Componentes euclidianas de la orientación y sus referencias en tiempo real. ....	125
Figura 58. Imágenes de la cámara izquierda del experimento en tiempo real para MDID.....	126
Figura 59. Señales de control (der.) y posiciones angulares (izq.) para STD en tiempo real.....	126
Figura 60. Orientación y referencia (izq.) y errores (der.) para STD en tiempo real. ....	127
Figura 61. Imágenes de la cámara izquierda del experimento en tiempo real para STD.....	127
Figura 62. Tabla comparativa entre los controladores MDID y STD en tiempo real. ....	128

## Lista de Símbolos

$q$ ó $\theta$	Vector de posiciones angulares
$w$ ó $\dot{\theta}$	Vector de velocidades angulares
$J(\bullet)$	Matriz Jacobiana
$\tau$	Esfuerzo de torsión
$\mathbb{R}^n$	Espacio n-dimensional de los reales
$\lim_{x \rightarrow y} z(x)$	Límite de $z(x)$ cuando $x$ tiende a $y$
$x_{\text{ref}}$	Señal de referencia para la variable $x$
$\ \bullet\ $	Norma-2 del término $\bullet$
$x_k$	Valor de la variable $x$ en la $k^{\text{ésima}}$ muestra
$T$	Período de muestreo.
$\text{sign}(\bullet)$	Función signo
$\text{sigm}(\bullet)$	Función sigmoidea, en este trabajo $\tanh(\bullet)$
$A^T$	Transpuesta de la matriz $A$
$A^{-1}$	Inversa de la matriz $A$
$A^+$	Pseudo-inversa de la matriz $A$
$\text{rank}(A)$	Rango de la matriz $A$

## Lista de Símbolos de Álgebra Geométrica

$x_e$	Punto expresado en el espacio euclidiano
$x$	Punto expresado en el espacio conformal
$M$	Motor
$L_x$	Línea $x$ expresada en $G_{4,1}$
$\pi_x$	Plano $x$ expresado en $G_{4,1}$
$R$	Rotor expresado en $G_{4,1}$
$G_{p,q,r}$	Álgebra geométrica con bases $p$ , $q$ y $r$
$\wedge$	Producto exterior en álgebra geométrica
$\sim$	Reversión
$T$	Traslador expresado en $G_{4,1}$
$e_0$	Punto en el origen del espacio conformal
$e_\infty$	Punto en el infinito del espacio conformal

# 1. Introducción

En las últimas décadas hemos presenciado avances sustanciales en el campo de la Robótica, en gran parte debido al desarrollo de *hardware* y *software* más eficientes como procesadores de alta velocidad, interfaces de fácil uso y acceso, sensores de mayor exactitud y precisión, entre otros. Esto ha permitido maximizar las capacidades de los dispositivos robóticos tales como precisión, exactitud y rendimiento; además de la posibilidad de realizar trabajos de alto riesgo como manejar residuos químicos peligrosos para el ser humano, acceder a lugares de dimensiones reducidas o ambientes hostiles, o en tareas de alta precisión como cirugías apoyadas por robots.

La necesidad de realizar estas actividades ha creado una especial relación entre la robótica y el control automático, permitiendo el crecimiento de aplicaciones de las técnicas de control automático en el campo de la robótica. Otra disciplina que también sostiene una relación directa con la robótica es la visión computacional. El uso de un sistema de visión por computadora permite al mecanismo robótico interactuar de una manera natural con su ambiente, debido a que emula la capacidad humana de observar una escena y tomar decisiones en base a la información obtenida en mediante dicha observación. Esta interacción es muy

importante para la mayoría de las tareas que usualmente realizan los mecanismos robóticos.

La complejidad de los dispositivos robóticos y su potencial de aplicación han sido detonantes en el crecimiento del interés de la comunidad científica por este campo, y en el número de trabajos que tratan sobre el control de robots, ya sea aplicando técnicas y algoritmos de control desarrollados en otros campos o creando algoritmos nuevos.

Entre los robots más empleados en la industria se encuentran los manipuladores robóticos, los cuales consisten de una secuencia de *eslabones* conectados por *uniones*, en las cuales puede existir un *actuador* (eléctrico, hidráulico, etc.) que permita el movimiento de los eslabones y el posicionamiento deseado de un elemento actuador final. No existe una definición única de robot manipulador. Quizás la definición más aceptada sea la de la International Federation of Robotics (IFR) [1], que dice:

*“Por robot industrial de manipulación se entiende a una máquina de manipulación automática, programable y multifuncional con tres o más ejes que pueden posicionar y orientar piezas, herramientas o dispositivos especiales para la realización de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento”*

Esta federación también ha establecido una clasificación de robots basada en la forma en que dicho mecanismo interactúa con su ambiente. Las clases son: robot secuencial, robot de trayectoria controlable, robot adaptativo y robot telemanipulado.

El uso de manipuladores robóticos en la industria permite automatizar muchas actividades normalmente realizadas por el hombre. Estas actividades dependen en gran medida de las características del manipulador a utilizar como: Grados de Libertad (GDL), número de actuadores, dimensiones de los eslabones y el tipo de elemento actuador final que pueda manejar.

Entre las tareas que puede llevar a cabo un manipulador se encuentran las domésticas como cortar el césped, aspirar el polvo de una habitación, de ornamento; las de aplicaciones industriales como la colocación de partes en un proceso de ensamblado, en la soldadura y el pintado de partes automotrices,

empacado de productos terminados, entarimado o apilado de objetos, el corte de piezas base, entre otros.

## 1.1. Planteamiento del Problema

Considere un mecanismo robótico que debe realizar una tarea asistido por un sistema de visión estéreo (SVE). Dicho sistema puede tener varias configuraciones como las mostradas en la Figura 1.

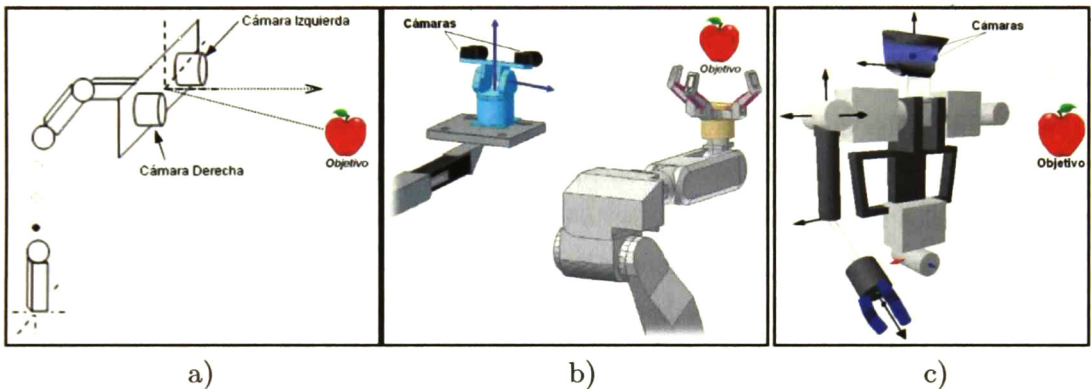


Figura 1. Sistemas robóticos asistidos por visión computacional.

En la Figura 1-a se aprecia una configuración donde el SVE se encuentra montado en el último eslabón de un robot de  $n$  grados de libertad, y la tarea a realizar es el *seguimiento visual de un objetivo*. En la Figura 1-b se aprecia otra configuración, la cual consta de un SVE montado en una base móvil y manipuladores robóticos. El SVE se encuentra empotrado sobre uno de los robots, el cual determina la orientación del mismo, y la tarea es ver y manipular el objetivo al mismo tiempo. La Figura 1-c muestra un robot humanoide, el cual posee una configuración similar a la del centro. Cuenta con un SVE localizado en la cabeza del robot, un manipulador relacionado con la *cabeza* y otro manipulador que representa un *brazo*. Aquí la tarea es idéntica a la configuración anterior, sin embargo, la transformación entre las bases de los manipuladores es constante para este caso.

Todas las configuraciones descritas tienen algo en común: la tarea global puede ser dividida en dos partes. Primero: el sistema debe localizar el objetivo en la

escena y extraer la información que defina la pose del mismo. Segundo: el sistema debe reconfigurar la estructura de los manipuladores para obtener una pose (posición y orientación) deseada en sus efectores finales. En el caso 1-a, la pose deseada para el manipulador del sistema se obtiene de tal forma que la orientación del SVE esté alineada con el vector de posición del objetivo, para mantener a éste dentro del espacio de visión del SVE. Para los casos 1-b y 1-c, la pose deseada para el otro manipulador se define enteramente por la pose del objetivo, y por cómo se desea que el robot interactúe con el mismo. En otras palabras, para la segunda parte el problema requiere de un algoritmo de control cuyas entradas sean obtenidas por el procesamiento de las imágenes, y su salida sean las poses de los efectores finales de los manipuladores.

El diseñar un algoritmo de control demanda un análisis de las características físicas de los mecanismos robóticos; lo que nos lleva a la necesidad de obtener un modelo matemático de los mismos, para obtener así, una estrategia de control que nos permita lograr el correcto movimiento de los efectores finales del sistema.

Sin embargo, en la mayoría de los casos, obtener un modelo exacto de mecanismos robóticos es una tarea muy complicada o imposible. Esto debido a factores como:

Dinámicas No Lineales, causadas por las fuerzas centrífugas, de Coriolis y gravitacionales; las cuales son muy difíciles de calcular.

Variaciones Paramétricas Internas, provocadas por el calentamiento de los actuadores, desgaste en las uniones.

Perturbaciones Externas, como ruido en los sensores, cambio de masa en los objetos manipulados o interacciones físicas (como golpes o empujones) con elementos del ambiente de trabajo.

Dinámicas no modeladas, debido a la flexibilidad de los eslabones, que suelen considerarse totalmente rígidos.

Por lo tanto, el algoritmo de control deberá ser robusto ante los factores mencionados y el algoritmo de visión, el cual lleve a cabo la primera parte de la tarea, deberá ser robusto a cambios de iluminación, de escala y orientación del objetivo; de tal forma que siempre localice al objetivo en la escena.

## 1.2. Algoritmos de Control

Una gran variedad de técnicas y estrategias de control se han utilizado en la robótica, las cuales deben poseer características como: alta robustez, sensibilidad al ambiente de trabajo, y bajo costo computacional y energético. Además, el algoritmo de control debe implementarse en tiempo real por lo que deberá ser diseñado en tiempo discreto, o diseñarse en tiempo continuo y aplicarse en forma digital. Esto debido a la forma que opera una computadora, la cual otorga la facilidad de poder programar los algoritmos fácil y rápidamente.

El problema más común en el control de robots es el seguimiento de trayectorias deseadas para la pose del efector final y/o de sus eslabones. Este problema ha sido estudiado extensamente y un importante número de metodologías se han utilizado para resolverlo. Entre ellas se encuentran las siguientes:

### **Linealización por retroalimentación de estado**

En este caso los modelos no lineales del sistema se linealizan mediante un lazo de retroalimentación, de tal forma que se pueden utilizar técnicas de control para sistemas lineales. Esta estrategia de control puede encontrarse en forma detallada en [2], al igual que las tres estrategias siguientes.

### **PID**

Este controlador está conformado por términos proporcional, integral y derivativo. El diseño de las ganancias de dichos términos permite manipular la respuesta en lazo cerrado del sistema tanto en tiempo como en frecuencia. Cabe mencionar que el uso de este tipo de control es muy popular en la industria [3].

### **Pasividad**

Esta técnica aprovecha la característica de un sistema pasivo, el cual siempre consume energía. Entonces, se utilizan funciones de energía para el diseño y análisis de sistemas de este tipo. La pasividad es una característica inherente al modelado Lagrangiano de los manipuladores [4].

### **Funciones de Lyapunov**

En este caso el controlador está diseñado en base del método directo de Lyapunov, el cual define funciones tipo energía para diseñar un controlador y al mismo tiempo demostrar la estabilidad del sistema en lazo cerrado. La ventaja de



esta técnica es que la región de estabilidad que proporciona el controlador es mayor que la obtenida utilizando técnicas lineales [5].

### **Control de Estructura Variable (VSC) *Variable Structure Control***

En esta técnica la estructura del controlador cambia de acuerdo a algún criterio para obtener un comportamiento deseado de la planta o sistema a controlar. El criterio por el cual cambia la estructura del controlador está basado en funciones de las variables de estado escogidas por el diseñador. Dentro del VSC se encuentran los *Modos Deslizantes*, cuya principal característica es la robustez que presenta a perturbaciones externas [6; 7].

### **Momento de fuerza calculado**

Esta técnica consiste en el cálculo de los momentos de fuerza necesarios para cancelar las dinámicas no-lineales naturales de la estructura mecánica del manipulador, por lo tanto requiere de un conocimiento exacto del modelo del manipulador, la cual es su principal desventaja [8].

### **Control inteligente**

En este tipo de controladores se distinguen básicamente 2 tipos: los que se basan en redes neuronales y los que utilizan control difuso. Básicamente las redes neuronales son un sistema que se entrena para emular una dinámica deseada, ya sea del controlador, del modelo del sistema a identificar, o ambas [9].

En el control difuso se obtienen varios sub-modelos, cuya estructura ofrece alguna ventaja sobre el modelo inicial (como linealidad) y se diseña un controlador no lineal en base a este conjunto de sub-modelos como en [10].

### **Control adaptivo**

La idea de este esquema de control es el cambiar las ganancias del controlador en función del cambio de los parámetros del modelo del sistema a controlar, como en [11; 12], o se puede utilizar el esquema proponiendo un modelo lineal nominal y un lazo de control extra que se adapta a las no-linealidades de la planta que se vayan presentando con el transcurso del tiempo como en [13].

Dentro de las estrategias antes mencionadas, el Control por Modos Deslizantes (CMD) es una técnica de control que se ha utilizado ampliamente debido a su *baja sensibilidad a perturbaciones* y a *variaciones paramétricas*, ya que es común que

procesos industriales consistan de plantas operando bajo condiciones de incertidumbre y ruido. Además, el diseño de la ley de control es relativamente simple lo cual reduce el costo computacional de su implementación en tiempo real.

Entre las técnicas de control basadas en CMD, y que se utilizaron en la realización de este trabajo, se encuentran: modos deslizantes anidados, modos deslizantes integrales y modos deslizantes de alto orden. La combinación de estas técnicas nos permitirá diseñar estrategias de control que nos permitan abordar y solucionar el problema de seguimiento de trayectorias en mecanismos robóticos.

### 1.3. Modelado Matemático

El diseño de controladores requiere, casi siempre, el obtener previamente una representación matemática, ó modelo, de la planta a controlar. En nuestro caso, la planta es un mecanismo robótico y existen varios modelos que se pueden obtener de la misma, así como diversas formas de hacerlo. Dos de los principales modelos que se pueden obtener de un robot son su cinemática y su dinámica. El método Denavit-Hartenberg [14] es uno de los más usados para obtener modelos cinemáticos, y para modelar dinámica suelen usarse el de Euler-LaGrange ó Newton-Euler. [14] Sin embargo, estos han sido desarrollados basados en el álgebra euclidiana con representación matricial, lo que genera una alta complejidad en el modelo y en su obtención.

Otra opción para modelar mecanismos robóticos es el uso de cuaterniones, los cuales son un *anillo de división* sobre los reales y surgieron como una extensión de los números complejos. Se utilizan para modelar rotaciones en 3D [15] y, por lo tanto, son una opción para el modelado y control de mecanismos robóticos [16]. Las desventajas que poseen los cuaterniones son: solo pueden representar rotaciones, para implementar traslaciones se necesita extender los cuaterniones a cuaterniones duales, su multiplicación no es una operación simple, su parametrización es muy compleja y a veces la representación cuaterniónica pierde su significado físico [17].

Esto nos motivó a utilizar **álgebra geométrica conformal**, el cual es otro marco teórico que nos permite obtener o replantear los modelos antes descritos de una manera simple, compacta e intuitiva. Este álgebra también permite definir las leyes

de control diseñadas usualmente en algebra euclidiana, haciendo posible usarlo tanto en la obtención del modelo como en el diseño del controlador [18].

## 1.4. Visión Computacional

Otro aspecto importante, en cuanto a control de robots se refiere, es el de percibir el ambiente con el cual el robot interactuará. Para esto, se utilizan sensores que monitoreen en tiempo real dicho ambiente. Un sistema de visión es uno de los sensores que ofrece muchas ventajas al respecto, ya que permite percibir propiedades importantes del ambiente de trabajo de un robot cuando realiza una tarea. Estas propiedades van desde presencia y movimiento, hasta color, forma, textura y geometría, lo que permite al robot interactuar de mejor forma con el ambiente al realizar sus tareas. Debido a lo expuesto anteriormente, las estrategias de control desarrolladas en este trabajo son asistidas por un sistema de visión.

Una de las ventajas de utilizar un sistema de visión, es la capacidad única que posee de percibir y procesar su ambiente. Información como color, forma, tamaño, estructura y posición relativa, pueden ser obtenidas por medio de un sistema de visión (SV); lo que le da la versatilidad necesaria para ser utilizado en infinidad de aplicaciones en robótica y otros campos. El uso de un sistema de visión estéreo (SVE) provee la ventaja de información en 3D; lo que es fundamental en tareas como agarre y manipulación de objetos, en complemento con el seguimiento visual.

Entre los usos más comunes que se le han dado a un sistema de visión podemos encontrar:

- Seguridad y vigilancia,
- Identificación de patrones,
- Clasificación de objetos,
- Manipulación y seguimiento de objetos, [19; 20]
- Auto-localización de robots,
- Navegación de robots móviles, entre otros.

Los usos anteriormente mencionados, procesan las imágenes obtenidas por el SV para localizar un objeto de interés en la escena de la cual se tomaron las imágenes.

Las técnicas más comunes usadas para procesar dichas imágenes pueden ser clasificadas en: estrategias basadas en modelos, estrategias basadas en características locales o globales y estrategias basadas en histogramas.

Los algoritmos basados en modelos utilizan un modelo pre-obtenido de un objeto, usualmente compuesto de líneas. Los puntos de interés de este modelo son proyectados en el plano de la imagen, y cuando las correspondencias entre los puntos de interés y el plano de la imagen son obtenidas, el objeto es localizado también [21].

Los métodos basados en características globales segmentan una región de la imagen, la cual contiene al objeto de interés, delimitada dependiendo de las características del objeto como color, textura, tamaño o movimiento. Entre los algoritmos usados para detectar dichas características se encuentran: detector de esquinas de Harris [22], el algoritmo de Shi-Tomasi [23] y el algoritmo SIFT (del inglés *Scalar Invariant Feature Transform*). [24]

Los métodos basados en histogramas, utilizan un histograma obtenido de una imagen del objeto aislado, y luego es comparado con una imagen de referencia en la escena a analizar. Los histogramas usualmente son obtenidos a partir de imágenes de color [25; 26], o información espacial del objeto [27].

## 1.5. Objetivos

Los objetivos principales de esta tesis se definen como sigue:

- Utilizar el álgebra geométrica conformal para obtener el modelo dinámico y cinemático de un cabezal robótico de 2 GDL y de dos brazos de 5 GDL, los cuales conforman al humanoide MexOne del laboratorio de Control Automático del CINVESTAV, Unidad Gdl.
- Diseñar un controlador robusto basado en Modos Deslizantes Integrales para los mecanismos robóticos modelados.
- Diseñar un controlador robusto basado en Modos Deslizantes de Alto Orden para los mecanismos robóticos.
- Desarrollar los análisis de estabilidad para los controladores propuestos en lazo cerrado, utilizando el enfoque de Lyapunov.

- Desarrollar un algoritmo de segmentación por color para el seguimiento de objetos por medio de un sistema de visión estéreo.
- Desarrollar un esquema de control de robots guiados por visión que incluya lo logrado en los objetivos anteriores e implementarla en tiempo real.

## 1.6. Estructura del Documento

El documento está organizado de la siguiente manera: el *capítulo 2* define el estado del arte de los tópicos relacionados con el trabajo presentado, la propuesta de solución para el problema planteado se encuentra en el *capítulo 3*, en el *capítulo 4* se muestran los resultados experimentales de los controladores propuestos, y, finalmente, en el *capítulo 5* se expresan las conclusiones y el trabajo futuro de lo presentado en esta tesis.

## 2. Antecedentes

En este capítulo se presentarán los fundamentos matemáticos de tópicos requeridos en el desarrollo del trabajo aquí presentado.

### 2.1. Álgebra Geométrica

En 1844, el matemático alemán, Hermann Gunther Grassmann [1809-1877], publicó en su “Teoría de la extensión lineal, una nueva rama de las matemáticas” (del alemán *Die Lineale Ausdehnungslehre: ein neuer Zweig der Mathematik*), la primera edición de su cálculo geométrico que ofrecía una manera original de describir operaciones geométricas. En este trabajo se presenta una extensión del álgebra euclidiana, apareciendo el *producto exterior* como el principal producto geométrico de esta teoría. Con esta operación, el producto de vectores generaba áreas, volúmenes y objetos de altas dimensiones.

El álgebra geométrica, también conocida como *álgebra de Clifford* (William K. Clifford [1845-1879]), es una herramienta que mediante una serie de operaciones y propiedades extiende el sistema de números reales y complejos hacia el cálculo vectorial incluyendo el productora exterior [28]. En la década de 1960, esta álgebra es reformulada por David Hestenes planteando las álgebras geométricas; las cuales

son una herramienta para unificar los conceptos de geometría con el álgebra y la representación geométrica, y facilitar su uso en física e ingeniería [29].

Una de las principales características del álgebra de Clifford (o álgebra geométrica), es que nos permite representar entidades de orden mayor con una simbología compacta, y poder operarlas de forma lineal. Las líneas, planos o esferas son ejemplos de entidades de orden mayor y son representadas como únicos elementos del álgebra de Clifford.

### 2.1.1. El Producto Clifford de Vectores en $\mathbb{R}^2$

Sería útil tener una multiplicación de vectores que satisfagan los mismos axiomas de multiplicación de los números reales (distributividad, asociatividad y conmutatividad) y que la norma de estos vectores se preserve en dicha multiplicación  $|ab| = |a||b|$ , donde  $a$  y  $b$  son vectores.

A esta multiplicación de vectores la llamamos *producto Clifford*. Pero como es imposible mantener una conmutatividad en dimensiones de  $n \geq 3$ , nos enfocaremos en la distributividad y asociatividad.

Tomando dos vectores con base ortonormal  $e_1 = [1,0]^T$  y  $e_2 = [0,1]^T$  en el espacio vectorial  $\mathbb{R}^2$ . La magnitud del vector  $r = xe_1 + ye_2$  está dada por  $|r| = \sqrt{x^2 + y^2}$ . Si el vector  $r$  es elevado al cuadrado utilizando el producto Clifford, la expresión sería  $rr = r^2 = |r|^2$

Usando la forma de coordenadas, introduciremos el producto de vectores de tal forma que

$$(xe_1 + ye_2)^2 = x^2 + y^2$$

Utilizando la regla distributiva sin asumir la conmutabilidad obtenemos

$$x^2e_1^2 + y^2e_2^2 + xy(e_1e_2 + e_2e_1) = x^2 + y^2$$

con esto podemos definir el producto de vectores (el producto Clifford).

**Definición 2.1.** Sean  $a = a_1e_1 + a_2e_2, b = b_1e_1 + b_2e_2 \in \mathbb{R}^2$  dos vectores con una base ortonormal  $e_1, e_2$ , la multiplicación utilizando el producto Clifford de estos vectores queda de la siguiente forma:

$$ab = a_1b_1 + a_2b_2 + (a_1b_2 - a_2b_1)e_1e_2$$

Esto se satisface si los vectores de la base obedecen las reglas de multiplicación

$$e_1^2 = e_2^2 = 1, \quad e_1 e_2 = -e_2 e_1 \text{ o equivalentemente } |e_1| = |e_2| = 1, \quad e_1 \perp e_2.$$

Usamos la propiedad asociativa para calcular

$$(e_1 e_2)^2 = (e_1 e_2)(e_1 e_2) = e_1 (e_2 e_1) e_2 = -e_1^2 e_2^2 = -1.$$

Como el cuadrado del elemento  $e_1 e_2$  es negativo, se sigue que no es un escalar ni un vector, sino una nueva clase de unidad llamado *bivector*, representando el área del plano orientado (ver Figura 2) del cuadrado con los lados  $e_1$  y  $e_2$ . Escribiremos  $e_{12} = e_1 e_2$ .

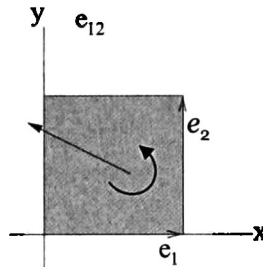


Figura 2. Bivector representado como un plano orientado.

### 2.1.2. Producto Exterior

Analizando el producto de los vectores  $a = a_1 e_1 + a_2 e_2$  y  $b = b_1 e_1 + b_2 e_2$ , podemos identificar dos partes

$$a \cdot b = a_1 b_1 + a_2 b_2 \Rightarrow \text{El producto escalar 'a punto b', y} \quad (2.1.1)$$

$$a \wedge b = (a_1 b_2 - a_2 b_1) e_{12} \Rightarrow \text{El producto exterior 'a wedge b'} \quad (2.1.2)$$

El bivector  $a \wedge b$  representa el segmento del plano orientado del paralelogramo con lados  $a$  y  $b$ . El área de este paralelogramo es  $|a_1 b_2 - a_2 b_1|$  y también representa la magnitud del bivector.

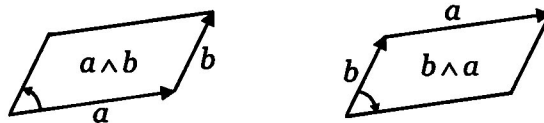


Figura 3. Sentido del bivector.

El bivector  $a \wedge b$  y  $b \wedge a$  tienen la misma magnitud pero con sentidos de rotación opuestos (ver Figura 3). Esto puede ser expresado simplemente escribiendo



$$a \wedge b = -b \wedge a \quad (2.1.3)$$

Notamos que el producto Clifford de los dos vectores es la suma de su parte escalar  $(a_1b_1 + a_2b_2)$  y bivectorial  $(a_1b_2 - a_2b_1)$ . El producto Clifford, así como el producto exterior y producto interior, puede ser extendido a mayores dimensiones.

### 2.1.3. Álgebra Geométrica *n*-dimensional

Un espacio vectorial *n*-dimensional puede ser expandido usando las bases ortonormales de sus vectores  $\{e_i\}$ ,  $i=1, \dots, n$  tal que  $e_i e_j = e_{ij}$ . Esto nos lleva a una base que expande un espacio vectorial lineal que corresponde al álgebra geométrica  $G_n$ . Los elementos generados por dicha base son

$$1, \{e_i\}, \{e_i \wedge e_j\}, \{e_i \wedge e_j \wedge e_k\}, \dots, I \quad (2.1.4)$$

donde  $I = e_1 \wedge e_2 \wedge \dots \wedge e_n$ .

En general un álgebra geométrica  $G_{p,q,r}$  ( $p, q, r \in \mathbb{N}$ ) es un espacio lineal de dimensión  $2^n$  con  $n = p + q + r$  con sub-espacios a los que llamaremos blades, y donde  $p, q$  y  $r$  corresponden al número de vectores base que cuadran a  $+1, -1$  y  $0$ , respectivamente. La notación  $G_{p,q,r}(\mathbb{R}_{p,q,r})$  es usada para determinar de cual espacio vectorial provienen los elementos del álgebra.

Para aclarar lo anterior, sean  $e_i, e_j \in G_{p,q,r}$  dos vectores de la base ortonormal de un espacio vectorial. Entonces, el producto geométrico de los vectores de la base nos da la base del álgebra geométrica, la cual se define como

$$e_i e_j = \begin{cases} 1 & \text{para } i = j \in \{1, \dots, p\} \\ -1 & \text{para } i = j \in \{p+1, \dots, q\} \\ 0 & \text{para } i = j \in \{q+1, \dots, r\} \\ e_{ij} = -e_{ji} & \text{para } i \neq j \end{cases} \quad (2.1.5)$$

El producto geométrico (o Clifford) de dos vectores se puede representar de la siguiente forma

$$ab = a \cdot b + a \wedge b. \quad (2.1.6)$$

La regla conmutativa  $a \cdot b = b \cdot a$ , junto con la regla anti-conmutativa  $a \wedge b = -b \wedge a$ , nos permite inferir

$$ba = a \cdot b - a \wedge b \quad (2.1.7)$$

Sumando y restando las ecuaciones (2.1.6) y (2.1.7), podemos definir

$$\begin{aligned} \mathbf{a} \cdot \mathbf{b} &= \frac{1}{2}(\mathbf{ab} + \mathbf{ba}) \\ \mathbf{a} \wedge \mathbf{b} &= \frac{1}{2}(\mathbf{ab} - \mathbf{ba}) \end{aligned} \quad (2.1.8)$$

El producto exterior puede ser extendido a dimensiones mayores, por ejemplo el producto  $(\mathbf{a} \wedge \mathbf{b}) \wedge \mathbf{c}$  es un *trivector*, y en general el *producto exterior* de  $k$  vectores es un  $k$ -vector o *k-blade*; dicha cantidad decimos que tiene grado  $k$ . El producto interior de 1-blade con un  $k$ -blade se puede generalizar por medio de

$$\mathbf{a} \cdot \mathbf{B}_k = \frac{1}{2}(\mathbf{aB}_k + (-1)^{k+1} \mathbf{B}_k \mathbf{a}). \quad (2.1.9)$$

Un *multivector* es una combinación lineal de vectores (o blades) de distintos grados, y se dice que es un multivector homogéneo si todos los blades que lo componen son del mismo grado.

Cada multivector  $\mathbf{M} \in \mathbf{G}_n$  puede ser escrito en su forma expandida como

$$\mathbf{M} = \sum_{k=0}^n \langle \mathbf{M} \rangle_k \quad (2.1.10)$$

donde  $\langle \mathbf{M} \rangle_k$  es la parte de  $\mathbf{M}$  que tiene grado  $k$ . Note que  $\langle \mathbf{M} \rangle_0 \in \mathbb{R}$ .

## Propiedades de Multivectores

Un elemento  $\mathbf{M} \in \mathbf{G}_n$  es invertible si existe otro elemento  $\mathbf{N} \in \mathbf{G}_n$  tal que  $\mathbf{MN} = \mathbf{NM} = 1$ , donde  $\mathbf{N}$  (si existe) es único, es llamado el inverso de  $\mathbf{M}$  y se denota por  $\mathbf{M}^{-1}$ . Una condición para la existencia del inverso de  $\mathbf{M}$  es que este sea no nulo i.e.  $\mathbf{M}^2 \neq 0$ .

Por asociatividad y multi-linealidad, el producto exterior se extiende a cualquier numero finito de multivectores y a escalares, con la siguiente propiedad

$$\lambda \wedge \mathbf{M} = \mathbf{M} \wedge \lambda = \lambda \mathbf{M}, \quad \text{para } \lambda \in \mathbb{R}, \mathbf{M} \in \mathbf{G}_n \quad (2.1.11)$$

y para el producto interior tenemos

$$\lambda \cdot \mathbf{M} = \mathbf{M} \cdot \lambda = 0, \quad \text{para } \lambda \in \mathbb{R}, \mathbf{M} \in \mathbf{G}_n. \quad (2.1.12)$$

Para cualquier blade  $\mathbf{A}$  y  $\mathbf{B}$  con grados  $r$  y  $s$  diferente de cero, tenemos

$$\mathbf{A} \cdot \mathbf{B} = \langle \mathbf{AB} \rangle_{|r-s|} \quad (2.1.13)$$

$$\mathbf{A} \wedge \mathbf{B} = \langle \mathbf{AB} \rangle_{r+s} \quad (2.1.14)$$

El *blade* de más alto grado es el  $n$ -*blade*  $I = e_1 \wedge \cdots \wedge e_n$  y es llamado *pseudoescalar*. Un álgebra geométrica se llama *no degenerada* si la magnitud del pseudoescalar es diferente de cero. Los pseudoescalares son indexados de acuerdo al álgebra a la que pertenecen, por ejemplo  $I_E \in G_3$  (álgebra euclidiana) y  $I_C \in G_{4,1}$  (álgebra conformal).

El dual de un multivector  $M \in G_n$  es denotado por

$$M^* = MI_n^{-1} \quad (2.1.15)$$

y por lo tanto el dual de un  $r$ -*blade* es un  $(n-r)$ -*blade*.

La reversión de un  $s$ -*blade*  $A = a_1 \wedge \cdots \wedge a_s$  se denota como

$$\tilde{A} = (a_1 \wedge \cdots \wedge a_s)^\sim = a_s \wedge a_{s-1} \wedge \cdots \wedge a_2 \wedge a_1 \quad (2.1.16)$$

y generalizando para cualquier multivector se obtiene

$$\langle \widetilde{M} \rangle_i = (-1)^{i(i-1)/2} \langle M \rangle_i, \text{ para } M \in G_n, 0 \leq i \leq n. \quad (2.1.17)$$

Dado  $D = ABC$ , donde  $A, B, C, D \in G_n$ , su reversión está dada por

$$\tilde{D} = \widetilde{ABC} = \tilde{C}\tilde{B}\tilde{A}. \quad (2.1.18)$$

La involución principal de  $G_n$  (también llamada *involución de grado*) es definida como

$$\langle \widehat{M} \rangle_i = (-1)^i \langle M \rangle_i. \quad (2.1.19)$$

La función exponencial de un multivector  $M$  puede ser expresada en series de expansión como

$$e^M = \sum_{k=0}^{\infty} \frac{M^k}{k!}. \quad (2.1.20)$$

## 2.1.4. Álgebra Geométrica Conformal

El contar con un álgebra que involucre a varias entidades geométricas (puntos, líneas, planos, esferas, círculos, etc.) permite el operar dichas entidades y sus relaciones geométricas de manera simple y compacta. El *álgebra geométrica conformal* posee dichas características y se relaciona con el espacio Euclidiano mediante una proyección estereográfica. Utilizaremos el álgebra geométrica  $G_{4,1}$  para trabajar con la geometría conformal y mostraremos como el espacio  $\mathbb{R}^3$  es representado en un espacio  $\mathbb{R}^{4,1}$ , donde los superíndices denotan una métrica pseudo-euclidiana. Este espacio está definido por las bases:  $e_i$  (vectores),  $e_{ij}$

(bivectores) con  $i, j = \{1, \dots, 5\}$ ; de las cuales  $e_{23}, e_{31}$  y  $e_{12}$  corresponden a las bases de los *cuaterniones de Hamilton*.

## Mapeo Conformal

Este resulta de hacer una proyección estereográfica y una homogenización de las coordenadas. Tomemos el ejemplo de mapear el espacio euclidiano  $\mathbb{R}^1$  con base  $e_1$  a  $G_{2,1}$  con bases  $p = \{e_1, e_+\}$  y  $q = \{e_-\}$ .

Sea  $x_e \in \mathbb{R}^1$  entonces su representación conformal correspondiente  $x \in G_{2,1}$  (ver la Figura 4) se obtiene como la intersección de la línea  $f(x) = -\frac{x}{x_e} + 1$  con el círculo unitario  $f(x)^2 + x^2 = 1$ , de donde obtenemos

$$x^2 - 2xx_e + x^2x_e^2 = 0. \quad (2.1.21)$$

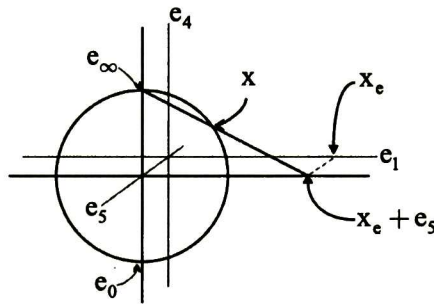


Figura 4. Proyección estereográfica de un punto en un círculo unitario.

Esta ecuación tiene 2 soluciones

$$x = 0, \quad x = 2 \frac{x_e}{x_e^2 + 1}. \quad (2.1.22)$$

si evaluamos en la solución no trivial obtenemos el punto de intersección como

$$P = \left( \frac{2x_e}{x_e^2 + 1}, \frac{x_e^2 - 1}{x_e^2 + 1} \right), \quad (2.1.23)$$

y agregando la coordenada homogénea ( $e_-$ ) tenemos

$$x = \frac{2x_e}{x_e^2 + 1} e_1 + \frac{x_e^2 - 1}{x_e^2 + 1} e_+ + e_- \quad (2.1.24)$$

para obtener, finalmente, la *representación conformal* de  $x_e$ . De (2.1.24) se pueden obtener las representaciones conformales de los *puntos en el infinito* y el *origen*.

Para el punto en el infinito, denotado por  $e_\infty$ , se aplica el límite  $x_e \rightarrow \infty$  en (2.1.24) de donde obtenemos

$$\begin{aligned}
 e_\infty &= \lim_{x_e \rightarrow \infty} (x), \\
 &= \lim_{x_e \rightarrow \infty} \left( \frac{2x_e}{x_e^2 + 1} e_1 + \frac{x_e^2 - 1}{x_e^2 + 1} e_+ + e_- \right), \\
 &= \lim_{x_e \rightarrow \infty} \left( \frac{1}{x_e} e_1 + e_+ + e_- \right), \\
 &= e_+ + e_-.
 \end{aligned} \tag{2.1.25}$$

Para el origen, denotado por  $e_0$ , aplicamos el límite  $x_e \rightarrow 0$  para obtener

$$\begin{aligned}
 e_0 &= \lim_{x_e \rightarrow 0} (x), \\
 &= \lim_{x_e \rightarrow 0} \left( \frac{2x_e}{x_e^2 + 1} e_1 + \frac{x_e^2 - 1}{x_e^2 + 1} e_+ + e_- \right), \\
 &= -e_+ + e_-.
 \end{aligned} \tag{2.1.26}$$

Los productos del punto en el infinito y el origen se definen como  $e_\infty \cdot e_0 = -2$ ,  $e_\infty \wedge e_0 = 2e_+ \wedge e_-$ . Con fines de simplificación algebraica, estos productos se normalizan a

$$\begin{aligned}
 e_\infty \cdot e_0 &= -1 \\
 e_\infty \wedge e_0 &= e_+ \wedge e_-
 \end{aligned} \tag{2.1.27}$$

donde  $e_0 = (-e_+ + e_-)/2$ . Note que se redefinió a  $e_0$ , pero esto es válido ya que es un vector homogéneo, y por lo tanto  $e_0 = ke_0$ ,  $k \in \mathbb{R}$ . Si definimos los vectores nulos como

$$e_- = \frac{1}{2} e_\infty + e_0, \quad e_+ = \frac{1}{2} e_\infty - e_0, \tag{2.1.28}$$

se puede representar a la transformación conformal usando estos vectores nulos con la forma

$$\begin{aligned}
 x &= \frac{2x_e}{x_e^2 + 1} + \frac{x_e^2 - 1}{x_e^2 + 1} \left( \frac{1}{2} e_\infty - e_0 \right) + \left( \frac{1}{2} e_\infty + e_0 \right) \\
 x &= \frac{2x_e}{x_e^2 + 1} + \frac{x_e^2 e_\infty}{x_e^2 + 1} + \frac{2e_0}{x_e^2 + 1}
 \end{aligned} \tag{2.1.29}$$

para finalmente obtener

$$x = x_e + \frac{1}{2} x_e^2 e_\infty + e_0. \tag{2.1.30}$$

## Espacio de Minkowski $\mathbb{R}^{1,1}$

Este espacio es generado por la *base ortonormal*  $\{e_+, e_-\}$  o la *base nula*  $\{e_\infty, e_0\}$ . El pseudoescalar unitario se define como

$$E = e_0 \wedge e_\infty = e_+ \wedge e_- \quad (2.1.31)$$

y tiene las siguientes propiedades

$$\begin{aligned} E^2 = 1, \quad \tilde{E} = -E, \quad E e_\infty = -e_\infty, \quad e_0 E = e_0, \quad e_+ E = e_-, \\ e_- E = e_+, \quad e_+ e_\infty = 1 + E, \quad e_- e_\infty = -1 - E, \quad e_\infty \wedge e_- = E, \quad e_+ \cdot e_\infty = 1. \end{aligned} \quad (2.1.32)$$

La base vectorial y líneas nulas de  $G_{1,1}$  son mostradas en la Figura 5, donde el área sombreada representa el pseudo-escalar unitario  $E$ .

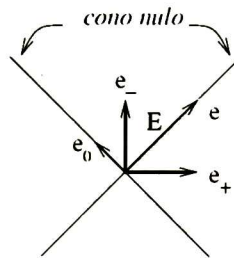


Figura 5. Vectores base y líneas nulas del plano de Minkowski.

## Definición del Álgebra Geométrica Conformal

El rol que juega el plano de Minkowski de generar vectores nulos y expandir el espacio vectorial Euclidiano  $\mathbb{R}^n$  a  $\mathbb{R}^{n+1,1}$ , resulta en el álgebra geométrica conformal  $G_{n+1,1}$ . El conjunto  $\mathbb{N}^{n+1}$  de todos los vectores nulos en  $\mathbb{R}^n$  es llamado *cono nulo*, y el conjunto de todos los puntos generalizados homogéneos de grado superior es llamado *horósfera*. En la Figura 6 se muestra la visualización del modelo homogéneo para la proyección estereográfica para el caso de 1D. Todos los puntos estereográficos proyectados yacen en el cono, que es el cono nulo relacionado con el plano de Minkowski. También se notan el *hiperplano*  $P(e_\infty, e_0)$  y la horósfera del álgebra conformal.

Un vector en este espacio puede ser representado como una descomposición ortogonal

$$a = P_E(a) + P_E^\perp(a) \quad (2.1.33)$$

donde los operadores de proyección  $P_E$  y  $P_E^\perp$  están dados por

$$\begin{aligned} P_E(a) &= (a \cdot E)E = \alpha e_0 + \beta e_\infty \in \mathbb{R}^{1,1} \\ P_E^\perp(a) &= (a \cdot E^*)\tilde{E}^* = (a \wedge E)E = a_e \in \mathbb{R}^n \end{aligned} \quad (2.1.34)$$

El espacio vectorial conformal derivado de  $\mathbb{R}^3$  es denotado por  $\mathbb{R}^{4,1}$ . Su base vectorial está dada por  $\{e_1, e_2, e_3, e_+, e_-\}$  ó por  $\{e_1, e_2, e_3, e_0, e_\infty\}$ . El álgebra correspondiente  $G_{4,1}$  contiene  $2^5 = 32$  elementos. El pseudo-escalar unitario se denota por  $I_c = e_{123+-} = e_{123\infty 0} = I_E E$ , donde  $I_E$  es el pseudo-escalar de  $G_3$ .

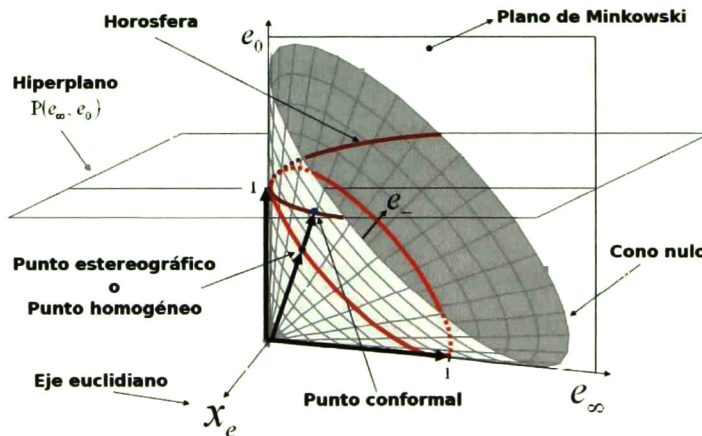


Figura 6. El modelo homogéneo para la proyección estereográfica en el caso 1D.

### 2.1.5. Entidades Geométricas Conformales

Una de las grandes ventajas del *álgebra geométrica conformal* es la de representar de manera simple y compacta entidades geométricas, así como ofrecer las herramientas algebraicas necesarias para obtener relaciones geométricas entre ellas. En las siguientes secciones se describirán las principales entidades geométricas conformales y como se generan.

#### Esfera

La esfera es la entidad geométrica de cálculo básica para esta álgebra, ya que por medio de ella se pueden generar todas las demás entidades.

Existen muchas formas de construir una esfera: definiendo su centro y radio, con cuatro puntos que yacen en ella, definiendo su círculo ecuatorial, sus dos polos,

entre otras. Se obtendrá la *ecuación conformal* de la esfera a partir de la *ecuación euclidiana* de la misma con centro en  $p_e$  y radio  $\rho$  como se muestra en la Figura 7.

$$(x_e - p_e)^2 = \rho^2 \quad (2.1.35)$$

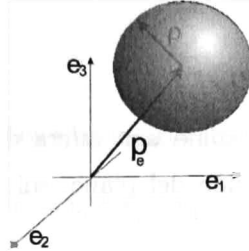


Figura 7. Esfera construida con centro en  $p_e$  y radio  $\rho$ .

Desarrollando obtenemos

$$\begin{aligned} x_e^2 - 2x_e \cdot p_e + p_e^2 &= \rho^2 \\ \frac{x_e^2}{2} - x_e \cdot p_e + \frac{p_e^2 - \rho^2}{2} &= 0 \quad (2.1.36) \\ \underbrace{\left( x_e + \frac{x_e^2}{2} e_\infty + e_0 \right)}_x \cdot \underbrace{\left( p_e + \frac{p_e^2 - \rho^2}{2} e_\infty + e_0 \right)}_s &= 0 \end{aligned}$$

de donde se obtiene la ecuación conformal de la esfera como

$$s = p_e + \frac{p_e^2 - \rho^2}{2} e_\infty + e_0. \quad (2.1.37)$$

Se puede observar que es un *1-vector* o vector de grado 1, mientras que la relación  $x \cdot s = 0$  determina cuando un punto yace sobre la esfera. Simplificando (2.1.37) obtenemos

$$s = p - \frac{1}{2} \rho^2 e_\infty \quad (2.1.38)$$

y las siguientes relaciones

$$\rho^2 = s \cdot s, \quad p = s + \frac{1}{2} (s \cdot s) e_\infty \quad (2.1.39)$$

También es posible definir la esfera con cuatro puntos que yacen en ella; esto se logra mediante el producto exterior o *wedge* de dichos puntos para definir la representación dual de la esfera como

$$s^i = x_1 \wedge x_2 \wedge x_3 \wedge x_4. \quad (2.1.40)$$



Cualquier otro punto  $x_i$  que yace sobre la esfera se puede definir en términos de los cuatro que la definieron, por lo que se obtiene la relación

$$s \wedge x_i = 0 \quad (2.1.41)$$

la cual nos ofrece otra forma de corroborar si un punto  $x_i$  yace sobre una esfera  $s$ .

## Plano

Un plano se puede imaginar como una esfera de radio infinito (ver Figura 8), por lo que, para obtener la ecuación del plano, solo es necesario intercambiar uno de los puntos de la esfera por  $e_\infty$ .

$$\pi^* = x_1 \wedge x_2 \wedge x_3 \wedge e_\infty \quad (2.1.42)$$

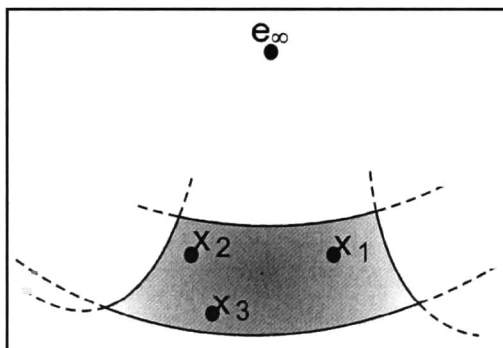


Figura 8. Plano definido por 3 puntos que yacen en él y el punto en el infinito.

Definiendo  $x_{1e} \wedge x_{2e} = -x_{1e} \wedge (x_{1e} - x_{2e})$  se puede re-escribir la ecuación anterior como

$$x_2 \wedge x_1 \wedge x_2 \wedge e_\infty = x_{3e} \wedge x_{1e} \wedge x_{2e} \wedge e_\infty + (x_{2e} - x_{1e}) \wedge (x_{3e} - x_{1e}) E \quad (2.1.43)$$

de donde podemos obtener

$$\pi^* = d I_E e_\infty + N E \quad (2.1.44)$$

donde  $d$  es la *distancia de Hesse* (distancia más corta del origen  $e_0$  al plano  $\pi$ ) y se define como

$$d = \frac{(x_{1e} \wedge x_{2e} \wedge x_{3e})}{|(x_{2e} - x_{1e}) \wedge (x_{3e} - x_{1e})|}, \quad (2.1.45)$$

y  $N = (x_{2e} - x_{1e}) \wedge (x_{3e} - x_{1e}) / |(x_{2e} - x_{1e}) \wedge (x_{3e} - x_{1e})|$  es el bivector unitario normal a  $\pi$

De (2.1.44) se puede obtener la representación estándar del plano como

$$\begin{aligned}\pi &= \pi^* I_c \\ &= de_\infty + n\end{aligned}\quad (2.1.46)$$

donde  $n = NI_E = Ne_{123}$  es el vector de grado uno normal a  $\pi$ .

## Círculo

El círculo se puede obtener por tres puntos no colineales que yacen en él

$$z^* = x_1 \wedge x_2 \wedge x_3, \quad (2.1.47)$$

ó por la intersección de dos esferas

$$z = s_1 \wedge s_2 \quad (2.1.48)$$

como se muestra en la Figura 9. Esto es realmente muy práctico, ya que el círculo es simplemente un *trivector* o vector de grado 3 en álgebra conformal, en comparación con las 3 ecuaciones paramétricas necesarias en álgebra euclidiana para representar un círculo en  $\mathbb{R}^3$

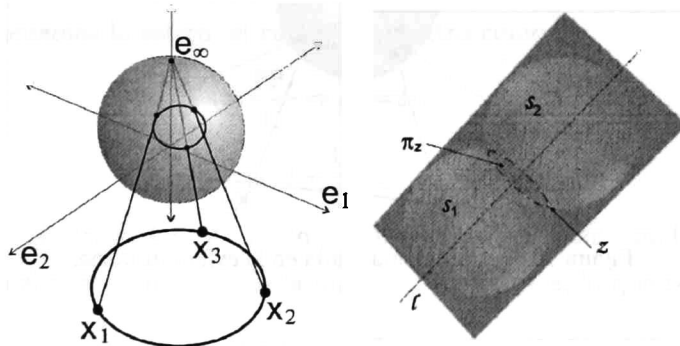


Figura 9. Círculo formado por tres puntos o la intersección de dos esferas.

Similar al caso de la esfera, cualquier punto  $x$  que yace sobre el círculo cumplirá la siguiente relación

$$x \cdot z = 0 \quad (2.1.49)$$

Se puede calcular el plano donde un círculo yace de la siguiente forma

$$\pi_z^* = z^* \wedge e_\infty, \quad (2.1.50)$$

y la línea del eje del círculo como

$$l^* = z \wedge e_\infty, \quad (2.1.51)$$

por lo tanto, la intersección de los elementos anteriores nos permite definir el centro del círculo de la siguiente manera

$$c = (l \cap \pi_z) = \Gamma^* \cdot \pi_z. \quad (2.1.52)$$

## Línea

Así como el plano se puede imaginar como una esfera de radio infinito, la línea puede conceptualizarse como un círculo de radio infinito (véase Figura 10). De esta forma, y utilizando (2.1.42), obtenemos

$$\begin{aligned} \vec{L} &= x_1 \wedge x_2 \wedge e_\infty \\ &= x_{1e} \wedge x_{2e} \wedge e_\infty + (x_{2e} - x_{1e})E \\ &= e_\infty m + nE \end{aligned} \quad (2.1.53)$$

donde  $n$  es el vector de dirección de la línea y  $m$  su momento.

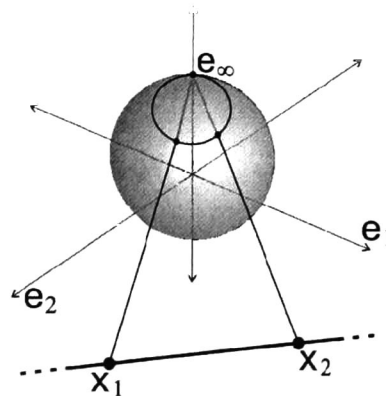


Figura 10. Línea mapeada en la esfera unitaria.

Esta entidad también puede ser generada usando un punto  $x_1$  que yace sobre ella y la dirección de la misma como

$$\vec{L} = x_1 \wedge n \wedge e_\infty. \quad (2.1.54)$$

## Par de Puntos

Si tomamos la parte bivectorial del producto de dos puntos,  $a$  y  $b$ , obtenemos una entidad conocida como *par de puntos*. Dicha entidad posee información sobre la localización de los 2 puntos, y está definida como

$$PP^* = a \wedge b. \quad (2.1.55)$$

Otra forma de obtener el par de puntos es como la intersección de una línea con una esfera (ver Figura 11), lo que resulta en

$$PP^* = sL^* \quad (2.1.56)$$

Esta expresión alternativa ofrece la ventaja de poder extraer uno de los puntos que conforman el par de puntos utilizando el centro de la esfera  $c$  y trasladándolo ambos lados una distancia igual al radio de la esfera  $\rho$ .

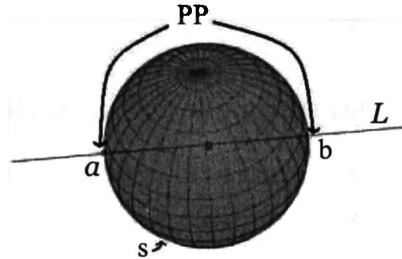


Figura 11. El par de puntos como la intersección de una línea y una esfera.

De (2.1.56) obtenemos la esfera, el radio y su centro como

$$s = \frac{PP^*}{L^*} \Rightarrow \rho^2 = s \cdot s, \quad (2.1.57)$$

$$c = s + \frac{1}{2} \rho e_\infty,$$

y construimos el *trasladador*  $d$  (dicha transformación conformal se verá más adelante) como el vector unitario de la dirección de la línea, lo que resulta en

$$T = 1 + \frac{1}{2} r d e_\infty, \quad (2.1.58)$$

y así recuperar los puntos como

$$\begin{aligned} p_1 &= TcT^{-1}, \\ p_2 &= T^{-1}cT. \end{aligned} \quad (2.1.59)$$

## Punto

Esta es la entidad geométrica más básica de todas y está en dualidad con la entidad de mayor grado del álgebra que es la esfera. De hecho, el punto puede pensarse como una esfera con radio cero.

Estrictamente hablando, los puntos conformales son obtenidos mediante la proyección estereográfica de los puntos euclidianos sobre la esfera de radio unitario con centro en el origen (ver Figura 12) y se definen como

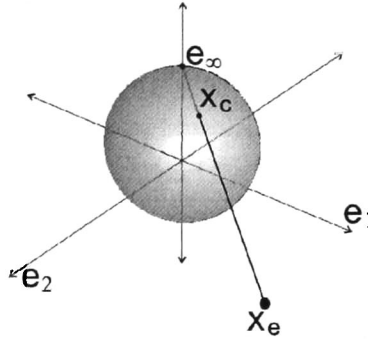


Figura 12. Punto generado por su proyección estereográfica en la esfera unitaria.

$$x = x_e + \frac{1}{2} x_e^2 e_\infty + e_0. \quad (2.1.60)$$

En esta álgebra todos los puntos son elementos del cono nulo, y poseen las siguientes propiedades

$$\{x \in G_{n+1,1} \mid x^2 = 0, x \cdot e_\infty = -1\} \quad (2.1.61)$$

Existe otra forma de representar el mismo punto, la cual se conoce como la forma dual, y consiste en un cuatrivector generado por la intersección de 4 esferas.

$$s' = x_1 \wedge x_2 \wedge x_3 \wedge x_4 \Leftrightarrow x' = s_1 \wedge s_2 \wedge s_3 \wedge s_4 \quad (2.1.62)$$

### 2.1.6. Transformaciones Conformales

Las transformaciones de grupo son parte de las herramientas que ofrece un álgebra para establecer relaciones, isomorfismos y equivalencias entre las entidades definidas en ella. Cualquier transformación conformal puede ser expresada como una composición de *reflexiones* en hiperplanos. En general, una transformación conformal tiene la forma

$$g(x) = Vx\widehat{V}^{-1} = kx'V \quad (2.1.63)$$

donde  $x \in G_{n+1,1}$ ,  $V$  es un versor y  $k$  un escalar. Como el *cono nulo* es invariante bajo  $V$  (i.e.  $(x')^2 = x^2 = 0$ ) es conveniente aplicar un escalamiento  $k$  para asegurar que  $x' \cdot e_\infty = x \cdot e_\infty = -1$

Un *versor* es cualquier multivector que puede expresarse como el producto geométrico de vectores invertibles. Los versores en  $G_n$  forman un grupo llamado *grupo de versores*, y los de magnitud unitaria un subgrupo llamado *grupo alfiler o perno* (pin group).

Un *versor par* es aquel que cumple  $V = \widehat{V}$  y es llamado *spinor* o *rotor* si  $V\widehat{V} = |\mathbf{V}|^2$  y un *versor impar* cumple  $V = -\widehat{V}$

A continuación se explicarán a detalle algunas transformaciones conformales basadas en versores.

## Reflexión

La reflexión de un punto  $x$  respecto a un plano  $\pi$  es igual al mismo punto menos dos veces la distancia dirigida del plano al punto (ver Figura 13), esto es  $x' = x - 2(\pi \cdot x)\pi^{-1}$

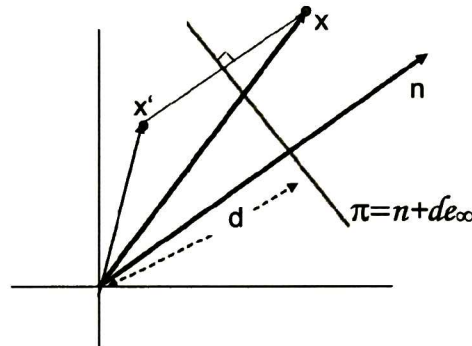


Figura 13. Reflexión de un punto  $x$  respecto a un plano  $\pi$

Para simplificar esta expresión, recordemos la propiedad  $ab + ba = 2(b \cdot a)$ , entonces podemos reescribir a la reflexión como

$$\begin{aligned} x' &= x - (\pi x - x \pi) \pi^{-1} \\ &= x - \pi x \pi^{-1} - x \pi \pi^{-1} \\ &= -\pi x \pi^{-1} \end{aligned} \tag{2.1.64}$$

En general, para cualquier entidad geométrica  $Q$ , la reflexión con respecto al plano  $\pi$  está dada por  $Q' = \pi x \pi^{-1}$

## Traslación

Esta se puede conceptualizar como dos reflexiones consecutivas respecto a dos planos paralelos (ver Figura 14).

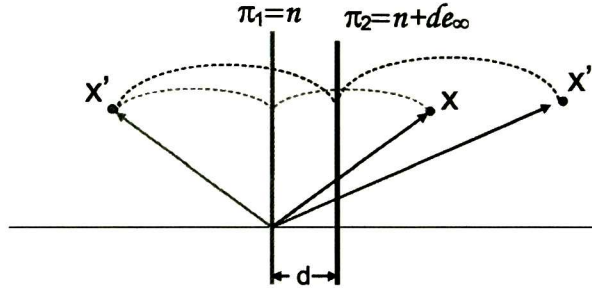


Figura 14. Reflexión entre planos paralelos, la traslación.

Sin pérdida de generalidad, se puede considerar que los planos están normalizados (normal de magnitud uno) y donde uno de ellos pasa por el origen. Entonces, la traslación se representa como

$$Q' = \underbrace{(\pi_2 \pi_1)}_{\tilde{T}_a} Q \underbrace{(\pi_1^{-1} \pi_2^{-1})}_{\tilde{T}_a} \quad (2.1.65)$$

con

$$\begin{aligned} \tilde{T}_a &= \pi_2 \pi_1, \\ &= (n + d e_\infty)(n + 0 e_\infty) \\ &= 1 + \frac{1}{2} e_\infty a = e^{e_\infty a/2} \end{aligned} \quad (2.1.66)$$

donde  $a = 2dn$  y  $\|n\| = 1$ . La distancia de traslación es dos veces la distancia entre los planos.

Los trasladadores cuya dirección son los vectores euclidianos base  $\{e_1, e_2, e_3\}$  se obtienen como

$$T_{e_1} = e^{-r e_1 e_\infty/2}, \quad T_{e_2} = e^{-r e_2 e_\infty/2}, \quad T_{e_3} = e^{-r e_3 e_\infty/2}, \quad (2.1.67)$$

donde  $r$  es la distancia de traslación.

## Rotación

Se puede describir una rotación por medio de dos reflexiones consecutivas a través de dos planos  $\pi_1$  y  $\pi_2$  que se cruzan en el punto  $c_e$  (ver Figura 15).

Supongamos que el punto en común es el origen, entonces los planos tienen la forma  $\pi = n$  y la rotación está dada por

$$Q' = \underbrace{(\pi_2 \pi_1)}_{R_\theta} Q \underbrace{(\pi_1^{-1} \pi_2^{-1})}_{\tilde{R}_\theta} \quad (2.1.68)$$

donde

$$R_\theta = \pi_2 \pi_1 = n_2 \cdot n_1 + n_2 \wedge n_1 \quad (2.1.69)$$

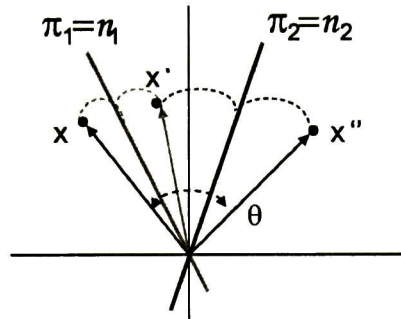


Figura 15. Reflexión entre planos no paralelos, la rotación.

Recordando las fórmulas de ángulos entre dos vectores

$$a \cdot b = |a||b|\cos(\alpha), \quad a \wedge b = |a||b|\sin(\alpha), \quad (2.1.70)$$

es posible re-escribir el rotor de la siguiente manera

$$R_\theta = \cos\left(\frac{\theta}{2}\right) - \sin\left(\frac{\theta}{2}\right)l = e^{-\frac{\theta}{2}l}, \quad (2.1.71)$$

donde  $l = n_1 \wedge n_2$  y  $\theta = 2\alpha$  (dos veces el ángulo entre los planos).

Los rotores cuya dirección son los vectores euclidianos base  $\{e_1, e_2, e_3\}$  se obtienen como

$$R_{e_1} = e^{-\theta e_{23}/2}, \quad R_{e_2} = e^{-\theta e_{31}/2}, \quad R_{e_3} = e^{-\theta e_{12}/2}, \quad (2.1.72)$$

donde  $\theta$  es el ángulo de rotación.

### 2.1.7. Ángulos entre Hiperplanos

Una de las relaciones más importante y útil entre entidades conformales, principalmente planos y líneas, es el ángulo que forman un par de dichas entidades. A continuación se abordará la forma de calcularlo para el caso de par de planos y par de líneas.



## Ángulo entre Planos

Sean  $\pi_a = a + d_a e_\infty$  y  $\pi_b = b + d_b e_\infty$  dos planos con normas  $a, b$  respectivamente, entonces

$$\begin{aligned}\pi_a \pi_b &= (a + d_a e_\infty)(b + d_b e_\infty) = a \cdot b, \\ |\pi| &= \sqrt[3]{\pi \cdot \pi} = \sqrt[3]{n \cdot n} = |n|.\end{aligned}\tag{2.1.73}$$

Ahora, de la ecuación de ángulo entre vectores

$$a \cdot b = |a||b|\cos(\theta)\tag{2.1.74}$$

y sustituyendo en términos de planos se tiene

$$\pi_a \cdot \pi_b = |\pi_a||\pi_b|\cos(\theta).\tag{2.1.75}$$

## Ángulo entre Líneas

Sean las líneas  $L_a^* = e_\infty m_a + n_a E$  y  $L_b^* = e_\infty m_b + n_b E$  con direcciones  $n_a, n_b$  respectivamente, y que se intersecan en un punto  $c$ . Obtenemos el producto interno de las líneas como

$$\begin{aligned}L_a^* &= a \wedge c \wedge e_\infty = a_e \wedge c_e \wedge e_\infty + (a_e - c_e)E, \\ L_b^* &= b \wedge c \wedge e_\infty = b_e \wedge c_e \wedge e_\infty + (b_e - c_e)E, \\ L_a^* \cdot L_b^* &= n_a \cdot n_b + a_e \cdot b_e e_\infty - a_e \cdot b_e e_\infty = n_a \cdot n_b.\end{aligned}\tag{2.1.76}$$

La norma de una línea se calcula como

$$\begin{aligned}|L^*| &= \sqrt[3]{L^* \cdot L^*} \\ &= \sqrt[3]{n \cdot n} \\ &= |n|\end{aligned}\tag{2.1.77}$$

Finalmente, obtenemos una expresión similar a la obtenida para los planos

$$L_a^* \cdot L_b^* = |L_a^*||L_b^*|\cos(\theta).\tag{2.1.78}$$

### 2.1.8. Movimientos Rígidos: Trasladores, Rotores y Motores

Por definición, un *cuerpo rígido* es aquel que no se deforma. El movimiento de este tipo de cuerpos comprende a las rotaciones, traslaciones y combinaciones de las anteriores. Además, en este tipo de movimientos siempre es posible definir un sistema de coordenadas, respecto al centro de masas del cuerpo, cuyo origen tiene

velocidad instantánea cero i.e. el cuerpo se mueve como si solo rotara respecto al origen del sistema de coordenadas.

Las transformaciones conformales vistas con anterioridad, nos permiten representar movimientos de cuerpos rígidos. Por ejemplo, los bivectores del álgebra geométrica permiten representar rotaciones de entidades conformales en 3D.

En  $G_{4,1}$  una rotación puede ser expresada por un rotor en su forma exponencial como  $R = e^{-\theta/2}$  y una traslación por el trasladador  $T = e^{dne_\infty}$ . Para expresar un movimiento rígido, se utiliza la aplicación consecutiva de un rotor y un trasladador y se define como el producto de ellos de forma

$$M = TR . \quad (2.1.79)$$

Esta transformación es un versor de grado par llamado *motor*, y se aplica a cualquier entidad conformal (puntos, líneas, planos, esferas, etc.) de la misma forma que los rotores, i.e. para una entidad  $Q$  su transformación resulta en

$$\begin{aligned} Q' &= MQ\tilde{M} \\ &= (TR)Q(\tilde{R}\tilde{T}) \end{aligned} \quad (2.1.80)$$

El conjunto de todos los motores forma otra álgebra, llamada *álgebra de motores*  $G_{3,0,1}$  [30], la cual es isométrica al álgebra de *cuaterniones duales*.

### Movimiento de Tornillo (*screw motion*)

Un movimiento de tornillo es una rotación sobre una línea (en general, que no pase por el origen) combinada con una traslación a lo largo de dicha línea. Usando rotores y trasladadores es posible definir este tipo de movimientos de la siguiente manera.

Para modelar la rotación de un punto  $x$  alrededor de una línea arbitraria  $L$ , se traslada  $x$  con el vector de distancia entre  $L$  y el origen, para luego realizar la rotación en el origen y finalmente trasladar de regreso el punto transformado. Entonces, el motor  $M \in G_{4,1}$  describe una rotación general que tiene la forma

$$M = TR\tilde{T} = R_L \quad (2.1.81)$$

donde se aprecia la secuencia de traslación, rotación y traslación de regreso (*reversión*), respectivamente. A  $R_L$  se le puede ver como un rotor sobre la línea  $L$  y al aplicarlo sobre un punto resulta en

$$x' = R_L x \widetilde{R}_L = TR \widetilde{T} x TR \widetilde{T} \quad (2.1.82)$$

Usando la forma exponencial del rotor y trasladador obtenemos

$$\begin{aligned} R_L &= TR \widetilde{T} \\ &= e^{e_\infty a/2} e^{-\theta/2} e^{-e_\infty a/2} \\ &= \left(1 + \frac{e_\infty a}{2}\right) e^{-\theta/2} \left(1 - \frac{e_\infty a}{2}\right), \\ &= e^{\left(1 + \frac{e_\infty a}{2}\right) (-\theta/2) \left(1 - \frac{e_\infty a}{2}\right)} \\ &= e^{\frac{\theta}{2} (1 + e_\infty (a \cdot 1))} \end{aligned} \quad (2.1.83)$$

la cual es la fórmula de *rotación general* [31]. El uso de la forma exponencial de rotores y trasladadores permite operarlos y derivarlos de una manera más simple.

Se puede demostrar que la parte exponencial de  $R_L$  coincide con los componentes de la línea sobre la que se efectúa la rotación. Para lograr lo anterior, usamos  $L^* = e_\infty m + 1E$  y definimos el momento  $m = (a \wedge 1)$  con  $a$  como la traslación ortogonal del origen a la línea, entonces la línea la expresamos como

$$L^* = e_\infty (a \wedge 1) + 1E, \quad (2.1.84)$$

o en su forma estándar como

$$\begin{aligned} L &= e_\infty (a \wedge 1) I_E + 1I_E, \\ &= e_\infty (a \cdot 1) + 1. \end{aligned} \quad (2.1.85)$$

De esta forma, dada una línea de magnitud unitaria en la forma estándar  $L$ , se puede obtener una rotación general alrededor de la misma definiendo la rotación como

$$R_L = e^{\frac{\theta L}{2}} \quad (2.1.86)$$

Ahora, para el *movimiento rígido* (rotar + trasladar) sobre el mismo eje, es conjuntar la rotación general  $R_L$  en un ángulo  $\theta$  con una traslación sobre la misma línea  $L$  una distancia  $d$ . El motor resultante tiene la forma

$$\begin{aligned}
M &= T_d R_L = T_d T R \tilde{T} \\
&= e^{\frac{d\mathbf{e}_\infty n}{2}} e^{\frac{\theta}{2}(1+\mathbf{e}_\infty(a \cdot l))} = e^{\frac{d\mathbf{e}_\infty n}{2}} e^{\frac{\theta}{2}(1+\mathbf{e}_\infty(a \cdot l))} \\
&= e^{\frac{\theta}{2} \left( 1 + \mathbf{e}_\infty \left( \frac{a \cdot l - d n / \theta}{m} \right) \right)} = e^{\frac{\theta}{2}(1+\mathbf{e}_\infty m)}
\end{aligned} \tag{2.1.87}$$

El bivector en la parte exponencial,  $-\frac{\theta}{2}(1+\mathbf{e}_\infty m)$  es la representación de un tornillo. Si  $m=0$  el motor actúa como un *rotor*, si  $l=0$  el motor actúa como *trasladador*. Si  $m \perp l^*$  el motor actúa como una rotación general, y si  $m \not\perp l^*$  actúa como movimiento de tornillo.

## Propiedades de los Rotores

Dado un rotor  $R \in G_{4,1}$  se puede extraer el ángulo de rotación  $\theta$  usando la siguiente relación

$$c = \cos\left(\frac{\theta}{2}\right) = \frac{R + \tilde{R}}{2}. \tag{2.1.88}$$

y despejamos para obtener

$$\theta = 2 \cos^{-1}(c). \tag{2.1.89}$$

Para definir el eje de rotación sustraemos del rotor la parte escalar y nos quedamos con la bivectorial, que definiremos como  $B$ , resultando en

$$B = R - c = \frac{R - \tilde{R}}{2} \tag{2.1.90}$$

Entonces, *dualizando* y normalizando obtenemos

$$\hat{B}^* = \frac{B^*}{\sqrt{B^* \cdot B^*}}. \tag{2.1.91}$$

Por último, estandarizamos con  $L = \hat{B}^* I_c$  para obtener el eje de rotación normalizado i.e. una línea expresada en AGC. Nótese que para una rotación general  $R_L$  lo anterior también es válido, ya que la diferencia sería solo el momento de la línea dado por  $\mathbf{e}_\infty(t \cdot l)$ .

Para obtener el rotor  $R$  de una rotación general  $R_L$  (es decir, trasladar el eje de rotación hasta el origen) aplicamos

$$R = (R_L \wedge E)E \tag{2.1.92}$$

lo que es equivalente a la *reyección* de  $R_L$  del plano de Minkowsky. En general, las rotaciones  $R, R_L$  deben cumplir que su magnitud sea unitaria, veamos el caso de rotaciones generales

$$\begin{aligned}
 R_L \widetilde{R}_L &= TR \widetilde{T} \widetilde{T} R \widetilde{T} \\
 &= TR(1) \widetilde{R} \widetilde{T} \\
 &= T(1) \widetilde{T} \\
 &= 1.
 \end{aligned}
 \tag{2.1.93}$$

## Propiedades de los Motores

Dado  $M \in G_{4,1}$  podemos extraer la parte de rotación (si es rotación general se extrae la rotación al origen) de la siguiente manera

$$\begin{aligned}
 M &= TR, \\
 &= \left(1 + \frac{1}{2} e_\infty t\right) R, \\
 &= R + \frac{1}{2} e_\infty t R
 \end{aligned}
 \tag{2.1.94}$$

Entonces, aplicando la *reyección* al plano de Minkowsky (para eliminar la parte que tiene  $e_\infty$ ) obtenemos el rotor como

$$R = (M \wedge E) E. \tag{2.1.95}$$

Para obtener el trasladador manipulamos algebraicamente la ecuación (2.1.94) resultando en

$$R' = (e_- \cdot M) I_E, \tag{2.1.96}$$

con el que podemos expresar el motor como

$$M = R + e_\infty R' I_E. \tag{2.1.97}$$

Entonces, igualando (2.1.94) y (2.1.97) se obtiene

$$\begin{aligned}
 R + \frac{1}{2} e_\infty t R &= R + e_\infty R' I_E \\
 &= \frac{1}{2} e_\infty t R \\
 &= e_\infty R' I_E
 \end{aligned}$$

para luego resultar en

$$\begin{aligned}
\frac{1}{2}e_{\infty}tR\tilde{R} &= e_{\infty}R'I_E\tilde{R} \\
e_{\infty}t &= 2e_{\infty}R'I_E\tilde{R} \\
e_0 \cdot e_{\infty}t &= e_0 \cdot 2e_{\infty}R'I_E\tilde{R} \\
t &= 2R'I_E\tilde{R}
\end{aligned}
\tag{2.1.98}$$

el cual es el vector de traslación. De  $t$  se puede construir el trasladador  $T$

Dado que  $M \in G_{4,1}$  es el resultado del producto geométrico de versores ( $R\tilde{R}=1$ ,  $T\tilde{T}=1$ ), entonces  $M$  es también un versor que cumple  $M\tilde{M}=1$ .

## 2.2. Modelo Matemático Clásico

Dos características importantes que se pueden modelar de un manipulador son su *cinemática* y su *dinámica*. La manera más simple de modelar la cinemática de un manipulador es utilizando el concepto de cadena cinemática. Una cadena cinemática es un conjunto de cuerpos rígidos, también llamados eslabones, acoplados por pares cinemáticos. Un par cinemático, o unión, es el acoplamiento de dos cuerpos rígidos de tal forma que delimite su movimiento relativo.

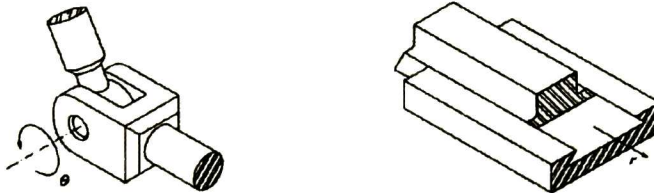


Figura 16. Uniones: Rotativa (izquierda) y Prismática (derecha).

Existen muchos tipos de pares cinemáticos, pero la mayoría se pueden modelar como una combinación del par de rotación o *revolutivo* y el par de deslizamiento o *prismático*, los cuales se consideran los pares básicos. En la Figura 16 se muestran estos tipos de pares cinemáticos o uniones.

Al primer eslabón de una cadena cinemática se le suele llamar *base* y al último se le denomina elemento *actuador final* como se muestra en la Figura 17. En ese

caso, la cadena cinemática tiene seis GDL y posee cinco uniones rotativas y una prismática.

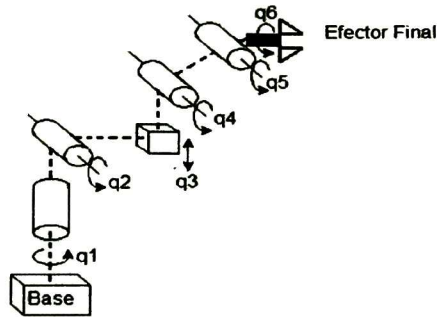


Figura 17. Manipulador serial de 6 grados de libertad.

### 2.2.1. Modelo Cinemático

Cinemática es el estudio de los posibles movimientos y configuraciones de un sistema mecánico. Por lo tanto, este tipo de modelado se refiere a la descripción del movimiento del manipulador con respecto a un sistema de referencia fijo, sin considerar las fuerzas y momentos que generan dicho movimiento. En pocas palabras, a la cinemática solo le concierne la geometría del sistema. En el modelado de manipuladores se pueden distinguir tres tipos de cinemática: la cinemática directa, la cinemática inversa y la diferencial.

#### Cinemática Directa

La cinemática directa de un manipulador consiste en determinar el mapeo entre las variables de las uniones (posiciones angulares o lineales) y la pose (posición y orientación) del elemento actuador final, con respecto a algún sistema de referencia.

De la mecánica de los cuerpos rígidos, se puede expresar la cinemática directa de un manipulador por medio de la matriz de transformación homogénea de dimensiones  $4 \times 4$  que se muestra a continuación:

$${}^bT_e(q) = \begin{bmatrix} {}^bR_e(q) & {}^bP_e(q) \\ O^T & 1 \end{bmatrix} \quad (2.2.1)$$

Donde  $q$  es el vector de las variables de uniones con dimensiones  $n \times 1$ ,  ${}^bP_e(q)$  es el vector de la posición del elemento actuador final de dimensiones  $3 \times 1$ ,

${}^bR_e(q) = [{}^b n_e, {}^b s_e, {}^b a_e]$  es la matriz de rotación del marco de referencia del elemento actuador final  $e$  con respecto al marco de referencia base  $b$  y  $O$  es un vector de ceros de dimensiones  $3 \times 1$ .

El superíndice que precede a una variable denota el marco de referencia en el que está expresada. Nótese que la matriz  ${}^bR_e(q)$  es ortogonal y sus columnas son vectores unitarios de los ejes de referencia del elemento actuador final.

Uno de los procedimientos más efectivos y utilizados para obtener la cinemática directa de un manipulador, es la *Notación Modificada de Denavit-Hartenberg*, ver [2] para más detalles.

## Cinemática Inversa

A diferencia de la cinemática directa, que obtiene la posición y orientación del elemento actuador final en base a las variables de las uniones, la cinemática inversa determina los valores de las variables de las uniones a partir de una posición  $p_e$  y orientación  $R_e$  dadas para el elemento actuador final. Este problema es muy importante, ya que las tareas a realizar están definidas en base a posiciones y orientaciones del elemento actuador final. Entonces para que el manipulador las ejecute correctamente se debe traducir esta información en valores de las variables de uniones  $q$ .

En cuanto a la cinemática directa se refiere, una vez que se tiene la información de las variables de uniones, la posición y orientación del elemento actuador final está definida de forma única. En general esto no sucede para la cinemática inversa, lo que hace a este problema mucho más complejo. Esta complejidad se debe a las siguientes razones [2]:

1. Las ecuaciones a resolver son, en general, no lineales; para las cuales no siempre es posible encontrar soluciones en forma cerrada.
2. Pueden existir múltiples o hasta infinitas soluciones (en el caso de manipuladores redundantes).
3. Puede no existir solución, debido a la estructura cinemática del manipulador.



La existencia de solución se puede garantizar si la posición y orientación del elemento actuador final pertenecen al espacio de trabajo del manipulador.

## Cinemática Diferencial

Este tipo de cinemática se requiere al mapeo entre el vector de velocidades de uniones  $\dot{q}$  ( $n \times 1$ ) y el vector de velocidad del elemento actuador final  $v$ . Dicha relación se puede observar en la ecuación siguiente

$$v = \begin{pmatrix} \dot{p} \\ w \end{pmatrix} = J(q)\dot{q} \quad (2.2.2)$$

donde  $\dot{p}$  ( $3 \times 1$ ) es el vector de velocidad lineal,  $w$  ( $3 \times 1$ ) es el vector de velocidad angular, y  $J(q)$  es lo que se conoce como matriz *Jacobiana* o *Jacobiano*.

Existen dos formas para calcular el jacobiano de un sistema mecánico: el *jacobiano geométrico* y el *jacobiano analítico*.

El *jacobiano geométrico* se obtiene mediante un procedimiento geométrico, basado en el cálculo de lo que contribuye cada velocidad de unión a la velocidad lineal y angular del elemento actuador final. Estas contribuciones están dadas por la siguiente relación:

$$\begin{pmatrix} \dot{p} \\ w \end{pmatrix} = \begin{pmatrix} \xi_1 a_1 + \bar{\xi}_1 (a_1 \times p_{1n}) & \dots & \xi_n a_n + \bar{\xi}_n (a_n \times p_{nn}) \\ \bar{\xi}_1 a_1 & \dots & \bar{\xi}_n a_n \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{pmatrix} = J(q)\dot{q} \quad (2.2.3)$$

Definiendo  $Z_k$  como el eje de rotación de la unión  $k$  como se muestra en la Figura 18, entonces  $a_k$  es el vector unitario de  $Z_k$  y  $p_{kn}$  denota el vector definido desde el origen del marco de referencia  $k$  al origen del marco de referencia  $n$ .

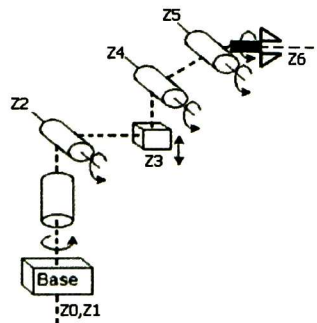


Figura 18. Ejes de rotación para un manipulador de 6 grados de libertad.

Los términos  $\xi_i$  y  $\bar{\xi}_i$ , definen las características del tipo de unión de la siguiente manera:

$$\bar{\xi}_i = 1 - \xi_i$$

$\xi_i = 0$  si la unión  $i$  es rotativa.

- $\xi_i = 1$  si la unión  $i$  es prismática.

Nótese que  $J_n$  es una función de  $q$  a través de los vectores  $a_k$  y  $p_{kn}$ , los cuales pueden ser obtenidos sobre la base de la cinemática directa.

El *jacobiano analítico* suele utilizarse si la orientación y posición del elemento actuador final están especificados en términos de un número mínimo de parámetros en el espacio de trabajo del manipulador.

Si estructuramos el vector de posición y orientación del elemento actuador final de la siguiente forma:

$$x = (p_e \quad \phi_e)^T$$

donde  $p_e$  es el vector de posición y  $\phi_e$  el de orientación. Entonces es posible calcular el jacobiano por la diferenciación directa de la ecuación anterior, por ejemplo:

$$\dot{x} = (\dot{p}_e \quad \dot{\phi}_e)^T = J_a(q)\dot{q} \quad (2.2.4)$$

Existe una relación entre los dos tipos de jacobianos, determinada por:

$$J_n = \begin{pmatrix} I & 0 \\ 0 & T(\phi_e) \end{pmatrix} J_a \quad (2.2.5)$$

donde  $T(\phi_e)$  es una matriz de transformación que depende de los parámetros usados para representar la orientación del elemento actuador final. Cabe mencionar que siempre se puede encontrar una matriz de mapeo entre el jacobiano analítico y el geométrico si  $T(\phi_e)$  tiene rango pleno.

### 2.2.2. Modelo Dinámico

El modelado dinámico de un robot manipulador consiste en encontrar el mapeo entre las fuerzas ejercidas sobre la estructura del manipulador y las posiciones, velocidades y aceleraciones de las uniones. Las formas más usadas para calcular dicho modelo son: las ecuaciones de Lagrange y las ecuaciones de Newton-Euler. La primera es más simple y sistemática, y la segunda más eficiente desde un punto de

vista computacional. En el resto de esta sección se explicarán los fundamentos de estos dos procedimientos de modelado dinámico.

## Ecuaciones de LaGrange

Si tomamos en cuenta que las variables de las uniones  $q_i$  constituyen un conjunto de coordenadas generalizadas del sistema, el modelo dinámico del manipulador puede ser inferido por las ecuaciones de LaGrange como

$$\frac{d}{dt} \left( \frac{\partial \mathfrak{S}}{\partial \dot{q}} \right) - \frac{\partial \mathfrak{S}}{\partial q} = \tau \quad (2.2.6)$$

donde  $\tau$  es el vector de los efectos generados por los actuadores en las uniones, un esfuerzo torsión de para una unión rotativa y un fuerza lineal para una unión prismática, y  $\mathfrak{S}$  es un término conocido como el *Lagrangiano*, y equivale a

$$\mathfrak{S} = K - U \quad (2.2.7)$$

donde  $K$  es la energía cinética del sistema y  $U$  es la energía potencial.

Entonces, necesitamos obtener las expresiones de  $K$  y  $U$  para definir el modelo dinámico.

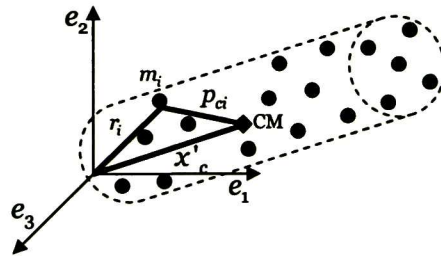


Figura 19. Eslabón como un sistema de partículas y su centro de masas.

La **energía cinética**  $K$  del manipulador es la suma de las energías cinéticas de cada eslabón que conforman el manipulador. Cada eslabón puede ser visto como un sistema de  $z$  partículas, como el mostrado en la Figura 19, y la energía cinética del eslabón estaría dada por

$$K_j = \sum_{i=1}^z \frac{1}{2} m_i V_i^2 \quad (2.2.8)$$

Note que  $r_i$  representa posición de la partícula,  $x'_c$  la posición actual del centro de masas y  $p_{ci}$  la posición de la partícula respecto al centro de masas, lo que nos da la relación  $r_i = x'_c + p_{ci}$  y su derivada como

$$V_i = \dot{x}'_c + \dot{p}_{ci} \quad (2.2.9)$$

donde  $V_i$  es la velocidad de la partícula,  $\dot{x}'_c$  la velocidad del centro de masas del eslabón y  $\dot{p}_{ci}$  la velocidad de la partícula respecto al centro de masas.

Usando (2.2.9) podemos reescribir a (2.2.8) como

$$\begin{aligned} K_j &= \sum_{i=1}^z \frac{1}{2} m_i (\dot{x}'_c + \dot{p}_{ci})^2 \\ &= \frac{1}{2} \sum_{i=1}^z m_i (\dot{x}'_c)^2 + \sum_{i=1}^z m_i \dot{x}'_c \dot{p}_{ci} + \frac{1}{2} \sum_{i=1}^z m_i \dot{p}_{ci}^2 \end{aligned} \quad (2.2.10)$$

Como  $\dot{x}'_c$  no depende del índice de la sumatoria, se puede sacar fuera de la misma. Además, por definición tenemos que la masa del eslabón es

$$m_j = \sum_{i=1}^z m_i, \text{ y también que } 0 = \sum_{i=1}^z m_i \dot{p}_{ci}.$$

lo que nos resulta en

$$K_j = \frac{1}{2} m_j (\dot{x}'_c)^2 + \frac{1}{2} \sum_{i=1}^z m_i \dot{p}_{ci}^2 \quad (2.2.11)$$

Así, podemos ver que la energía cinética de un eslabón respecto a un sistema de referencia, puede verse como la suma de dos partes [32]: (a) *la energía cinética de la masa total moviéndose respecto al sistema de referencia con la misma velocidad del centro de masas*, más (b) *la energía cinética del movimiento de las partículas respecto al centro de masas (momento de inercia del eslabón)*.

Otra forma de ver a  $K$  es como la siguiente forma cuadrática

$$K = \frac{1}{2} \dot{q}^T H(q) \dot{q} \quad (2.2.12)$$

donde  $H(q)$  es la matriz de inercia ( $n \times n$ ) del manipulador, la cual es simétrica y definida positiva.

La **energía potencial**  $U$  no depende de la velocidad con que se desplazan los eslabones, sino de la posición actual de los centros de masa de los mismos,  $x'_{ci}$  para el  $i^{\text{ésimo}}$  eslabón, y para calcularlos se utiliza la ecuación de cinemática directa del manipulador. La ecuación de la energía potencial del sistema se define como

$$U = \sum_{i=1}^n U_i = \sum_{i=1}^n \mathbf{x}'_{ci} \mathbf{F}_i \quad (2.2.13)$$

donde  $\mathbf{F}_i$  son las fuerzas conservativas actuantes en el eslabón. Como suponemos que trabajamos con manipuladores rígidos, la única fuerza conservativa es la gravedad, por lo que  $\mathbf{F}_i = m_i \hat{\mathbf{g}}$ .

Una vez definidas la energía potencial y cinética del manipulador, podemos obtener las derivadas parciales que conforman a las ecuaciones de LaGrange (2.2.6). Primero, note que  $U$  no depende de  $\dot{\mathbf{q}}$  por lo que

$$\frac{\partial \mathfrak{S}}{\partial \dot{\mathbf{q}}} = \frac{\partial \mathbf{K}}{\partial \dot{\mathbf{q}}} - \frac{\partial U}{\partial \dot{\mathbf{q}}} = \frac{\partial \mathbf{K}}{\partial \dot{\mathbf{q}}} \quad (2.2.14)$$

y como  $\mathbf{K}$  es una forma cuadrática expresada por (2.2.12), la derivada parcial resulta en

$$\frac{\partial \mathfrak{S}}{\partial \dot{\mathbf{q}}} = \mathbf{H}(\mathbf{q}) \dot{\mathbf{q}}. \quad (2.2.15)$$

Ahora, podemos obtener la derivada respecto al tiempo como

$$\frac{d}{dt} \left( \frac{\partial \mathfrak{S}}{\partial \dot{\mathbf{q}}} \right) = \frac{d}{dt} (\mathbf{H} \dot{\mathbf{q}}) = \mathbf{H} \ddot{\mathbf{q}} + \dot{\mathbf{H}} \dot{\mathbf{q}}. \quad (2.2.16)$$

Siguiendo con la otra derivada parcial obtenemos

$$\frac{\partial \mathfrak{S}}{\partial \mathbf{q}} = \frac{\partial \mathbf{K}}{\partial \mathbf{q}} - \frac{\partial U}{\partial \mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^T \left( \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \right) \dot{\mathbf{q}} - \frac{\partial U}{\partial \mathbf{q}}. \quad (2.2.17)$$

y sustituyendo (2.2.16) y (2.2.17) en (2.2.6) obtenemos

$$\mathbf{H} \ddot{\mathbf{q}} + \dot{\mathbf{H}} \dot{\mathbf{q}} - \frac{1}{2} \dot{\mathbf{q}}^T \left( \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \right) \dot{\mathbf{q}} + \frac{\partial U}{\partial \mathbf{q}} = \boldsymbol{\tau}. \quad (2.2.18)$$

Si definimos

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{H}} - \frac{1}{2} \dot{\mathbf{q}}^T \left( \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \right) \quad (2.2.19)$$

como el vector de fuerzas centrífugas y de Coriolis, y a

$$\mathbf{g}(\mathbf{q}) = \frac{\partial U}{\partial \mathbf{q}} \quad (2.2.20)$$

como el vector de fuerzas gravitatorias de dimensiones  $(n \times 1)$ , obtenemos las siguientes *ecuaciones de movimiento de LaGrange*

$$\mathbf{H}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}. \quad (2.2.21)$$

Una propiedad interesante de  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ , es que es una función cuadrática de  $\dot{\mathbf{q}}$ , por lo tanto podemos escribir su elemento genérico como

$$C_{ij} = \sum_{k=1}^n c_{ijk} \dot{q}_j \dot{q}_k$$

donde

$$c_{ijk} = \frac{1}{2} \left( \frac{\partial H_{ij}}{\partial q_k} + \frac{\partial H_{ik}}{\partial q_j} + \frac{\partial H_{jk}}{\partial q_i} \right),$$

estos términos son conocidos como los *Símbolos de Christoffel* del primer tipo, y hacen que la matriz  $\dot{H} - 2C$  sea anti-simétrica; esta propiedad es muy útil para fines de control. Para más detalles sobre la obtención de las Ecuaciones de LaGrange referirse a [14].

El modelo dinámico en la forma (2.2.21) está representado por un conjunto de  $n$  ecuaciones diferenciales no lineales de segundo orden acopladas, que relaciona las posiciones, velocidades y aceleraciones de las uniones con los momentos de fuerza de las mismas. Los elementos  $H$ ,  $C$  y  $g$  son funciones de los parámetros cinemáticos y dinámicos de los eslabones, y se pueden obtener expresando la energía cinética y potencial en términos de la posición y velocidad de las uniones.

## Ecuaciones de Newton-Euler

Este tipo de formulación permite obtener la dinámica de un manipulador rígido sin derivar las expresiones explícitas de los términos  $H$ ,  $C$  y  $g$ . Las ecuaciones son obtenidas como resultado de dos cálculos recursivos; nombrados *recursivo hacia adelante*, para calcular las velocidades y aceleraciones de los eslabones del eslabón 0 al  $n$ ; y *recursivo hacia atrás*, para calcular las fuerzas y momentos de los eslabones desde eslabón  $n$  al 0. Si definimos a  $p_i$  como el vector de posición del origen del marco de referencia del eslabón  $i$ . Entonces la velocidad lineal del origen del marco de referencia del eslabón  $i$  está dada por

$$\dot{p}_i = \dot{p}_{i-1} + w_{i-1} \times p_{i-1} + \xi_i \dot{q}_i a_i \quad (2.2.22)$$

donde la velocidad angular del eslabón  $i$  está dada por

$$w_i = w_{i-1} + \bar{\xi}_i \dot{q}_i a_i \quad (2.2.23)$$

Derivando (2.2.22) y (2.2.23) obtenemos

$$\begin{aligned} \ddot{p}_i = & \ddot{p}_{i-1} + \dot{w}_{i-1} \times p_{i-1,i} + w_{i-1} \times (w_{i-1} \times p_{i-1,i}) \\ & + \xi_i (\ddot{q}_i a_i + 2w_{i-1} \times \dot{q}_i a_i) \end{aligned} \quad (2.2.24)$$

y

$$\dot{w}_i = \dot{w}_{i-1} + \bar{\xi}_i (\ddot{q}_i a_i + w_{i-1} \times \dot{q}_i a_i) \quad (2.2.25)$$

Además, si definimos  $p_{ci}$  como el centro de masas del eslabón  $i$ , su aceleración estaría dada por

$$\ddot{p}_{ci} = \ddot{p}_i + \dot{w}_i \times r_i + w_i \times (w_i \times r_i) \quad (2.2.26)$$

donde  $r_i$  es el vector de posición del centro de masas del eslabón  $i$  con respecto al marco de referencia  $i$ , como lo define la siguiente ecuación

$$r_i = p_{ci} - p_i.$$

Con referencia a la figura 16, donde definimos a  $O_i$  como el origen del marco de referencia del  $i^{\text{ésimo}}$  eslabón, las ecuaciones de Newton definen un balance de fuerzas actuando sobre el eslabón  $i$  en la forma de

$$\gamma_i = m_i \ddot{p}_{ci} - m_i \hat{g} + \gamma_{i+1} \quad (2.2.27)$$

donde  $\gamma_i$  denota la fuerza ejercida por el eslabón  $i-1$  sobre el eslabón  $i$  en el origen del marco de referencia del eslabón  $i$ . Sustituyendo (2.2.26) en (2.2.27) nos da

$$\gamma_i = m_i \ddot{p}_i + \dot{w}_i \times m_i r_i + w_i \times (w_i \times m_i r_i) + \gamma_{i+1} \quad (2.2.28)$$

El efecto de  $m_i \hat{g}$  será automáticamente introducido tomando  $\ddot{p}_0 = -\hat{g}$ .

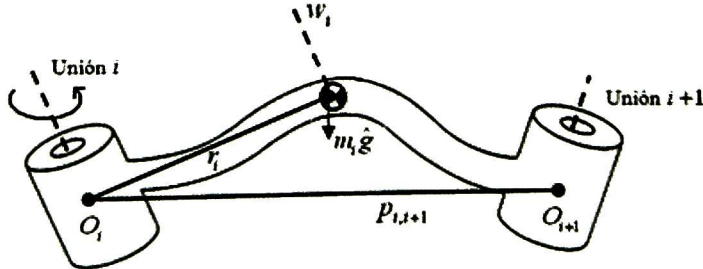


Figura 20. Caracterización del  $i^{\text{ésimo}}$  eslabón para las ecuaciones de Newton-Euler.

Las ecuaciones de Euler también ofrecen un equilibrio de momentos actuando sobre el eslabón  $i$  (referido al centro de masas) en la forma

$$\mu_i = \hat{I}_i \dot{w}_i + w_i \times (\hat{I}_i w_i) + r_i \times \gamma_i + \mu_{i+1} + p_{ci,i+1} \times \gamma_{i+1} \quad (2.2.29)$$

donde  $\hat{I}_i$  es el tensor de inercia del eslabón  $i$  con respecto de su centro de masas. Aplicando el *teorema de Steiner* ( ver Glosario), el tensor de inercia respecto al origen del marco de referencia  $i$  está dado por

$$I_i = \hat{I}_i + m_i S^T(r_i) S(r_i)$$

entonces (2.2.29), vía (2.2.28), puede ser reescrita como

$$\mu_i = I_i \dot{w}_i + w_i \times (I_i w_i) + m_i r_i \ddot{p}_i + \mu_{i+1} + p_{i,i+1} \times \gamma_{i+1}.$$

El momento de fuerza del  $i^{\text{ésimo}}$  eslabón  $u_i$  puede ser obtenido proyectando  $\gamma_i(\mu_i)$  sobre el eje  $Z_i$ , como

$$u_i = (\xi_i \gamma_i + \bar{\xi}_i \mu_i)^T a_i + I_{mi} \ddot{q}_i + F_{si} \text{sign}(\dot{q}_i) + F_{vi} \dot{q}_i$$

donde las contribuciones del motor de inercia y la fricción de las uniones se han agregado. Para información más detallada sobre este método referirse a [14].

## Dinámica Directa e Inversa

Obtener el modelo dinámico de un manipulador robótico es de interés, tanto para simulación como para el control de dicho dispositivo. Para la simulación necesitamos las denominadas *dinámicas directas* del manipulador, esto es que para un conjunto conocido de momentos de fuerza de unión  $u_i$ , se obtengan las posiciones, velocidades y aceleraciones resultantes. Para lo segundo, nos conciernen las *dinámicas inversas* del manipulador que, contrario a las dinámicas directas, de un conjunto conocido de posiciones, velocidades y aceleraciones de las uniones se obtienen los momentos de fuerza de uniones resultantes.

El proceso computacional para la dinámica directa es, en general, mayor que el necesario para obtener la dinámica inversa de un manipulador. Esto es porque el modelado dinámico da naturalmente el mapeo de posiciones, velocidades y aceleraciones de unión a los momentos de fuerza de unión, como se muestra en la ecuación (2.2.21).

En cambio para la dinámica inversa suele procederse de la siguiente manera: de la ecuación (2.2.21) se despeja

$$\ddot{q} = H(q)^{-1} (u - C(q, \dot{q}) \dot{q} - g(q)), \quad (2.2.30)$$

y de esta ecuación podemos obtener una representación en espacio de estados como



$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -H(x_1)^{-1}(C(x_1, x_2)x_2 + g(x_1)) \end{bmatrix} + \begin{bmatrix} 0 \\ H(x_1)^{-1}u \end{bmatrix} \quad (2.2.31)$$

donde  $x_1 = q$  es el vector de posiciones angulares ( $n \times 1$ ) de las uniones y  $x_2 = \dot{q}$  es el vector de velocidades angulares ( $n \times 1$ ), esta representación nos provee de  $2n$  ecuaciones diferenciales ordinarias, las cuales pueden ser integradas en el tiempo bajo condiciones iniciales conocidas para obtener los estados  $x_1$  y  $x_2$ . Cabe destacar que la representación (2.2.31) está en la forma No-Lineal Controlable por Bloques (NLCB).

Los términos  $H(x_1)$ ,  $C(x_1, x_2)$  y  $g(x_1)$  pueden ser obtenidos por medio de las ecuaciones de LaGrange, pero un método más eficiente consiste en usar las ecuaciones de Newton-Euler de la siguiente manera: los vectores  $C(x_1, x_2)$  y  $g(x_1)$  se obtienen de computar  $u$  con  $\ddot{x}_i = 0$  de la ecuación (2.2.31). Entonces, las columnas  $h_i$  de la matriz  $H(x_1)$  se obtienen como el momento de fuerza  $u$  con  $\hat{g} = 0$ ,  $x_2 = 0$ ,  $\dot{q}_i = 1$  y  $\dot{q}_j = 0$  para  $j \neq i$ . Iterando este procedimiento para  $i = 1, \dots, n$  podemos obtener toda la matriz de inercia.

El sistema (2.2.31), como la mayoría de los sistemas robóticos, se encuentra en una forma conocida como *Forma No-Lineal Controlable por Bloques*. En la siguiente sección se explicará en detalle este tipo de sistema.

## 2.3. Forma No-lineal Controlable por Bloques

Para un sistema MIMO (Multi Input-Multi Output) no lineal perturbado en la forma No-Lineal Controlable por Bloques (NLCB) tenemos:

$$\begin{aligned} \dot{x}_1 &= f_1(x_1) + b_1(x_1)x_2 + \lambda_1(x_1, t) \\ &\vdots \\ \dot{x}_{r-1} &= f_{r-1}(\bar{x}_{r-1}) + b_{r-1}(\bar{x}_{r-1})x_r + \lambda_{r-1}(\bar{x}_{r-1}, t) \\ \dot{x}_r &= f_r(\bar{x}_r) + b_r(\bar{x}_r)u + \lambda_r(\bar{x}_r, t) \end{aligned} \quad (2.3.1)$$

donde  $\bar{x}_i = (x_1, \dots, x_i)$ ,  $x_i \in X_i \subset \mathbb{R}^n$  La salida del sistema es  $y = h(X)$ , las funciones  $f_i(\bullet)$  y  $b_i(\bullet)$  son campos de vectores suaves definidos en  $X_i$ ; y  $\lambda_i(\bullet)$  son términos de perturbaciones generados por variaciones paramétricas e incertidumbre en el modelo de la planta.

El vector  $\lambda$  es denominado comúnmente como perturbación *matched* debido a que se encuentra dentro del sub-espacio de control, y los términos  $\lambda_i(\bullet)$  para  $i=1, \dots, r-1$  son conocidos como perturbaciones *no matched* debido a que se encuentran fuera de este sub-espacio. El sub-espacio de control es el generado por las columnas de  $b_r(\bullet)$ , y se denomina a una perturbación *matched*, si el vector de control  $u$  puede afectar a todos los componentes de la perturbación mediante la matriz de control  $b_r(\bullet)$ . La estructura del sistema está definida por los enteros  $(n_1, \dots, n_r)$  y cumplen con la condiciones

$$n_1 \leq \dots \leq n_r \text{ y } \sum_{i=1}^r n_i = n$$

donde  $n$  es el orden total del sistema.

## 2.4. Modos Deslizantes

Los Modos Deslizantes (MD) juegan un papel muy importante en la teoría de control de estructura variable. Debido a su baja sensibilidad a perturbaciones externas y variaciones paramétricas, este tipo de control representa una herramienta muy eficiente en el control de sistemas dinámicos complejos de alto orden operando bajo condiciones de incertidumbre, lo cual es muy común en muchos procesos de tecnología moderna.

La idea principal es forzar la ocurrencia de modos deslizantes en alguna variedad o superficie del espacio de estados. Por lo general estas superficies se diseñan como la intersección de *hiperplanos* en el espacio de estados, de tal forma que una vez que el sistema ha alcanzado esta superficie, la estructura del lazo de retroalimentación se adapta para "deslizar" el estado del sistema sobre esta superficie, entonces la respuesta del sistema depende solo del gradiente de la superficie deslizante y permanece insensible a las perturbaciones que cumplen con la condición *matching*.

### 2.4.1. Principios de Diseño del Control por Modos Deslizantes

Un control por MD es diseñado en dos etapas:

- I. Diseñar la *superficie deslizante*  $s(\mathbf{x}) = 0$ , como un sub-espacio del espacio de estados del sistema, de tal forma que el sistema tenga propiedades deseadas (estabilidad, robustez, seguimiento de trayectorias, etc.). Esta superficie puede ser diseñada utilizando cualquier técnica lineal o no-lineal.
- II. Escoger un control discontinuo acotado que fuerce al sistema a alcanzar la superficie diseñada y permanecer sobre la misma.

$$\mathbf{u}(\mathbf{x}) = \begin{cases} -\mathbf{u}_0 & \text{si } s(\mathbf{x}) > 0 \\ \mathbf{u}_0 & \text{si } s(\mathbf{x}) < 0 \end{cases}$$

## Control Equivalente

La naturaleza del control por MD es producir un cambio de alta frecuencia en la ley de control, por lo que es difícil de visualizar el comportamiento en lazo cerrado del sistema. Para poder visualizar y analizar este comportamiento por medio de la teoría clásica de ecuaciones diferenciales es necesario *regularizar* el sistema, i.e. cambiar o reducir el sistema original a uno que se comporte “muy similar” Uno de los métodos más conocidos y usados es el *control equivalente*, el cual describiremos a continuación.

Considere el sistema

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{u} \quad (2.4.1)$$

donde  $\mathbf{x} \in \mathbb{R}^n$  es el vector de estados,  $\mathbf{u} \in \mathbb{R}^m$  es la señal de control, y  $\text{rank}\{\mathbf{B}(\mathbf{x})\} = m$ .

Igualando la derivada de la superficie deslizante a cero, es posible encontrar el control equivalente y la *ecuación de modo deslizante*, la cual determina el comportamiento del sistema en lazo cerrado. Entonces tenemos:

$$\dot{s} = \frac{\partial s}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial s}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) + \frac{\partial s}{\partial \mathbf{x}} \mathbf{B}(\mathbf{x})\mathbf{u}_{\text{eq}} = 0 \quad (2.4.2)$$

donde  $\mathbf{u}_{\text{eq}}$  es el control equivalente.

Despejando nos queda

$$\mathbf{u}_{\text{eq}} = -\left[\frac{\partial s}{\partial \mathbf{x}} \mathbf{B}(\mathbf{x})\right]^{-1} \left[\frac{\partial s}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x})\right]. \quad (2.4.3)$$

Sustituyendo (2.4.3) en (2.4.2), obtenemos la ecuación de modo deslizante como

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) - \mathbf{B}(\mathbf{x}) \left[ \frac{\partial \mathbf{s}}{\partial \mathbf{x}} \mathbf{B}(\mathbf{x}) \right]^{-1} \left[ \frac{\partial \mathbf{s}}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) \right]. \quad (2.4.4)$$

La variedad deslizante  $\mathbf{s}(\mathbf{x})$  debe ser diseñada de tal forma que (2.4.4) tenga las propiedades deseadas.

## 2.4.2. Modos Deslizantes Anidados (MDA)

El control por modos deslizantes estándar no es robusto a las perturbaciones *no-matched*, ese es el punto que atacan los modos deslizantes anidados para los sistemas en la forma No-Lineal Controlable Por Bloques (NLCB), explicada en la sección 2.3 y que por conveniencia re-escribo a continuación.

$$\begin{aligned} \dot{\mathbf{x}}_1 &= \mathbf{f}_1(\mathbf{x}_1) + \mathbf{b}_1(\mathbf{x}_1)\mathbf{x}_2 + \lambda_1(\mathbf{x}_1, t) \\ &\quad \vdots \\ \dot{\mathbf{x}}_{r-1} &= \mathbf{f}_{r-1}(\bar{\mathbf{x}}_{r-1}) + \mathbf{b}_{r-1}(\bar{\mathbf{x}}_{r-1})\mathbf{x}_r + \lambda_{r-1}(\bar{\mathbf{x}}_{r-1}, t) \\ \dot{\mathbf{x}}_r &= \mathbf{f}_r(\bar{\mathbf{x}}_r) + \mathbf{b}_r(\bar{\mathbf{x}}_r)\mathbf{u} + \lambda_r(\bar{\mathbf{x}}_r, t) \end{aligned} \quad (2.4.5)$$

El principio de los MDA es aprovechar la estructura de los sistemas en la forma NLCB, de tal manera que en el  $i^{\text{ésimo}}$  bloque del sistema, siendo  $i=1, \dots, r$  y  $r$  el orden del sistema, se toma al vector  $\mathbf{x}_{i+1}$  como un vector de control ficticio, pseudo-control o control virtual; el cual nos ayudará a definir la dinámica de dicho bloque y la referencia para el bloque siguiente. Esto se puede ver claramente en (2.4.5) cuyo primer bloque  $\dot{\mathbf{x}}_1$  tomaría como control virtual a  $\mathbf{x}_2$ .

Este proceso se hace de forma iterativa hasta llegar al  $r^{\text{ésimo}}$  bloque donde aparece el vector de control real  $\mathbf{u}$  [6].

De esta manera las perturbaciones fuera del sub-espacio generado por  $\mathbf{u}$ , se encuentran dentro del sub-espacio generado por sus respectivos controles virtuales y pueden ser rechazadas. En otras palabras, poder rechazar perturbaciones *no matched*.

## Principios de Diseño del Control por MDA

Consideremos un sistema en la forma NLCB (2.3.1), entonces para diseñar un control por MDA se necesitan  $r$  pasos.

**Paso 1:** Se toma a  $x_2$  como un control virtual para el primer bloque del sistema, por lo que, al definir este, estaremos obteniendo la referencia para el segundo bloque  $x_{2,ref}$ . El primer bloque tendría entonces la forma:

$$\begin{aligned}\dot{x}_1 &= f_1(x_1) + b_1(x_1)x_{2,ref} + \lambda_1(x_1, t) \\ s_1 &= x_1 - y_{ref}\end{aligned}\tag{2.4.6}$$

donde  $s_1$  sería la superficie deslizante para el primer bloque. Entonces el control virtual estaría definido por

$$x_{2,ref} = b_1(x_1)^{-1}[-f_1(x_1) - k_1 \text{sigm}(\varepsilon_1 s_1)].\tag{2.4.7}$$

con  $k_1$  como la ganancia de control, y  $\varepsilon_1$  define la pendiente de la función sigmoidea. La condición de estabilidad de este bloque es

$$k_1 > |\lambda_1(x_1)| + \alpha_1\tag{2.4.8}$$

donde  $\alpha_1$  es un escalar positivo que define el máximo período de alcance a  $s_1$ . La función de Lyapunov  $V_1 = \frac{s_1^2}{2}$  demuestra la estabilidad robusta de (2.4.6), i.e.  $s_1 \rightarrow 0$  en tiempo finito.

**Paso  $i = \{2, \dots, r-1\}$ :** Este bloque tendría la forma

$$\begin{aligned}\dot{x}_i &= f_i(\bar{x}_i) + b_i(\bar{x}_i)x_{i+1,ref} + \lambda_i(\bar{x}_i, t) \\ s_i &= x_i - x_{i,ref}\end{aligned}\tag{2.4.9}$$

y el control virtual estaría definido por

$$x_{i+1,ref} = b_i(\bar{x}_i)^{-1}[-f_i(\bar{x}_i) - k_i \text{sigm}(\varepsilon_i s_i) + \dot{x}_{i,ref}].\tag{2.4.10}$$

La condición de estabilidad de este bloque es

$$k_i > |\lambda_i(\bar{x}_i)| + \alpha_i\tag{2.4.11}$$

donde  $\alpha_i$  define el máximo período de alcance a  $s_i$ , y  $V_i = s_i^2/2$  demuestra la estabilidad robusta del bloque, i.e.  $s_i \rightarrow 0$  en tiempo finito.

**Paso  $r$ :** El último bloque tiene la forma

$$\begin{aligned}\dot{x}_r &= f_r(\bar{x}_r) + b_r(\bar{x}_r)u + \lambda_r(\bar{x}_r, t) \\ s_r &= x_r - x_{r,ref}\end{aligned}\tag{2.4.12}$$

y la señal de control se define como

$$u = b_r(\bar{x}_r)^{-1}[-f_r(\bar{x}_r) - k_r \text{sigm}(\varepsilon_r s_r) + \dot{x}_{r,ref}].\tag{2.4.13}$$

La condición de estabilidad de este bloque es

$$k_r > |\lambda_r(\bar{x}_r)| + \alpha_r\tag{2.4.14}$$

donde  $\alpha_r$  define el máximo período de alcance a  $s_r$ ,  $V_r = s_r^2/2$  demuestra la estabilidad robusta del bloque, i.e.  $s_r \rightarrow 0$  en tiempo finito, y el objetivo de control del sistema global es alcanzado. El análisis de estabilidad y la obtención de las cotas de las ganancias de control se explican a detalle en [6].

### 2.4.3. Modos Deslizantes Integrales (MDI)

Durante la respuesta del sistema a un controlador de modos deslizantes se pueden definir dos etapas: *la etapa de aproximación y la etapa deslizante*. La etapa de aproximación se presenta cuando la respuesta del sistema se acerca a la superficie, por su parte la etapa deslizante es cuando ya está sobre la superficie. En la primera el sistema no es robusto a las perturbaciones externas y por lo general para darle robustez durante toda su respuesta se implementa control de alta ganancia, pero inevitablemente se compromete la estabilidad del sistema limitando así la aplicación de este tipo de control.

Otra alternativa para darle robustez durante toda la respuesta al sistema, es usar el concepto de modos deslizantes integrales (MDI) [33]. La base de este tipo de modos deslizantes es la eliminación de la etapa de aproximación asegurando así robustez durante toda la respuesta del sistema.

El diseño del controlador por MDI presupone la existencia de un controlador diseñado con anterioridad para la planta nominal (sin perturbaciones), agregándole a este controlador existente un término discontinuo para forzar la ocurrencia de modos deslizantes y así obtener una respuesta como si la planta fuera la nominal con su controlador en lazo cerrado, a pesar de las perturbaciones externas y variaciones paramétricas del sistema.

## Principios de Diseño de Control por MDI

Para un sistema como

$$\dot{x} = f(x) + b(x)u + \lambda(x, t)$$

donde  $x \in X_n \subset \mathbb{R}^n$  es el vector de estados,  $b(x)$  es la matriz de control y  $\lambda(x, t)$  es el vector de perturbaciones *matched* debida a factores externos, variaciones

paramétricas e incertidumbre de modelado. Se asume a  $\lambda(\mathbf{x}, t)$  como acotado por una función escalar positiva conocida de la siguiente manera

$$\|\lambda(\mathbf{x}, t)\| \leq h^+(\mathbf{x}, t)$$

Entonces definimos

$$\mathbf{u} = \mathbf{u}_0 + \mathbf{u}_1 \quad (2.4.15)$$

donde  $\mathbf{u}_0$  es el control prediseñado para el sistema nominal, y  $\mathbf{u}_1$  es el control discontinuo que diseñaremos para rechazar la perturbación. Entonces

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{b}(\mathbf{x})\mathbf{u}_0 + \mathbf{b}(\mathbf{x})\mathbf{u}_1 + \lambda(\mathbf{x}, t) \quad (2.4.16)$$

Ahora definimos la superficie deslizante como

$$\mathbf{s} = \sigma(\mathbf{x}) + \mathbf{z} \quad (2.4.17)$$

donde el término  $\sigma(\mathbf{x})$  es una función de los estados del sistema (definida por algún criterio) y  $\mathbf{z}$  es el término integral que será determinado después.

Si derivamos (2.4.17) y sustituimos (2.4.16) en el resultado, tenemos

$$\dot{\mathbf{s}} = \frac{\partial \sigma}{\partial \mathbf{x}} (\mathbf{f}(\mathbf{x}) + \mathbf{b}(\mathbf{x})\mathbf{u}_0 + \mathbf{b}(\mathbf{x})\mathbf{u}_1 + \lambda(\mathbf{x}, t)) + \dot{\mathbf{z}}, \quad (2.4.18)$$

y definiendo

$$\dot{\mathbf{z}} = \frac{\partial \sigma}{\partial \mathbf{x}} (-\mathbf{f}(\mathbf{x}) - \mathbf{b}(\mathbf{x})\mathbf{u}_0) \quad \mathbf{z}(0) = -\sigma(\mathbf{x}(0))$$

la ecuación (2.4.18) se reduce a

$$\dot{\mathbf{s}} = \frac{\partial \sigma}{\partial \mathbf{x}} (\mathbf{b}(\mathbf{x})\mathbf{u}_1 + \lambda(\mathbf{x}, t)). \quad (2.4.19)$$

El paso siguiente es definir el control por MD clásico (de primer orden) que rechace las perturbaciones como

$$\mathbf{u}_1 = -\mathbf{K}(\mathbf{x})\mathbf{b}(\mathbf{x})^{-1} \text{sign}(\mathbf{s}) \quad (2.4.20)$$

donde  $\mathbf{K}(\mathbf{x})$  se escoge de tal forma que se fuerce la ocurrencia de modos deslizantes por el control discontinuo  $\mathbf{u}_1$ , además de que ocurran desde el primer instante ya que  $\mathbf{s}(0) = \sigma(\mathbf{x}(0)) + \mathbf{z}(0) = 0$ . Por lo tanto

$$\mathbf{s} = 0, \quad \dot{\mathbf{s}} = 0,$$

entonces, ya estando en la etapa deslizante, tenemos que

$$\dot{\mathbf{s}} = \frac{\partial \sigma}{\partial \mathbf{x}} (\mathbf{b}(\mathbf{x})\mathbf{u}_{1\text{eq}} + \lambda(\mathbf{x}, t)) = 0,$$

de donde podemos obtener la igualdad

$$\mathbf{b}(\mathbf{x})\mathbf{u}_{1\text{eq}} = -\lambda(\mathbf{x}, t) \quad (2.4.21)$$

y sustituyendo en (2.4.16) nos queda

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{b}(\mathbf{x})\mathbf{u}_0$$

lo que es equivalente al sistema nominal con la señal de control  $\mathbf{u}_0$ , el cual era nuestro objetivo.

#### 2.4.4. Algoritmo Super-Twisting (ST)

El algoritmo *Super-Twisting* ha sido desarrollado para controlar sistemas de grado relativo uno, con una ley de control totalmente continua, permitiendo compensar exactamente perturbaciones Lipshitz y asegurando convergencia en tiempo finito. Además permite evitar el efecto *Chattering* en Control de Estructura Variable. Su nombre proviene de las trayectorias en el plano *2-deslizante* que se caracterizan por un simular un *Twisting* sobre el origen, para más detalles refiérase a [33]. La ley de control continua está constituida por dos términos; uno se define por la derivada de un término discontinuo y el otro es una función continua de la variable deslizante conocida. De manera más formal, sea un sistema

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{t}, \mathbf{x}, \mathbf{u}), \mathbf{s} = \mathbf{g}(\mathbf{t}, \mathbf{x}) \in \mathbb{R}, \mathbf{u} = \mathbf{U}(\mathbf{t}, \mathbf{x}) \in \mathbb{R} \quad (2.4.22)$$

donde  $\mathbf{x} \in \mathbb{R}^n$   $\mathbf{t}$  es tiempo y  $\mathbf{f}$   $\mathbf{g}$  son funciones suaves. El objetivo de control es lograr que  $\mathbf{s} = 0$ . Ahora, sea uno el grado relativo del sistema y definiendo  $y_1 = \mathbf{s}$   $y_2 = \dot{\mathbf{s}}$ , el objetivo de control es equivalente a estabilizar el sistema siguiente

$$\begin{aligned} \dot{y}_1 &= y_2 \\ \dot{y}_2 &= \vartheta(\mathbf{t}, \mathbf{x}) + \eta(\mathbf{t}, \mathbf{x})\dot{\mathbf{u}} \end{aligned} \quad (2.4.23)$$

donde  $y_2$  no es medible pero con la posibilidad de saber su signo. La ecuación (2.4.23) define la dinámica de la superficie deslizante  $\mathbf{s}$ , y al definir dicha dinámica por medio del control  $\mathbf{u}$  se define también el comportamiento de (2.4.22) respecto a  $\mathbf{s}$ .

Ahora, si  $\vartheta(\mathbf{t}, \mathbf{x}), \eta(\mathbf{t}, \mathbf{x})$  son funciones desconocidas pero que cumplen con las condiciones

$$\begin{aligned} \Phi &> 0, \\ |\vartheta(\mathbf{t}, \mathbf{x})| &\leq \Phi, \\ 0 < \Gamma_m &\leq \eta(\mathbf{t}, \mathbf{x}) \leq \Gamma_M, \end{aligned}$$

entonces, el algoritmo de control se define como



$$\begin{aligned}
\mathbf{u} &= \mathbf{u}_1(t) + \mathbf{u}_2(t) \\
\mathbf{u}_1(t) &= -W_1 |y_1|^p \text{sign}(y_1) \\
\dot{\mathbf{u}}_2(t) &= -W_2 \text{sign}(y_1)
\end{aligned} \tag{2.4.24}$$

donde  $W_1$  y  $W_2$  son ganancias de diseño del controlador, y las condiciones de estabilidad suficientes para (2.4.23) en lazo cerrado son

$$0 < \rho \leq 0.5 \quad ; \quad W_1^2 \geq \frac{4\Phi\Gamma_m(W_2 + \Phi)}{\Gamma_m^3(W_2 - \Phi)} \quad ; \quad W_2 > \frac{\Phi}{\Gamma_m}$$

Este algoritmo es ampliamente usado en sustitución de controles discontinuos como en [34] y [35]. Las propiedades del algoritmo ST en el dominio de la frecuencia son estudiadas en [36] y [37]. Este tipo de control también ha sido usado para diseñar diferenciadores exactos robustos [38] y observadores [39; 40; 41], suministrando su característica de convergencia en tiempo finito en presencia de entradas desconocidas.

## 2.5. Visión Computacional

El uso de un sistema de visión permite la percepción de información importante del ambiente de trabajo de un robot. El procesamiento de dicha información puede ser de gran ayuda en la interacción del robot con su ambiente. En este capítulo se describen los modelos y algoritmos basados en visión computacional que se utilizaron en este trabajo.

### 2.5.1. Cámara *Pinhole*

El modelo de *cámara estenopeica* o de *pinhole* es el más común en la literatura, y su descripción utilizando geometría proyectiva se puede encontrar en [42]. Este modelo describe las relaciones matemáticas entre las coordenadas de un punto en  $\mathbb{R}^3$  y su proyección sobre el plano de la imagen de la cámara, donde la apertura de la cámara se representa por un hueco de alfiler o *pinhole* como se puede observar en la Figura 21. Por simplicidad, tomaremos el origen del sistema de coordenadas de la cámara en el *centro* de la cámara  $\mathbf{c}$ , el plano de la imagen se encuentra a una distancia  $f$  del centro, y la línea  $L$  normal al plano de la imagen que pasa por el centro de la cámara se conoce como el *eje principal* de la cámara.

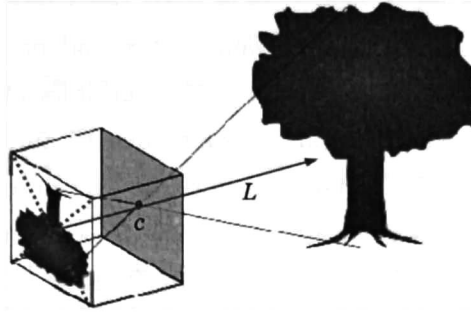


Figura 21. Cámara *pinhole*

Entonces, un punto  $X = (x, y, z)^T$  en la escena es mapeado a la intersección entre el plano de la imagen y la línea formada por  $X$  y  $c$ , dicha intersección equivale al punto  $x = (f_x/z, f_y/z)^T$  en las coordenadas de la imagen. Este mapeo se puede expresar en coordenadas homogéneas como

$$x = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ z \end{bmatrix}. \quad (2.5.1)$$

Lo anterior representa el caso ideal del modelo, es decir, cuando el centro de coordenadas coincide con el centro de la cámara. En el caso real existe una traslación  $t$  y una rotación  $R$  del centro de la cámara respecto al sistema de coordenadas. Además, el sensor de la cámara (usualmente un arreglo de CCD's) no es totalmente plano, ni cuadrado. Estos factores se contemplan en una matriz  $K$  conocida como *matriz de parámetros internos* de la cámara. Esto nos lleva a obtener la matriz de proyección de la cámara de la siguiente manera

$$P = K[R | t]. \quad (2.5.2)$$

Al proceso de encontrar la matriz  $K$  se le conoce como *calibración* de la cámara [43].

### 2.5.2. Segmentación HSV

Un espacio de color es un modelo bajo el cual se define la composición de los colores. La composición se determina por el valor de los vectores base del espacio de color, y la combinación lineal de los vectores base genera todo el espacio de

El contar con un SVE permite percibir profundidad aparte de color, textura, flujo óptico y otras propiedades. En otras palabras, provee la capacidad de percibir una dimensión más que solo con una cámara. La desventaja de un SVE es que es costoso, desde el punto de vista computacional, el procesar en tiempo real las imágenes. Además de calibrar cada cámara del SVE (parámetros intrínsecos), es necesario encontrar la transformación entre el sistema coordenadas de la cámara derecha respecto al sistema de coordenadas de la cámara izquierda. A los parámetros de dicha transformación se les conoce como *parámetros extrínsecos* del SVE. Al proceso se le conoce como *calibración del sistema estéreo* y puede ser consultado en [43]. Una vez obtenidos los parámetros intrínsecos y extrínsecos, se puede obtener la profundidad de un punto ubicado en ambas imágenes por medio de un proceso de triangulación.

Este proceso consiste en determinar la posición en 3D de un punto  $X$  usando la proyección del punto en la imagen derecha ( $x_d$ ) y en la izquierda ( $x_l$ ). Con la información de los centros de las cámaras ( $c_d$  y  $c_l$ ) y las proyecciones se puede obtener un triángulo, cuyas relaciones trigonométricas nos permitan obtener el punto  $X$ . Este proceso se ilustra en la parte izquierda de la Figura 24.

Sin embargo, en un SVE real existe ruido en las proyecciones lo que ocasiona que los rayos ópticos no se intersequen. Una forma de solucionar este problema es encontrar la distancia dirigida entre los dos rayos de las proyecciones y obtener el punto medio entre el par de puntos (Figura 24 derecha). Este procedimiento puede ser encontrado a detalle en [42].

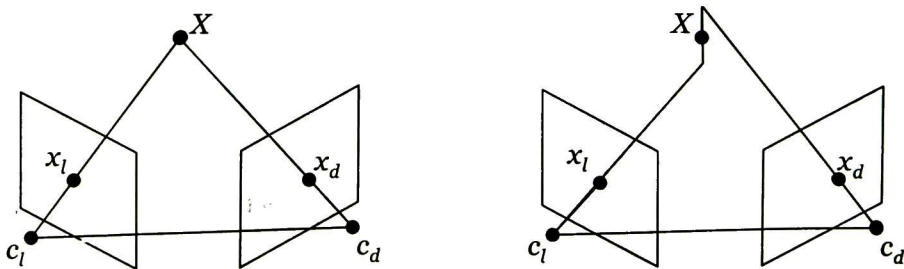


Figura 24. Triangulación: Ideal (Izquierda) y Real (Derecha).

### 2.5.4. Calibración Mano-Ojo

El problema de calibración Mano-Ojo consiste en encontrar la transformación  $X$  entre el efector final de un manipulador y el sistema de coordenadas de una cámara  $c_1$ , la cual se encuentra fijamente montada al manipulador. Se resuelve mediante el cómputo simultáneo de dos transformaciones espaciales en un lazo cerrado de transformaciones del mismo tipo. La Figura 25 muestra el escenario de este tipo de calibración. Se puede apreciar un manipulador en dos posiciones diferentes con un SVE montado en su último eslabón. El superíndice de los elementos en la figura indica la posición del manipulador. También se observa un objeto de referencia (un tablero de ajedrez) que nos permitirá cerrar el lazo de transformaciones.

La transformación  $A^i$  es la transformación entre la cámara y la referencia en la posición  $i$  del manipulador, y se obtiene por algoritmos de procesamiento de imágenes del SVE. La transformación  $B^i$  es la transformación entre la base del manipulador y su último eslabón en la posición  $i$  del manipulador, y se obtiene por cálculos de cinemática directa.

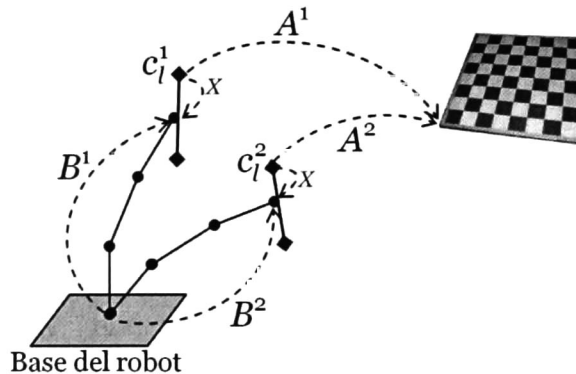


Figura 25. Calibración Mano-Ojo.

Entonces, de la figura podemos obtener la siguiente ecuación

$$AX = XB \tag{2.5.6}$$

donde  $A = A^2(A^1)^{-1}$  y  $B = (B^2)^{-1}B^1$ .

Existen muchos métodos ampliamente estudiados para resolver la ecuación (2.5.6), los cuales se pueden dividir en dos grupos. El grupo del método *clásico* se

basa en resolver por separado la parte rotacional y traslacional de la ecuación, ver [45], y el otro grupo resuelve ambas partes simultáneamente, ver [46]. En este trabajo utilizaremos una técnica desarrollada bajo el segundo esquema y en el marco del álgebra geométrica conformal [47].

### 3. Propuesta de Solución

Un esquema general para un sistema de visión estéreo se muestra en la Figura 26, donde  $T_p$  es la posición del objetivo,  $A$  es la orientación actual del SVE,  $O_b$  es el origen del sistema de coordenadas base y  $O_c$  es el origen del sistema de coordenadas de la cámara, convenientemente fijado al último eslabón de la cadena cinemática del manipulador.

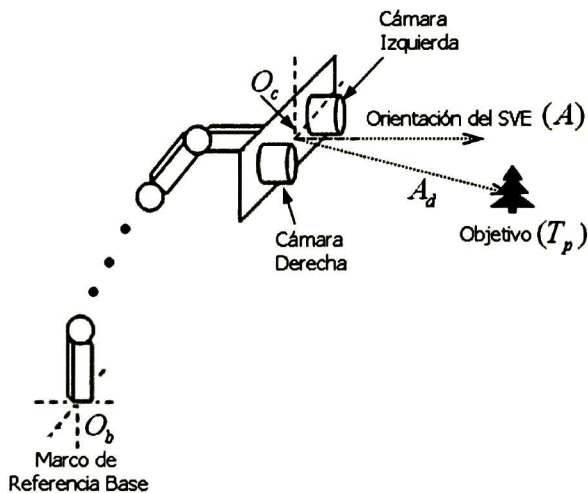


Figura 26. Manipulador Asistido por Visión y el Objetivo.

El modelo cinemático de esta estructura puede ser definido como

$$A = f(\theta) \quad (3.0.1)$$

donde  $A \in \mathbb{R}^3$   $\theta = [\theta_1, \dots, \theta_n]^T$  es el vector de posiciones angulares de las uniones y el vector  $f(\theta)$  está definido por la cinemática directa del manipulador.

### 3.1. Localización del Objetivo

La localización del objetivo es obtener un vector en 3D que relacione la posición del objetivo  $T_p$  con el origen del sistema de coordenadas del sistema estéreo  $O_c$ . Para lograr esto, primero se calcula el vector entre  $T_p$  y el centro de una de las cámaras (usualmente la izquierda  $C_I$ ) al cual definimos como  $CT$ . Luego, se calcula el vector entre  $O_c$  y  $C_I$ , el cual es obtenido por la calibración Mano-Ojo [46] y llamaremos  $OC$ . Dichos vectores se pueden apreciar en la Figura 27.

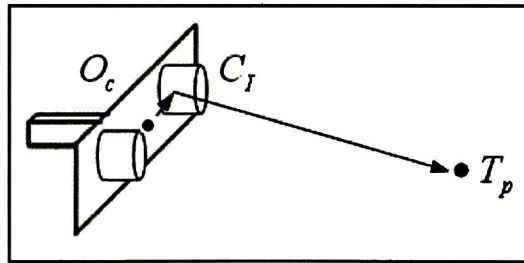


Figura 27. Calibración Mano-Ojo de un SVE.

Finalmente, se suman  $OC$  y  $CT$  para obtener la posición en 3D de  $T_p$  con respecto al marco de referencia del sistema estéreo.

#### 3.1.1. Calibración del Sistema de Visión Estéreo (SVE)

Para poder obtener  $CT$ , primero deben calibrarse las cámaras del SVE. **Calibrar** una cámara es el proceso mediante el cual se conocen los *parámetros intrínsecos* de la cámara, para más detalle vea [43].

Con ambas cámaras ya calibradas, se procede a calibrar el sistema estéreo. Esto es, obtener la transformación entre un sistema de coordenadas fijo en el centro de

la cámara izquierda y otro fijo en el centro de la cámara derecha (*parámetros extrínsecos*); de tal forma que podamos relacionar la posición de un objetivo en las imágenes de ambas cámaras.

### 3.1.2. Segmentación por Color

La posición en 2D de un objetivo en una imagen se puede obtener segmentando el objeto en dicha imagen. Para segmentar un objetivo de un color determinado, se compara dicho parámetro con los parámetros de cada uno de los píxeles en la imagen, de tal forma que si son iguales o similares se les considera como parte del objetivo.

El color usualmente se representa con un vector de 3 coordenadas, las cuales representan su posición dentro de algún espacio de color. El espacio de color más conocido e importante es el RGB (Red-Green-Blue), el cual es un espacio de color aditivo. Los colores rojo, verde y azul fueron escogidos por su similitud con los tres conos sensitivos del ojo humano. La principal desventaja de este espacio es que es muy sensible a los cambios de iluminación, por lo que complica los algoritmos de segmentación de color. Un espacio más robusto a este factor es el HSV (Hue-Saturation-Value) donde: *Hue* (Tonalidad) está definida por la longitud de onda dominante en un color, *Saturation* (Saturación) se refiere al grado de pureza de un color y *Value* (Valor o Brillo) es la cantidad blanco que contiene un color, o su luminosidad. Existe un mapeo del espacio RGB al HSV, para más detalles sobre esto vea [44].



Figura 28. Segmentación en HSV para un SVE



Un ejemplo de segmentación por color en HSV se puede apreciar en la Figura 28. Cada imagen corresponde a cada una de las cámaras en el SVE; podemos apreciar que el objeto segmentado (en este caso un taladro), ha sido localizado en la imagen y encerrado en un cuadro naranja.

Después de haber localizado el objetivo en ambas imágenes se puede relacionar dicha información para triangular la posición en 3D del objetivo, procedimiento que se explica a continuación.

### 3.1.3. Triangulación

Una vez calibrado el SVE, se puede obtener la posición en 3D de un punto en el espacio, si conocemos la correspondencia entre la proyección en 2D de ese punto sobre ambas imágenes del sistema estéreo. Esto se logra retro-proyectando cada uno de los puntos en las imágenes como rayos que parten del centro de cada cámara e intersecan el plano de la imagen en tal punto, por lo tanto esos rayos se intersecarán en el punto en 3D. Para más información consulte [42].

## 3.2. Modelado Propuesto Usando Álgebra Geométrica

En esta sección se tratará en detalle la obtención de modelos cinemáticos y dinámico de manipuladores seriales en tiempo continuo y tiempo discreto. Se usará álgebra geométrica  $G_{4,1}$  para ambos casos, la cual, como ya vimos en secciones anteriores, ofrece la ventaja de ser un método sencillo para representar transformaciones como rotaciones, traslaciones, reflexiones, entre otras. Además, la formulación en álgebra geométrica facilita los cálculos algebraicos.

### 3.2.1. Modelo de Cinemática Directa

La cinemática directa de un robot manipulador, consiste en calcular la pose (posición y orientación) del efector final usando las variables de unión (posiciones). Para esto, se necesita encontrar las transformaciones definidas por las uniones, i.e. se necesita modelar las uniones del robot. Como se mencionó anteriormente, las uniones básicas de las que se componen los robots son unión rotativa y unión

prismática, y estas se pueden modelar definiendo un *motor* por unión de la siguiente manera:

$$M_i = e^{\frac{1}{2}q_i D_i} \quad (3.2.1)$$

donde  $q_i$  es la  $i^{\text{ésima}}$  variable de unión (ángulo de rotación  $\theta$  para rotativas / distancia de traslación  $d$  para prismáticas),  $D_i$  es la entidad que define el motor ( $D=L$  es la línea que define el eje de rotación en *rotativas* y  $D=t e_\infty$  define la dirección de traslación en *prismáticas*).

Supongamos que definimos la *posición del efector final* de un manipulador serial de  $n$  GDL con el punto  $x_{ef}$  y su *orientación* con la línea  $L_{ef}$ , entonces la pose inicial del efector final está dada por

$$X = \begin{bmatrix} x_{ef} \\ L_{ef} \end{bmatrix}. \quad (3.2.2)$$

La cinemática directa de dicho manipulador (con  $n$  uniones) se define como

$$\begin{aligned} X' &= M_1 \cdots M_n X \widetilde{M}_n \cdots \widetilde{M}_1 \\ &= \left( \prod_{i=1}^n M_i \right) X \left( \prod_{i=1}^n \widetilde{M}_{n-i+1} \right), \end{aligned} \quad (3.2.3)$$

donde  $X$  es la pose inicial,  $X'$  es la pose actual del efector final.

Ahora, los motores que definen dicho modelo dependen de las variables de unión  $q_i$  y de las entidades  $D_i$  (líneas  $L_i$  ó direcciones  $t_i$ ), que los definen de acuerdo a (3.2.1). Dichas líneas y direcciones también son funciones de las variables de unión, ya que la variable de unión  $q_i$  afecta a los eslabones posteriores en la cadena cinemática, i.e. a las entidades  $D_j = L_j$  ó  $t_j e_\infty$  para  $j=i+1, \dots, n$ . Se usará un apóstrofe para distinguir a las entidades iniciales  $D_j$  de las actuales  $D'_j$ .

Estas relaciones se pueden definir mediante las siguientes ecuaciones:

- 1) Para la primera unión  $D'_1 = D_1$  y  $M_1$  se obtiene de (3.2.1).
- 2) Para la  $i^{\text{ésima}}$  unión ( $i=2, \dots, n$ )

$$D'_i = \left( \prod_{j=1}^{i-1} M_j \right) D_i \left( \prod_{j=1}^{i-1} \widetilde{M}_{i-j} \right) \quad (3.2.4)$$

y  $M_i$  se obtiene de (3.2.1).

Con el fin de ilustrar el proceso explicado, consideremos el manipulador serial de 6 GDL de la marca AdeptSix600 (que se encuentra en el laboratorio del

CINVESTAV. Unidad Gdl.) mostrado en la Figura 29. Las seis uniones de las que se conforma el manipulador son rotativas, por lo tanto  $D_i = L_i$  son los ejes de rotación iniciales de las uniones para  $i = 1, \dots, 6$ .

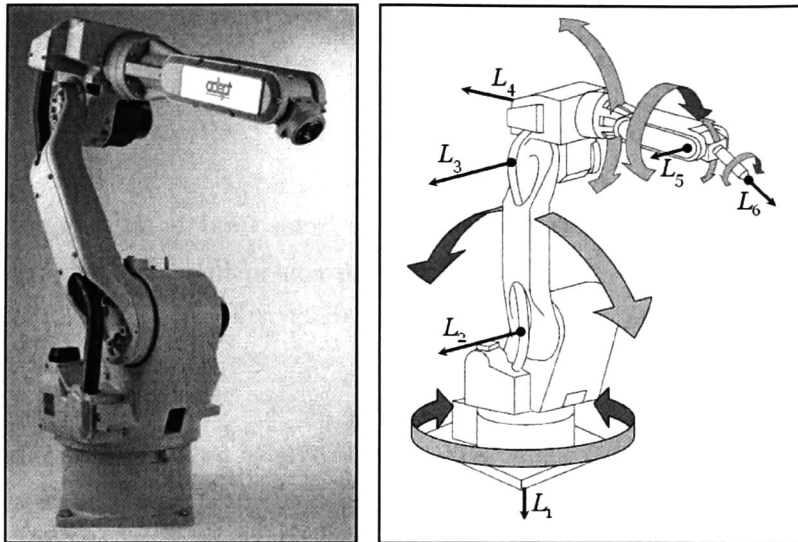


Figura 29. Manipulador serial AdeptSix600 y sus ejes de rotación.

Entonces, podemos definir a los ejes de rotación actuales y a los motores del manipulador como

$$\begin{aligned} L'_1 &= L_1 & \Rightarrow M_1 &= e^{-L'_1 q_1/2} \\ L'_2 &= M_1 L_2 \tilde{M}_1 & \Rightarrow M_2 &= e^{-L'_2 q_2/2} \\ &\vdots & &\vdots \end{aligned} \quad (3.2.5)$$

$$L'_6 = M_1 \cdots M_5 L_6 \tilde{M}_5 \cdots \tilde{M}_1 \Rightarrow M_6 = e^{-L'_6 q_6/2}$$

y la cinemática directa del manipulador como

$$X' = M_1 \cdots M_6 X \tilde{M}_6 \cdots \tilde{M}_1. \quad (3.2.6)$$

Para una explicación más detallada sobre el uso de ejes y motores para modelar mecanismos robóticos véase [32].

### 3.2.2. Modelo de Cinemática Diferencial

La cinemática diferencial de un robot manipulador relaciona la velocidad lineal y angular del efector final con la derivada de las variables de unión (velocidades de

unión). La cinemática diferencial se puede obtener derivando la ecuación de cinemática directa, la cual está dada por

$$\dot{X}' = \left( \prod_{i=1}^n M_i \right) X \left( \prod_{i=1}^n \tilde{M}_{n-i+1} \right). \quad (3.2.7)$$

La diferenciación total directa de (3.2.7) resulta en

$$\dot{X}' = \sum_{j=1}^n \partial_{q_j} \left( \prod_{i=1}^n M_i X \prod_{i=1}^n \tilde{M}_{n-i+1} \right) dq_j. \quad (3.2.8)$$

Por propiedades de la derivada de un producto, se puede re-definir a (3.2.8) como

$$\begin{aligned} \dot{X}' = & \sum_{j=1}^n \left[ \partial_{q_j} \left( \prod_{i=1}^j M_i \right) \prod_{i=j+1}^n M_i X \prod_{i=1}^n \tilde{M}_{n-i+1} \right. \\ & \left. + \prod_{i=1}^n M_i X \prod_{i=1}^{n-j} \tilde{M}_{n-i+1} \partial_{q_j} \left( \prod_{i=n-j+1}^n \tilde{M}_{n-i+1} \right) \right] \end{aligned} \quad (3.2.9)$$

Debido a su forma exponencial, la derivada del motor  $M = e^{\frac{1}{2}qD}$  y la de su reversión  $\tilde{M} = e^{\frac{1}{2}qD}$ , se pueden definir como

$$\frac{d(M)}{dt} = -\frac{1}{2}MD\dot{q}, \quad \frac{d(\tilde{M})}{dt} = \frac{1}{2}\tilde{M}D\dot{q}. \quad (3.2.10)$$

Entonces, podemos obtener la derivada del producto de motores como

$$\partial_{q_j} \left( \prod_{i=1}^j M_i \right) = -\frac{1}{2} \prod_{i=1}^j M_i D_j \quad (3.2.11)$$

y

$$\partial_{q_j} \left( \prod_{i=n-j+1}^n \tilde{M}_{n-i+1} \right) = \frac{1}{2} \left( \prod_{i=n-j+1}^n \tilde{M}_{n-i+1} \right) D_j. \quad (3.2.12)$$

Sustituyendo (3.2.11) y (3.2.12) en (3.2.9) se obtiene

$$\begin{aligned} \dot{X}' = & \sum_{j=1}^n \left[ -\frac{1}{2} \left( \prod_{i=1}^j M_i \right) D_j \prod_{i=j+1}^n M_i X \prod_{i=1}^n \tilde{M}_{n-i+1} \right. \\ & \left. + \frac{1}{2} \prod_{i=1}^n M_i X \prod_{i=1}^{n-j} \tilde{M}_{n-i+1} \left( \prod_{i=n-j+1}^n \tilde{M}_{n-i+1} \right) D_j \right] \dot{q}_j. \end{aligned} \quad (3.2.13)$$

Usando la igualdad  $\prod_{i=n-j+1}^n \tilde{M}_{n-i+1} \prod_{i=1}^j M_i = 1$  y sabiendo que los bivectores conmutan podemos re-escribir a (3.2.13) de la forma

$$\begin{aligned} \dot{X}' = & -\frac{1}{2} \sum_{j=1}^n \left[ \left( \prod_{i=1}^j M_i D_j \prod_{i=n-j+1}^n \tilde{M}_{n-i+1} \right) \left( \prod_{i=1}^j M_i \prod_{i=j+1}^n M_i X \prod_{i=1}^n \tilde{M}_{n-i+1} \right) \right. \\ & \left. - \left( \prod_{i=1}^n M_i X \prod_{i=1}^{n-j} \tilde{M}_{n-i+1} \prod_{i=n-j+1}^n \tilde{M}_{n-i+1} \right) \left( \prod_{i=1}^j M_i D_j \prod_{i=n-j+1}^n \tilde{M}_{n-i+1} \right) \right] \dot{q}_j. \end{aligned} \quad (3.2.14)$$

Si agrupamos términos en (3.2.14) nos queda

$$\begin{aligned} \dot{X}' = & -\frac{1}{2} \sum_{j=1}^n \left[ \left( \prod_{i=1}^j M_i D_j \prod_{i=n-j+1}^n \tilde{M}_{n-i+1} \right) \left( \prod_{i=1}^n M_i X \prod_{i=1}^n \tilde{M}_{n-i+1} \right) \right. \\ & \left. - \left( \prod_{i=1}^n M_i X \prod_{i=1}^n \tilde{M}_{n-i+1} \right) \left( \prod_{i=1}^j M_i D_j \prod_{i=n-j+1}^n \tilde{M}_{n-i+1} \right) \right] \dot{q}_j, \end{aligned} \quad (3.2.15)$$

y, usando la ecuación de cinemática directa (3.2.7) y la igualdad (3.2.4), la ecuación (3.2.15) resulta en

$$\dot{X}' = \frac{1}{2} \sum_{j=1}^n [X' D'_j - D'_j X'] \dot{q}_j \quad (3.2.16)$$

la cual define la cinemática diferencial de un manipulador serial de  $n$  grados de libertad con cualquier tipo de uniones. La forma clásica de representar (3.2.16) es

$$\dot{X}' = J(q) \dot{q} \quad (3.2.17)$$

donde

$$J(q) = [\alpha_1 \quad \dots \quad \alpha_n] \text{ con } \alpha_j = \frac{1}{2} (X' D'_j - D'_j X'), \text{ y } q = [q_1 \quad \dots \quad q_n]^T$$

Si se deseara modelar solo la posición del efector final  $x_{ef}$ , se podría obtener una cinemática diferencial un poco más simplificada usando la propiedad (2.1.9) para obtener

$$\dot{x}_{ef}' = J_x(q) \dot{q}, \quad J_x(q) = [x'_{ef} \cdot D'_1 \quad \dots \quad x'_{ef} \cdot D'_n]. \quad (3.2.18)$$

Con fines de comparación, re-escribiremos la ecuación de cinemática diferencial obtenida con el método clásico definida en (2.2.3) como

$$\begin{pmatrix} \dot{p} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} \xi_1 a_1 + \bar{\xi}_1 (a_1 \times p_{1n}) & \dots & \xi_n a_n + \bar{\xi}_n (a_n \times p_{nn}) \\ \bar{\xi}_1 a_1 & \dots & \bar{\xi}_n a_n \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{pmatrix} = J(q) \dot{q}. \quad (3.2.19)$$

Se puede observar que la ecuación (3.2.16) obtenida por medio de AGC es más simple y compacta que la obtenida por el método clásico definida en (3.2.19).

## Modelo Cinemático Diferencial Discreto

Si definimos las variables de estado de (3.2.17) como  $x_1 = q$ ,  $x_2 = \dot{X}$ , y el término de control como  $u = \dot{q}$ , se puede obtener una representación en espacio de estados de la forma

$$\begin{aligned}\dot{x}_1 &= u, & \dot{x}_2 &= J(x_1)u \\ y &= x_2,\end{aligned}\tag{3.2.20}$$

donde  $y$  es la salida del sistema. Ahora, si discretizamos (3.2.20) por el método de Euler resulta en el siguiente modelo discreto

$$\begin{aligned}x_{1,k+1} &= x_{1,k} + Tu_k \\ x_{2,k+1} &= x_{2,k} + TJ(x_{1,k})u_k \\ y_k &= x_{2,k}\end{aligned}\tag{3.2.21}$$

con  $x_{i,k} = x_i(kT)$  para  $i = \{1, 2\}$ , y  $T$  el período de muestreo.

Si agregamos un vector de perturbaciones  $d_k(x_{1,k}, k)$  debido a incertidumbre en el modelo y factores externos, condiciones comunes en modelos reales, obtenemos un modelo perturbado de (3.2.21) como

$$\begin{aligned}x_{1,k+1} &= x_{1,k} + Tu_k \\ x_{2,k+1} &= x_{2,k} + TJ(x_{1,k})u_k + d_k(x_{1,k}, k) \\ y_k &= x_{2,k}.\end{aligned}\tag{3.2.22}$$

### 3.2.3. Modelo Dinámico

En esta sección desarrollaremos el modelo dinámico del manipulador expresado en algebra geométrica conformal (AGC) usando las ecuaciones de Euler-LaGrange, basándonos en [32] y [48]. Además, extenderemos el modelo dinámico conformal para incluir uniones prismáticas en los manipuladores seriales.

El modelo general de la dinámica obtenido en (2.2.21), y repetido aquí por conveniencia, es

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau\tag{3.2.23}$$

La matriz de Inercias  $H(q)$  se puede expresar usando AGC de la siguiente manera

$$H(q) = M_v + M_l\tag{3.2.24}$$

donde  $M_v$  representa la energía cinética de la masa total de cada eslabón moviéndose respecto al sistema de coordenadas del mismo eslabón (i.e. con la

misma velocidad del centro de masas del eslabón), y  $M_I$  expresa la energía cinética del movimiento de las partículas que conforman a cada eslabón, respecto al centro de masas del mismo eslabón (i.e. momentos de inercia). Dichas matrices se definen como:

$$M_v = V^T m V \quad (3.2.25)$$

$$M_I = \delta I \quad (3.2.26)$$

con

$$V = \begin{bmatrix} x'_{c1} \cdot D'_1 & x'_{c2} \cdot D'_1 & \cdots & x'_{cn} \cdot D'_1 \\ 0 & x'_{c2} \cdot D'_2 & \cdots & x'_{cn} \cdot D'_2 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & x'_{cn} \cdot D'_n \end{bmatrix}, \quad m = \begin{bmatrix} m_1 & 0 & \cdots & 0 \\ 0 & m_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & m_n \end{bmatrix}$$

$$\delta = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \quad I = \begin{bmatrix} I_1 & 0 & \cdots & 0 \\ I_2 & I_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ I_n & I_n & \cdots & I_n \end{bmatrix}$$

donde  $x'_{ci}$  es la posición actual del centro de masas,  $I_i$  la inercia y  $D'_i$  la entidad que define el motor, todos pertenecientes al  $i^{\text{ésimo}}$  eslabón. Si recordamos que la inercia de un cuerpo rígido puede ser obtenida como  $I = m\ell_c^2$  donde  $\ell_c^2$  es la distancia del eje de rotación del cuerpo al centro de masas del mismo, entonces podemos expresar la inercia del  $j^{\text{ésimo}}$  eslabón como

$$I_j = m_j \ell_{c,j}^2, \quad (3.2.27)$$

y definir la matriz de inercias  $I$  como

$$I = mL_{cm} \quad (3.2.28)$$

donde

$$L_{cm} = \begin{bmatrix} \ell_{c,1}^2 & 0 & \cdots & 0 \\ \ell_{c,2}^2 & \ell_{c,2}^2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ \ell_{c,n}^2 & \ell_{c,n}^2 & \cdots & \ell_{c,n}^2 \end{bmatrix}.$$

La matriz de Fuerzas Centrifugas y de Coriolis  $C(\dot{q}, q)$  se puede obtener de la propiedad

$$\dot{H}(q) = C + C^T, \quad (3.2.29)$$

y de  $H(\mathbf{q}) = \mathbf{V}^T \mathbf{mV} + \delta I$ , la cual fue obtenida anteriormente. Derivando  $H(\mathbf{q})$  resulta en

$$\begin{aligned} \dot{H}(\mathbf{q}) &= \frac{d}{dt}(\mathbf{V}^T \mathbf{mV} + \delta I), \\ &= \frac{d}{dt}(\mathbf{V}^T \mathbf{mV}), \text{ dado que } \delta I = \text{cte.} \\ &= \mathbf{V}^T \dot{\mathbf{mV}} + \dot{\mathbf{V}}^T \mathbf{mV}, \\ &= \mathbf{V}^T \dot{\mathbf{mV}} + (\mathbf{V}^T \dot{\mathbf{mV}})^T \end{aligned} \quad (3.2.30)$$

Entonces, de (3.2.29) y (3.2.30) se obtiene que

$$\mathbf{C}(\dot{\mathbf{q}}, \mathbf{q}) = \mathbf{V}^T \dot{\mathbf{mV}} \quad (3.2.31)$$

donde  $\dot{\mathbf{V}}$  es la diferenciación directa de  $\mathbf{V}$  y su obtención se explicará enseguida.

Es posible descomponer a la matriz  $\mathbf{V}$  como

$$\mathbf{V} = \mathbf{Y} \cdot \mathbf{D},$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{x}'_{c1} & 0 & \cdots & 0 \\ 0 & \mathbf{x}'_{c2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{x}'_{cn} \end{bmatrix}, \mathbf{D} = \begin{bmatrix} \mathbf{D}'_1 & 0 & \cdots & 0 \\ \mathbf{D}'_2 & \mathbf{D}'_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{D}'_r & \mathbf{D}'_r & \cdots & \mathbf{D}'_n \end{bmatrix}. \quad (3.2.32)$$

Entonces, derivando se obtiene

$$\dot{\mathbf{V}} = \dot{\mathbf{Y}} \cdot \mathbf{D} + \mathbf{Y} \cdot \dot{\mathbf{D}}, \quad (3.2.33)$$

donde

$$\dot{\mathbf{Y}} = \begin{bmatrix} \dot{\mathbf{x}}'_{c1} & 0 & \cdots & 0 \\ 0 & \dot{\mathbf{x}}'_{c2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dot{\mathbf{x}}'_{cn} \end{bmatrix}, \dot{\mathbf{D}} = \begin{bmatrix} \dot{\mathbf{D}}'_1 & 0 & \cdots & 0 \\ \dot{\mathbf{D}}'_2 & \dot{\mathbf{D}}'_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \dot{\mathbf{D}}'_r & \dot{\mathbf{D}}'_r & \cdots & \dot{\mathbf{D}}'_n \end{bmatrix}. \quad (3.2.34)$$

Los términos  $\dot{\mathbf{x}}'_{ci}$  se pueden obtener por la ecuación de cinemática diferencial para posición (3.2.18) vista en la sección anterior, por lo que obtenemos

$$\begin{bmatrix} \dot{\mathbf{x}}'_{c1} \\ \vdots \\ \dot{\mathbf{x}}'_{cn} \end{bmatrix} = \mathbf{Y} \cdot \mathbf{D}\dot{\mathbf{q}} = \mathbf{V}\dot{\mathbf{q}}. \quad (3.2.35)$$

Por otro lado,  $\dot{\mathbf{D}}_i$  representa la velocidad del eje  $i$  generada por el movimiento (rotación y/o traslación) de los ejes que le anteceden. Estos se pueden calcular usando el mismo procedimiento que se usó para obtener (3.2.16), resultando en



$$\begin{aligned} \begin{bmatrix} \dot{D}'_1 \\ \dot{D}'_2 \\ \vdots \\ \dot{D}'_n \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} (D'_1 D'_1 & 0 & \cdots & 0 \\ D'_2 D'_1 & D'_2 D'_2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ D'_n D'_1 & D'_n D'_2 & \cdots & D'_n D'_n \end{bmatrix} \\ &- \begin{bmatrix} D'_1 D'_1 & 0 & \cdots & 0 \\ D'_1 D'_2 & D'_2 D'_2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ D'_1 D'_r & D'_2 D'_n & \cdots & D'_n D'_n \end{bmatrix} \dot{q} \end{aligned} \quad (3.2.36)$$

La matriz de gravedad  $g(q)$  está definida como

$$G = \frac{\partial U}{\partial q} = \frac{\partial}{\partial q} \left( \sum_{i=1}^n F_i \cdot x'_{ci} \right) = \sum_{i=1}^n F_i \cdot \frac{\partial}{\partial q} (x'_{ci}) \quad (3.2.37)$$

donde  $F_i = m_i \hat{g}$  es fuerza de gravedad ejercida sobre el  $i^{\text{ésimo}}$  eslabón, y como no depende de  $q$  podemos extraerla de la derivada. Entonces, por la cinemática diferencial se puede obtener la derivada parcial como

$$\frac{\partial}{\partial q} (x'_{ci}) = \begin{bmatrix} x'_{ci} \cdot D'_1 \\ \vdots \\ x'_{ci} \cdot D'_i \end{bmatrix}, \quad (3.2.38)$$

y podemos desarrollar la sumatoria en (3.2.37) para obtener

$$G = \begin{bmatrix} x'_{c1} \cdot D'_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} F_1 + \begin{bmatrix} x'_{c2} \cdot D'_1 \\ x'_{c2} \cdot D'_2 \\ \vdots \\ 0 \end{bmatrix} F_2 + \cdots + \begin{bmatrix} x'_{cn} \cdot D'_1 \\ x'_{cn} \cdot D'_2 \\ \vdots \\ x'_{cn} \cdot D'_n \end{bmatrix} F_n \quad (3.2.39)$$

o en forma matricial

$$G = \begin{bmatrix} x'_{c1} \cdot D'_1 & x'_{c2} \cdot D'_1 & \cdots & x'_{cn} \cdot D'_1 \\ 0 & x'_{c2} \cdot D'_2 & \cdots & x'_{cn} \cdot D'_2 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & x'_{cn} \cdot D'_n \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{bmatrix} = V^T F, \quad (3.2.40)$$

y si expresamos

$$F = \begin{bmatrix} m_1 & 0 & \cdots & 0 \\ 0 & m_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & m_n \end{bmatrix} \begin{bmatrix} \hat{g} \\ \hat{g} \\ \vdots \\ \hat{g} \end{bmatrix} = m a, \quad (3.2.41)$$

podemos definir



Todas las entidades y medidas de la Figura 30 se definen respecto al sistema de coordenadas dado por los vectores ortonormales  $\mathbf{e}_1$ ,  $\mathbf{e}_2$  y  $\mathbf{e}_3$ .  $L_i$  es el eje de rotación y  $q_i$  la posición angular de la  $i^{\text{ésima}}$  unión,  $A$  es la orientación del SVE y  $\mathbf{p}_t$  es el objetivo a seguir, cuyas coordenadas euclidianas son  $\mathbf{p}_t = x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3$ .

Entonces, podemos obtener las relaciones de cinemática directa del sistema como

$$\begin{aligned} L_{1,0} &= \mathbf{e}_{12} \rightarrow L_1 = L_{1,0} \\ L_{2,0} &= \mathbf{e}_{23} \rightarrow L_2 = M_1 L_{2,0} \tilde{M}_1 \\ A_0 &= \mathbf{e}_{31} + \ell \mathbf{e}_3 \mathbf{e}_\infty \rightarrow A = M_2 M_1 A_0 \tilde{M}_1 \tilde{M}_2 \\ \mathbf{x}_{\text{ef},0} &= \ell \mathbf{e}_3 + 0.5\ell^2 \mathbf{e}_\infty + \mathbf{e}_0 \rightarrow \mathbf{x}_{\text{ef}} = M_2 M_1 \mathbf{x}_{\text{ef},0} \tilde{M}_1 \tilde{M}_2 \end{aligned} \tag{3.2.44}$$

donde  $M_i = e^{-\frac{1}{2}L_i q_i}$  para  $i = \{1, 2\}$ . Este mecanismo se usará en la implementación en tiempo real de los controladores discretos propuestos en esta tesis.

Supongamos que deseamos obtener las variables de unión ( $q_1$  y  $q_2$ ) necesarias para que el SVE enfoque al objetivo  $\mathbf{p}_t$ , i.e. que  $\mathbf{p}_t$  yacza sobre la línea definida por  $A$ . Ahora, sabemos que una vez enfocado el objetivo,  $A$  puede definirse en su forma dual como

$$A^* = \mathbf{x}_{\text{ef}} \wedge \mathbf{p}_t. \tag{3.2.45}$$

Entonces, usando (3.2.45) y (3.2.44) podemos obtener la igualdad

$$\left[ \left( M \mathbf{x}_{\text{ef},0} \tilde{M} \right) \wedge \mathbf{p}_t \right] I_c = M A_0 \tilde{M}. \tag{3.2.46}$$

donde  $M = M_2 M_1$  e  $I_c = \mathbf{e}_1 \mathbf{e}_2 \mathbf{e}_3 \mathbf{e}_0 \mathbf{e}_\infty$  es el pseudo-escalar conformal. La forma general de la solución es equivalente a resolver la ecuación (3.2.46) para obtener  $q_1$  y  $q_2$ .

Una forma de simplificar el problema de la cinemática inversa consiste en proyectar el sistema sobre planos estratégicos, de tal forma que dividamos el problema global en sub-problemas más simples.

Por ejemplo, si proyectamos el sistema del caso anterior sobre el plano  $\pi_1 = \mathbf{e}_3$  (formado por  $\mathbf{e}_1$  y  $\mathbf{e}_2$ , i.e. una proyección desde arriba del sistema) obtenemos la figura de la izquierda. Entonces  $q_1$  se obtiene como

$$q_1 = \tan^{-1}(y/x) \tag{3.2.47}.$$

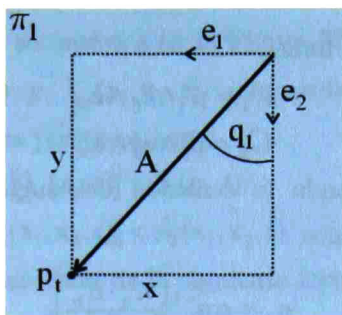


Figura 31. Proyección del sistema robótico en el plano  $\pi_1$ .

Ahora, para obtener  $q_2$  proyectamos el sistema sobre el plano definido por  $x_t$ ,  $x_{ef,0}$  y el origen ( $\pi_2 = [x_t \wedge x_{ef,0} \wedge e_0 \wedge e_\infty] I_c$ ) el cual es ortogonal a  $\pi_1$ , como se muestra en la Figura 32.

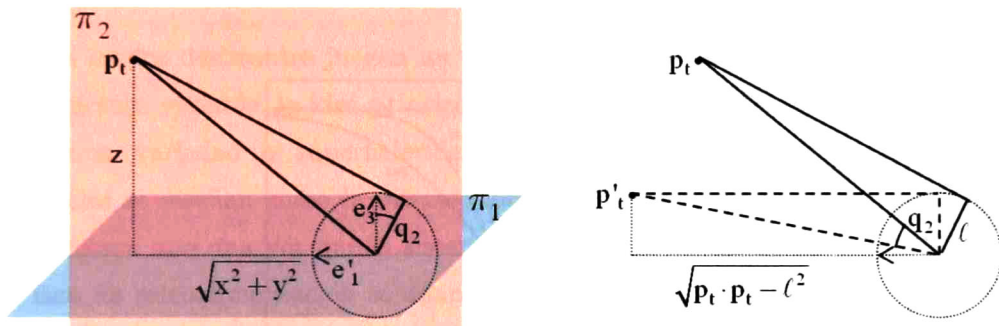


Figura 32. Proyección del sistema robótico en el plano  $\pi_2$ .

En la parte izquierda de la Figura 32 se muestra la proyección del sistema en el plano  $\pi_2$ , este es el plano de rotación para  $q_2$  (una vez que se ha aplicado la rotación  $q_1$ ) donde la parte euclidiana de  $p_t$  esta dada por  $p_{t,e} = (\sqrt{x^2 + y^2})e'_1 + ze_3$  y  $e'_1 = M_1 e_1 \tilde{M}_1$ . En la parte de la derecha se muestra otra interpretación del sistema proyectado, en la cual  $p_t$  es el resultado de aplicar  $M_2$  a  $p'_t$ , es decir

$$p_t = M_2 p'_t \tilde{M}_2, \quad (3.2.48)$$

de donde podemos obtener que la parte euclidiana de  $p'_t$  siempre cumple con

$$p'_{t,e} = (\sqrt{p_t \cdot p_t - \ell^2})e'_1 + \ell e_3$$

Si definimos las siguientes líneas

$$\begin{aligned} L_a^* &= p_t' \wedge e_0 \wedge e_\infty \\ L_b^* &= p_t \wedge e_0 \wedge e_\infty \end{aligned} \tag{3.2.49}$$

podemos obtener  $q_2$  usando la ecuación del ángulo entre dos líneas dada en (2.1.78), de donde obtenemos

$$q_2 = \cos^{-1} \left( \frac{L_a^* \cdot L_b^*}{|L_a^*| |L_b^*|} \right), \tag{3.2.50}$$

con lo que finalizamos la obtención de la cinemática inversa del mecanismo robótico. Este ejercicio, aunque simple, muestra las ventajas de usar AGC en la resolución de problemas geométricos debido a la simplicidad con la que representa entidades geométricas complejas y las relaciones entre ellas.

A continuación se procederá con el diseño de los algoritmos de control propuestos en este trabajo.

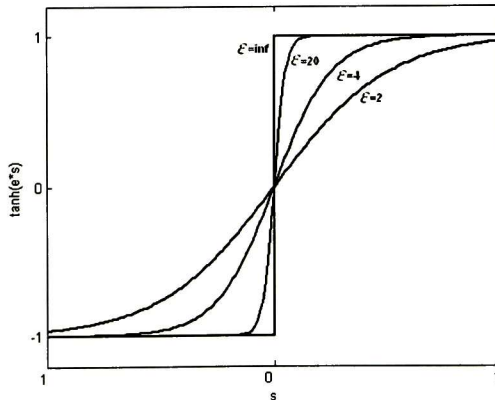


Figura 33. Función sigmoidea para varios valores de pendiente.

### 3.3. Suposiciones de Diseño

Antes de empezar a describir los controladores propuestos en este trabajo, definiremos las suposiciones que se tomarán en cuenta durante el diseño de los mismos. Estas se presentan a continuación:

S1) La función signo puede ser aproximada por una función sigmoidea de acuerdo a la siguiente ecuación

$$\lim_{\epsilon \rightarrow \infty} \text{sigm}(\epsilon S) = \text{sign}(S)$$

con  $\text{sigm}(\varepsilon S) = \tanh(\varepsilon S)$ , como se muestra en la Figura 33 para varios valores de  $\varepsilon$ .

S2) Los términos  $\lambda_u(x_1, t)$  y  $\lambda_m(x_1, x_2, t)$ , que serán definidos después, están acotados por funciones escalares positivas como

$$\begin{aligned} \|\lambda_u(x_1, t)\| &< \beta_1(x_1, t) \\ \|\lambda_m(x_1, x_2, t)\| &< \beta_2(x_1, x_2, t) \end{aligned}$$

y la derivada de  $\lambda_u(x_1, t)$  está acotada de la siguiente forma

$$\begin{aligned} \|\dot{\lambda}_u\| &\leq \beta_u \|\dot{s}_1\| \\ \beta_u &> 0 \end{aligned}$$

S3)  $\text{rank}[b_2(y_1)] = n$ , donde  $n$  denota el número de grados de libertad (GDL) del manipulador.

### 3.4. Control Robusto de Robots por Modos Deslizantes

Los modos deslizantes juegan un papel muy importante en la teoría de control de estructura variable, la idea principal es forzar la ocurrencia de modos deslizantes en alguna variedad o superficie del espacio de estados. Por lo general estas superficies se diseñan como la intersección de *hiperplanos* en el espacio de estados, de tal forma que una vez que el sistema ha alcanzado esta superficie, la estructura del lazo de retroalimentación se adapta para "deslizar" el estado del sistema sobre esta superficie, entonces la respuesta del sistema depende solo del gradiente de la superficie deslizante y permanece insensible a las perturbaciones que cumplen con la condición *matching*.

### 3.5. Control por Modos Deslizantes Integrales Anidados (MDIA)

Este es el primer algoritmo de control que se propone en este trabajo, para resolver el problema de seguimiento de trayectorias de las posiciones angulares en un manipulador robótico.

El control por MDIA reúne las ventajas de los modos deslizantes anidados (MDA) y los modos deslizantes integrales (MDI) para obtener una ley de control robusta a perturbaciones *matched* y *no matched*, que esté en modo deslizante (y

donde el término  $\lambda_m$ , el cual representa a las perturbaciones *matched* a rechazar del sistema, está dado por

$$\lambda_m = \dot{x}_{2,d} + \lambda(x_1, x_2, t). \quad (3.4.16)$$

Diseñando el control como  $u = u_0 + u_1$ , la dinámica de  $s_2$  queda

$$\dot{s}_2 = f_2(x_1, x_2) + b_2(x_1)u_0 + b_2(x_1)u_1 - \dot{x}_{2,c} + \dot{z}_2 + \lambda_m, \quad (3.4.17)$$

y eligiendo  $\dot{z}_2$  como

$$\dot{z}_2 = -f_2(x_1, x_2) - b_2(x_1)u_0 + \dot{x}_{2,c} \quad (3.4.18)$$

con  $z_2(0) = -e_2(0)$  (para cumplir  $s_2(0) = 0$  desde el instante inicial) y definiendo

$$u_0 = b_2(x_1)^{-1}(-f_2(x_1, x_2) + \dot{x}_{2,c} - c_2 e_2) \quad (3.4.19)$$

como una ley de control por retroalimentación del error donde  $c_2 > 0$ , podemos utilizar (3.4.18) y (3.4.19) para que la ecuación (3.4.17) quede

$$\dot{s}_2 = b_2(x_1)u_1 + \lambda_m, \quad (3.4.20)$$

y así poder definir el término de control  $u_1$  exclusivamente para rechazar  $\lambda_m$ . Esto lo lograremos al inducir modo deslizante en (3.4.20), para esto se elige

$$u_1 = -k_2 b_2(x_1)^{-1} \text{sign}(s_2) \quad (3.4.21)$$

con  $k_2 > 0$ . Resumiendo, la dinámica de  $s_1$  y  $s_2$  en lazo cerrado queda como

$$\begin{aligned} \dot{s}_1 &= -k_1 \text{sign}(\varepsilon_1 s_1) + \lambda_u \\ \dot{s}_2 &= -k_2 \text{sign}(s_2) + \lambda_m \end{aligned} \quad (3.4.22)$$

mientras que la dinámica de las variables de error en lazo cerrado son

$$\begin{aligned} \dot{e}_1 &= -c_1 e_1 + e_2 + x_{2,1} + \lambda_u \\ \dot{e}_2 &= -c_2 e_2 + b_2(x_1)u_1 + \lambda_m \end{aligned} \quad (3.4.23)$$

Después de eso, se definen las ganancias de control de la siguiente manera

$$\begin{aligned} k_1 &> \frac{\beta_1}{1 - \|\Delta(s_1, \varepsilon_1)\|}, \\ k_2 &> \beta_2, \end{aligned} \quad (3.4.24)$$

donde

$$\Delta(s_1, \varepsilon_1) = \text{sign}(s_1) - \text{sign}(\varepsilon_1 s_1), \quad (3.4.25)$$

finalizando así el diseño del controlador.

Las ganancias de control descritas en (3.4.24) se definieron para asegurar la estabilidad del sistema en lazo cerrado. Esto se analizará en más detalle a continuación.

## Análisis de Estabilidad del controlador por MDIA

Primero, debemos demostrar que la dinámica de las superficies en lazo cerrado es estable, para así corroborar que se rechazan las perturbaciones. Para esto definimos una función de Lyapunov para el segundo bloque de la siguiente forma (para más detalles véase Anexo A2):

$$V_2 = \frac{1}{2} s_2^T s_2. \quad (3.4.26)$$

Derivando (3.4.26) obtenemos  $\dot{V}_2 = s_2^T \dot{s}_2$ , entonces usando la ecuación (3.4.22) resulta en

$$\begin{aligned} \dot{V}_2 &= s_2^T (-k_2 \text{sign}(s_2) + \lambda_m) \\ &= -k_2 \|s_2\| + s_2^T \lambda_m \\ &< -k_2 \|s_2\| + \|\lambda_m\| \|s_2\|. \end{aligned} \quad (3.4.27)$$

Ahora, usando la suposición  $S2$  nos queda

$$\dot{V}_2 < \|s_2\| (-k_2 + \beta_2(x_1, x_2, t)). \quad (3.4.28)$$

Por lo tanto, para lograr que  $\dot{V}_2 < 0$ , y por consiguiente que  $s_2$  converge a cero en tiempo finito, se debe cumplir que  $k_2 > \beta_2(x_1, x_2, t)$ . Esta condición de estabilidad se expresa en (3.4.24). Además, debido a que  $z_2(0) = -e_2(0)$  los modos deslizantes ocurren desde el instante inicial.

Procediendo con el primer bloque, de forma similar al segundo bloque, definimos

$$V_1 = \frac{1}{2} s_1^T s_1 \quad (3.4.29)$$

entonces

$$\begin{aligned} \dot{V}_1 &= s_1^T \dot{s}_1 \\ &= s_1^T (-k_1 \text{sign}(\varepsilon_1 s_1) + \lambda_u) \end{aligned} \quad (3.4.30)$$

Por medio de la igualdad (3.4.25) obtenemos  $\dot{V}_1$  como

$$\begin{aligned} \dot{V}_1 &= s_1^T (-k_1 (\text{sign}(s_1) - \Delta(\varepsilon_1, s_1)) + \lambda_u) \\ &= -k_1 \|s_1\| + k_1 s_1^T \Delta(\varepsilon_1, s_1) + s_1^T \lambda_u \\ &< -k_1 \|s_1\| + k_1 \|s_1\| \|\Delta(\varepsilon_1, s_1)\| + \|s_1\| \|\lambda_u\| \\ &< \|s_1\| [-k_1 (1 - \|\Delta(\varepsilon_1, s_1)\|) + \|\lambda_u\|] \end{aligned} \quad (3.4.31)$$

y, usando la suposición  $S2$ , la derivada (3.4.31) se convierte en

$$\dot{V}_1 < \|s_1\| (-k_1 (1 - \|\Delta(\varepsilon_1, s_1)\|) + \beta_1(x_1, t)). \quad (3.4.32)$$



De la ecuación anterior podemos concluir que para cumplir  $\dot{V}_1 < 0$ , y asegurar la estabilidad del bloque, se necesita que  $k_1 > \beta_1(x_1, t) / (1 - \|\Delta(\varepsilon_1, s_1)\|)$ . Esta condición está expresada en (3.4.24).

Debido a que la condición de estabilidad depende de  $\|\Delta(\varepsilon_1, s_1)\|$ , y esta es una función de  $s_1$ , podemos concluir que  $s_1$  converge solo a una vecindad de 0, y que esta vecindad depende de los parámetros  $k_1$  y  $\varepsilon_1$ . Cabe señalar que si  $\varepsilon_1 = \infty$  entonces  $\text{sigm}(\varepsilon_1 s_1) = \text{sign}(s_1)$  y la vecindad de convergencia sería el origen, i.e.  $s_1 \rightarrow 0$ ; igual que en el segundo bloque.

Para calcular dicha vecindad de convergencia re-escribimos la condición de estabilidad y despejamos para obtener

$$k_1 > \frac{\beta_1(x_1, t)}{(1 - \|\Delta(\varepsilon_1, s_1)\|)}$$

$$k_1 (1 - \|\Delta(\varepsilon_1, s_1)\|) > \beta_1(x_1, t)$$

y debido a que  $k_1 > 0$

$$\|\Delta(\varepsilon_1, s_1)\| < \frac{k_1 - \beta_1(x_1, t)}{k_1}.$$

Entonces, usando (3.4.25) resulta en

$$\|\text{sign}(s_1) - \text{sigm}(\varepsilon_1, s_1)\| < \frac{k_1 - \beta_1(x_1, t)}{k_1}$$

$$1 - \text{sigm}(\varepsilon_1, s_1) < \frac{k_1 - \beta_1(x_1, t)}{k_1}.$$

Dado que estamos utilizando  $\text{sigm}(\varepsilon_1 s_1) = \tanh(\varepsilon_1 s_1)$ , y usando la forma exponencial de la función tangente hiperbólica, podemos definir

$$1 - \tanh(\varepsilon_1 s_1) < \frac{k_1 - \beta_1(x_1, t)}{k_1}$$

$$1 - \frac{e^{2\varepsilon_1 s_1} - 1}{e^{2\varepsilon_1 s_1} + 1} < \frac{k_1 - \beta_1(x_1, t)}{k_1}$$

$$\frac{2}{e^{2\varepsilon_1 s_1} + 1} < \frac{k_1 - \beta_1(x_1, t)}{k_1}$$

y de (3.4.24) sabemos que  $k_1 > \beta_1(x_1, t)$ , entonces

$$\frac{2k_1}{k_1 - \beta_1(x_1, t)} < e^{2\varepsilon_1 s_1} + 1,$$

y

$$\frac{k_1 + \beta_1(x_1, t)}{k_1 - \beta_1(x_1, t)} < e^{2\varepsilon_1 s_1}$$

$$\frac{\ln\left(\frac{k_1 + \beta_1(x_1, t)}{k_1 - \beta_1(x_1, t)}\right)}{2\varepsilon_1} < s_1,$$

por lo tanto  $s_1$  converge a una vecindad  $\|s_1\| \leq \Omega$  dada por

$$\Omega = \frac{\ln\left(\frac{k_1 + \beta_1(x_1, t)}{k_1 - \beta_1(x_1, t)}\right)}{2\varepsilon_1}$$

Ahora, definiendo  $\varphi = \dot{s}_1$ , la función de Lyapunov  $V_\varphi = \frac{1}{2}\varphi^T\varphi$  y usando (3.4.22), los cálculos nos dan

$$\begin{aligned}\dot{V}_\varphi &= \varphi^T \dot{\varphi} \\ &= \varphi^T \left( -k_1 \varepsilon P \varphi + \dot{\lambda}_u \right)\end{aligned}\tag{3.4.33}$$

Bajo la suposición  $S2$ , ó  $\|\dot{\lambda}_u\| \leq \beta_u \|\varphi\|$ , se cumple que  $\dot{V}_\varphi < 0$  y por lo tanto  $\varphi$  converge a asintóticamente a cero.

Entonces, debido a que  $\dot{s}_1 \rightarrow 0$  y  $\dot{s}_2 \rightarrow 0$  obtenemos lo siguiente

$$\begin{aligned}x_{2,1} &= -\lambda_u \\ b(y_1)u_1 &= -\lambda_m\end{aligned}$$

y el sistema (3.4.23) puede ser reformulado como

$$\begin{aligned}\dot{e}_1 &= -c_1 e_1 + e_2 \\ \dot{e}_2 &= -c_2 e_2\end{aligned}\tag{3.4.34}$$

Si  $c_1 > 0$  y  $c_2 > 0$ , entonces la solución del sistema (3.4.34) tiende a cero, resultando que

$$\lim_{t \rightarrow \infty} e_1(t) = 0$$

y el objetivo de control es alcanzado.

### 3.6. Control Super-Twisting Anidado (STA)

Una desventaja del MDIA, es que sigue utilizando un control discontinuo (al menos en el último bloque). Lo que puede ocasionar controles con componentes de alta frecuencia, e incluso *chattering*. Para evitar esto se propone incluir en la estrategia de control el principio del algoritmo Super-Twisting (ST) [49; 50], el cual provee de controles totalmente continuos. La combinación del algoritmo ST con MDA nos permitirá obtener un controlador con un alto grado de suavidad, libre de

*chattering*, y robusto a perturbaciones *matched* y *no-matched*. El procedimiento para diseñar este algoritmo se explicará a continuación.

### 3.6.1. Algoritmo STA

Utilizando, al igual que en el control por MDIA, el modelo siguiente

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= f_2(x_1, x_2) + b_2(x_1)u + \lambda(x_1, x_2, t)\end{aligned}$$

donde  $f_2(x_1, x_2) = -H(x_1)^{-1}(C(x_1, x_2)x_2 + g(x_1))$ ,  $b_2(x_1) = H(x_1)^{-1}$  y  $\lambda(x_1, x_2, t)$  es el término generado por la perturbaciones externas, variaciones paramétricas y demás. La salida de este sistema es  $y = x_1$ .

Cabe recordar que durante el diseño de este controlador también son tomadas como hipótesis las suposiciones de la sección 3.3.

Sea  $x_{1ref}(t)$  una función diferenciable, pero con derivadas desconocidas, que representa la referencia para el vector de posiciones angulares del manipulador. Dicha función es obtenida de la etapa de localización del objetivo. Entonces definimos la variable de error como  $e_1 = x_1 - x_{1ref}(t)$  y su derivada resulta en

$$\dot{e}_1 = x_2 + \lambda_u(x_1, t) \quad (3.5.1)$$

donde  $\lambda_u(x_1, t)$  es el término de perturbaciones *no-matched* definido por

$$\lambda_u(x_1, t) = -\dot{x}_{1ref}(t). \quad (3.5.2)$$

Considerando a  $x_2$  como el control virtual en (3.5.1), y definiendo la variable de error para el segundo bloque como

$$e_2 = x_2 - x_{2ref} \quad (3.5.3)$$

de donde obtenemos  $x_2 = e_2 + x_{2ref}$ , y si sustituimos en (3.5.1) nos da

$$\dot{e}_1 = e_2 + x_{2ref} + \lambda_u(x_1, t). \quad (3.5.4)$$

Para inducir Modos Deslizantes en la dinámica de error del primer bloque, diseñamos la referencia del segundo bloque basándonos en el algoritmo Super-Twisting como

$$\begin{aligned}x_{2ref} &= -\sigma_1 N_1 \text{sigm}(\varepsilon_1, e_1) + \mu_1 \\ \dot{\mu}_1 &= -k_1 \text{sign}(e_1)\end{aligned} \quad (3.5.5)$$

donde  $N_1 = \text{diag}\{|e_{11}|^{1/2}, \dots, |e_{1n}|^{1/2}\}$  con  $e_1 = [e_{11} \ \dots \ e_{1n}]^T$  y  $\sigma_1 > 0$  y  $k_1 > 0$  son parámetros constantes de diseño, los cuales serán definidos después. La variable  $\mu_1$

define el orden del algoritmo y proviene del algoritmo ST, el cual es un tipo de modos deslizantes de 2do orden. Cabe notar que se utiliza una función sigmoidea, en vez de la función signo utilizada en el algoritmo ST, esto debido a que en pasos posteriores del algoritmo se usará la diferenciación de  $x_{2ref}$ .

Procediendo con el segundo bloque, obtenemos la dinámica del error derivando (3.5.3) lo que resulta en

$$\dot{e}_2 = f_2(x_1, x_2) + b_2(x_1)u - \dot{x}_{2ref} + \lambda(x_1, x_2, t). \quad (3.5.6)$$

El término  $\dot{x}_{2ref}$  se obtiene por diferenciación directa de  $x_{2ref}$ , y está compuesto por una parte conocida  $\dot{x}_{2,k}$  y una parte desconocida  $\dot{x}_{2,u}$  de acuerdo a

$$\dot{x}_{2ref} = \dot{x}_{2,k} + \dot{x}_{2,u}, \quad (3.5.7)$$

las cuales están definidas como

$$\dot{x}_{2,k} = -\sigma_1 P x_2 + \dot{\mu}_1, \quad \dot{x}_{2,u} = -\sigma_1 P \lambda_u \quad (3.5.8)$$

donde

$$P = \text{diag}\{\chi(e_{11}), \dots, \chi(e_{1n})\} \quad (3.5.9)$$

$$\chi(t) = \varepsilon_1 |t|^{1/2} [1 - \tanh^2(\varepsilon_1 t)] + \frac{1}{2} |t|^{-1/2} \tanh(\varepsilon_1 t) \text{sign}(t).$$

Ahora, se define la ley de control  $u$  como

$$u = u_a + u_b \quad (3.5.10)$$

donde  $u_a$  será la parte de la ley de control para eliminar la dinámica conocida del sistema, y  $u_b$  será utilizada para eliminar las perturbaciones y estabilizar el sistema resultante.

Entonces, usando (3.5.10) podemos re-escribir (3.5.6) como

$$\dot{e}_2 = f_2(x_1, x_2) + b_2(x_1)u_a + b_2(x_1)u_b - \dot{x}_{2ref} + \lambda(x_1, x_2, t) \quad (3.5.11)$$

donde  $u_a$  se diseña para eliminar los términos conocidos de (3.5.11) con la forma

$$u_a = -b_2(x_1)^{-1} [f_2(x_1, x_2) - \dot{x}_{2,k}] \quad (3.5.12)$$

y la parte  $u_b$  será usada para rechazar las perturbaciones *matched*. Usando (3.5.12) y (3.5.7) la ecuación (3.5.11) se convierte en

$$\dot{e}_2 = b_2(x_1)u_b + \lambda_m(x_1, x_2, t) \quad (3.5.13)$$

donde el vector de perturbaciones *matched* equivale a

$$\lambda_m(x_1, x_2, t) = -\dot{x}_{2,u} + \lambda(x_1, x_2, t). \quad (3.5.14)$$

El siguiente paso es diseñar  $u_b$  usando el algoritmo ST de la forma

$$\begin{aligned} \mathbf{u}_b &= -\mathbf{b}_2(\mathbf{x}_1)^{-1}[\sigma_2 \mathbf{N}_2 \text{sign}(\mathbf{e}_2) + \mu_2] \\ \dot{\mu}_2 &= -k_2 \text{sign}(\mathbf{e}_2) \end{aligned} \quad (3.5.15)$$

donde  $\mathbf{N}_2 = \text{diag}\{|e_{21}|^{1/2}, \dots, |e_{2n}|^{1/2}\}$  con  $\mathbf{e}_2 = [e_{21} \ \dots \ e_{2n}]^T$  y  $\sigma_2 > 0$ ,  $k_2 > 0$  siendo parámetros de control que serán definidos más adelante.

Usando (3.5.4), (3.5.5), (3.5.13) y (3.5.15), el sistema de error en lazo cerrado se obtiene como

$$\begin{aligned} \dot{\mathbf{e}}_1 &= -\sigma_1 \mathbf{N}_1 \text{sign}(\varepsilon_1, \mathbf{e}_1) + \mu_1 + \mathbf{e}_2 + \lambda_u \\ \dot{\mu}_1 &= -k_1 \text{sign}(\mathbf{e}_1) \\ \dot{\mathbf{e}}_2 &= -\sigma_2 \mathbf{N}_2 \text{sign}(\mathbf{e}_2) + \mu_2 + \lambda_m \\ \dot{\mu}_2 &= -k_2 \text{sign}(\mathbf{e}_2). \end{aligned} \quad (3.5.16)$$

Definimos las cotas descritas en la suposición S2 de la siguiente manera

$$\beta_1(\mathbf{x}_1, t) = \alpha_1 |e_1|^{1/2} \quad \beta_2(\mathbf{x}_1, \mathbf{x}_2, t) < \alpha_2 |\mathbf{e}_2|^{1/2} \quad (3.5.17)$$

con  $\alpha_1 \geq 0$ ,  $\alpha_2 \geq 0$ , y también una cota de la diferencia entre la función signo y sigmoidea,  $\Delta(\varepsilon_1, s_1) = \text{sign}(s_1) - \text{sign}(\varepsilon_1, s_1)$ , utilizada en el primer bloque como

$$\|\Delta(\varepsilon_1, s_1)\| < \delta_1. \quad (3.5.18)$$

Finalmente, se establecen las desigualdades que definen los parámetros de control con la forma

$$\sigma_1 > \frac{2\alpha_1}{1-2\delta_1}, \quad k_1 > \sigma_1 \frac{5\alpha_1\sigma_1 + 4\alpha_1^2}{2(\sigma_1 - 2\alpha_1)}, \quad \alpha = \alpha_1 + \sigma_1\delta_1. \quad (3.5.19)$$

$$\sigma_2 > 2\alpha_2, \quad k_2 > \sigma_2 \frac{5\alpha_2\sigma_2 + 4\alpha_2^2}{2(\sigma_2 - 2\alpha_2)}, \quad (3.5.20)$$

para definir por completo el algoritmo de control propuesto. Estos valores se definieron para asegurar la estabilidad del sistema en lazo cerrado; su obtención se explicará en detalle a continuación.

## Análisis de Estabilidad del control STA

Primero, definimos una función candidata de Lyapunov, basados en el desarrollo propuesto en [51], para la dinámica de  $\mathbf{e}_2$  de la forma

$$V_2(\mathbf{e}_2) = \Psi_2^T \mathbf{P}_2 \Psi_2 \quad (3.5.21)$$

donde  $\Psi_2^T = [|\mathbf{e}_2|^{1/2} \text{sign}(\mathbf{e}_2) \quad \mathbf{e}_2]$  y  $\mathbf{P}_2 = \frac{1}{2} \begin{bmatrix} 4k_2 + \sigma_2^2 & -\sigma_2 \\ -\sigma_2 & 2 \end{bmatrix}$ .

La diferenciación de (3.5.21) está dada por

$$\dot{V}_2(\mathbf{e}_2) = -\frac{1}{|\mathbf{e}_2|^{1/2}} \Psi_2^T Q_2 \Psi_2 + \frac{\lambda_u}{|\mathbf{e}_2|^{1/2}} \mathbf{q}_2^T \Psi_2 \quad (3.5.22)$$

donde

$$Q_2 = \frac{\sigma_2}{2} \begin{bmatrix} 2k_2 + \sigma_2^2 & -\sigma_2 \\ -\sigma_2 & 1 \end{bmatrix}, \quad \mathbf{q}_2^T = \begin{bmatrix} \left(2k_2 + \frac{\sigma_2^2}{2}\right) & \frac{-\sigma_2}{2} \end{bmatrix},$$

Entonces, las condiciones necesarias para que  $\dot{V}_2(\mathbf{e}_2)$  sea semi-definida negativa son

$$\sigma_2 > 2\alpha_2, \quad k_2 > \sigma_2 \frac{5\alpha_2\sigma_2 + 4\alpha_2^2}{2(\sigma_2 - 2\alpha_2)}.$$

Estas son las condiciones descritas en (3.5.20), por lo cual, podemos concluir que  $\mathbf{e}_2 \rightarrow 0$  en tiempo finito.

Procediendo con el primer bloque, de (3.5.18) tenemos que

$$\begin{aligned} \Delta_1(\varepsilon_1, \mathbf{e}_1) &= \text{sign}(\mathbf{e}_1) - \text{sign}(\varepsilon_1 \mathbf{e}_1) \\ \|\Delta_1(\varepsilon_1, \mathbf{e}_1)\| &< \delta_1 \end{aligned} \quad (3.5.23)$$

y usando (3.5.23) la dinámica del primer bloque se transforma en

$$\begin{aligned} \dot{\mathbf{e}}_1 &= -\sigma_1 N_1 \text{sign}(\mathbf{e}_1) + \mu_1 + \mathbf{e}_2 + \lambda_0 \\ \dot{\mu}_1 &= -k_1 \text{sign}(\mathbf{e}_1) \end{aligned} \quad (3.5.24)$$

donde  $\lambda_0 = \lambda_u + \sigma_1 N_1 \Delta(\varepsilon_1, \mathbf{e}_1)$ , además sabemos por (3.5.17) y (3.5.23) que  $\lambda_0$  está acotado por

$$\|\lambda_0\| < \alpha |\mathbf{e}_1|^{1/2} \quad \alpha = \alpha_1 + \sigma_1 \delta_1. \quad (3.5.25)$$

Similarmente al segundo bloque, usamos una función candidata de Lyapunov  $V_1(\mathbf{e}_1) = \Psi_1^T P_1 \Psi_1$  para el sistema (3.5.24) donde

$$\Psi_1^T = \begin{bmatrix} |\mathbf{e}_1|^{1/2} \text{sign}(\mathbf{e}_1) & \mathbf{e}_1 \end{bmatrix}, \quad P_1 = \frac{1}{2} \begin{bmatrix} 4k_1 + \sigma_1^2 & -\sigma_1 \\ -\sigma_1 & 2 \end{bmatrix},$$

y cuya derivada está dada por

$$\dot{V}_1(\mathbf{e}_1) = -\frac{1}{|\mathbf{e}_1|^{1/2}} \Psi_1^T Q_1 \Psi_1 + \frac{\lambda_0}{|\mathbf{e}_1|^{1/2}} \mathbf{q}_1^T \Psi_1 \quad (3.5.26)$$

$$\text{con } Q_1 = \frac{\sigma_1}{2} \begin{bmatrix} 2k_1 + \sigma_1^2 & -\sigma_1 \\ -\sigma_1 & 1 \end{bmatrix}, \quad \mathbf{q}_1^T = \begin{bmatrix} \left(2k_1 + \frac{\sigma_1^2}{2}\right) & \frac{-\sigma_1}{2} \end{bmatrix}.$$

Las condiciones necesarias para que  $\dot{V}_1(\mathbf{e}_1)$  sea semi-definida negativa son

$$\sigma_1 > \frac{2\alpha_1}{1-2\delta_1}, \quad k_1 > \sigma_1 \frac{5\alpha\sigma_1 + 4\alpha^2}{2(\sigma_1 - 2\alpha)}, \quad \alpha = \alpha_1 + \sigma_1\delta_1,$$

las cuales están descritas en (3.5.19).

Por lo tanto, se puede concluir que el origen del sistema de error es un punto de equilibrio, y que es *globalmente asintóticamente estable*, por lo que el objetivo de control se satisface.

### 3.7. Control Super-Twisting Integral Anidado (STIA)

La inclusión del concepto de Modo Deslizante Integral al diseño del STA, nos permite proponer el algoritmo STIA que ofrece las ventajas del algoritmo ST y los controladores descritos en la sección anterior.

#### 3.7.1. Algoritmo STIA

Se definirá el procedimiento para diseñar un controlador basado en la combinación del MDIA y el algoritmo ST, cuyo objetivo es el seguimiento de trayectorias.

Utilizando, de la misma manera que en los MDIA, el modelo siguiente

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f_2(x_1, x_2) + b_2(x_1)u + \lambda(x_1, x_2, t) \end{aligned}$$

donde  $f_2(x_1, x_2) = -H(x_1)^{-1}(C(x_1, x_2)x_2 + g(x_1))$ ,  $b_2(x_1) = H(x_1)^{-1}$  y  $\lambda(x_1, x_2, t)$  es el término generado por las perturbaciones externas, variaciones paramétricas y demás. La salida de este sistema es  $y = x_1$ . Durante el diseño del controlador, serán válidas también las suposiciones de la sección 3.3.

Sea  $x_{1ref}(t)$  una función diferenciable, pero con derivadas desconocidas, que representa la referencia para el vector de posiciones angulares del manipulador. Entonces, definimos la variable de error como  $e_1 = x_1 - x_{1ref}(t)$  y su derivada

$$\dot{e}_1 = x_2 + \lambda_u(x_1, t) \tag{3.6.1}$$

donde  $\lambda_u(x_1, t)$  es la perturbación *no-matched* definida por la igualdad

$$\lambda_u(x_1, t) = -\dot{x}_{1ref}.$$

Entonces, se define la función pseudo-deslizante  $s_1$  para el primer bloque como sigue

$$s_1 = e_1 + z_1 \quad (3.6.2)$$

con  $z_1(0) = -e_1(0)$  y donde  $z_1$  es la variable integral que será definida después. La dinámica de  $s_1$  puede ser obtenida como

$$\dot{s}_1 = x_2 + \dot{z}_1 + \lambda_u(x_1, t). \quad (3.6.3)$$

Ahora, considerando a  $x_2$  como el control virtual para este bloque, y utilizando el principio de los MDI, proponemos

$$x_{2ref} = x_{2,0} + x_{2,1} \quad (3.6.4)$$

donde  $x_{2,0}$  es la parte nominal del pseudo-control y  $x_{2,1}$  será diseñado para rechazar la perturbación. Definimos la superficie para el segundo bloque como

$$s_2 = e_2 + z_2, \quad e_2 = x_2 - x_{2ref} \quad (3.6.5)$$

con  $z_2$  como la variable integral. Entonces, podemos obtener  $x_2 = s_2 + x_{2ref} - z_2$  y al sustituirlo en (3.6.3) resulta en

$$\dot{s}_1 = s_2 - z_2 + x_{2,0} + x_{2,1} + \dot{z}_1 + \lambda_u. \quad (3.6.6)$$

A continuación, definimos la dinámica de  $z_1$  como

$$\dot{z}_1 = -(s_2 - z_2 + x_{2,0}) \quad (3.6.7)$$

con las condiciones iniciales  $z_1(0) = -e_1(0)$ , con el objetivo de asegurar la existencia de modo deslizante  $s_1(0)$  desde el instante inicial, y definimos  $x_{2,0}$  como sigue

$$x_{2,0} = -c_1 e_1 \quad (3.6.8)$$

donde  $c_1 > 0$ , para asegurar la estabilidad del sistema nominal, es decir sin perturbaciones. Entonces, la dinámica de  $s_1$  queda

$$\dot{s}_1 = x_{2,1} + \lambda_u. \quad (3.6.9)$$

La segunda parte del pseudo-control  $x_{2,1}$  se elige como

$$x_{2,1} = -k_1 \text{sigm}(\varepsilon_1 s_1) \quad (3.6.10)$$

donde  $k_1 > 0$ , para rechazar las perturbaciones del bloque. El desarrollo del controlador para el primer bloque es igual que en el control por MDIA.

Ahora podemos definir el control para el segundo bloque, donde utilizaremos el algoritmo ST. Derivando (3.6.5) sobre las trayectorias del sistema obtenemos

$$\dot{s}_2 = f_2(x_1, x_2) + b_2(x_1)u - \dot{x}_{2ref} + \dot{z}_2 + \lambda(x_1, x_2, t) \quad (3.6.11)$$

donde  $x_{2ref}$  está compuesto por una parte conocida  $\dot{x}_{2,c}$  y una desconocida  $\dot{x}_{2,c}$  de la siguiente forma

$$\dot{x}_{2ref} = \dot{x}_{2,c} + \dot{x}_{2,d},$$



con

$$\dot{x}_{2,c} = -k_1 \epsilon_1 P x_{2,1} - c_1 x_2 \quad \dot{x}_{2,d} = -(c_1 + k_1 \epsilon P) \lambda_u$$

y

$$P = \begin{pmatrix} 1 - \tanh^2(\epsilon_1 s_1(1)) & 0 \\ 0 & 1 - \tanh^2(\epsilon_1 s_1(2)) \end{pmatrix}.$$

Ahora, si definimos

$$\lambda_m = \dot{x}_{2,d} + \lambda(x_1, x_2, t). \quad (3.6.12)$$

como el término de perturbaciones *matched* del sistema, podemos re-escribir (3.6.11) de la forma

$$\dot{s}_2 = f_2(x_1, x_2) + b_2(x_1)u - \dot{x}_{2,c} + \dot{z}_2 + \lambda_m \quad (3.6.13)$$

Utilizando el espíritu del MDI, diseñamos el término de control como  $u = u_0 + u_1$  y la dinámica de  $s_2$  resulta en

$$\dot{s}_2 = f_2(x_1, x_2) + b_2(x_1)u_0 + b_2(x_1)u_1 - \dot{x}_{2,c} + \dot{z}_2 + \lambda_m, \quad (3.6.14)$$

y eligiendo  $\dot{z}_2$  como

$$\dot{z}_2 = -f_2(x_1, x_2) - b_2(x_1)u_0 + \dot{x}_{2,c} \quad (3.6.15)$$

con  $z_2(0) = -e_2(0)$  (para asegurar la ocurrencia de MD desde el instante inicial) la ecuación (3.6.14) se convierte en

$$\dot{s}_2 = b_2(x_1)u_1 + \lambda_m. \quad (3.6.16)$$

Diseñamos  $u_0$  como

$$u_0 = b_2(x_1)^{-1}(-f_2(x_1, x_2) + \dot{x}_{2,ref} - c_2 e_2) \quad (3.6.17)$$

donde  $c_2 > 0$ , y usamos el algoritmo ST para diseñar  $u_1$  como

$$u_1 = b_2(x_1)^{-1} \left[ -W_1 \|s_2\|^\rho \text{sign}(s_2) + \mu \right] \quad (3.6.18)$$

$$\dot{\mu} = -W_2 \text{sign}(s_2)$$

donde  $W_1, W_2, \rho$  son constantes positivas de diseño, de forma que logremos inducir modos deslizantes en (3.6.16).

Usando los términos de control ( $u_1$  y  $u_0$ ) y pseudo-control ( $x_{2,1}$  y  $x_{2,0}$ ) obtenemos la dinámica de las superficies deslizantes,  $s_1$  y  $s_2$ , en lazo cerrado como

$$\begin{aligned} \dot{s}_1 &= -k_1 \text{sign}(\epsilon_1 s_1) + \lambda_u, \\ \dot{s}_2 &= -W_1 \|s_2\|^\rho \text{sign}(s_2) + \mu + \lambda_m, \\ \dot{\mu} &= -W_2 \text{sign}(s_2), \end{aligned} \quad (3.6.19)$$

y la dinámica de las variables de error en lazo cerrado resultan en

$$\begin{aligned}
\dot{e}_1 &= -c_1 e_1 + e_2 - k_1 \text{sigm}(\varepsilon_1 s_1) + \lambda_u, \\
\dot{e}_2 &= -c_2 e_2 - W_1 \|s_2\|^\rho \text{sign}(s_2) + \mu + \lambda_m, \\
\dot{\mu} &= -W_2 \text{sign}(s_2).
\end{aligned} \tag{3.6.20}$$

Después de eso, y de la misma forma que en el algoritmo STA, se define la cota de la perturbación del segundo bloque como

$$\beta_2(x_1, x_2, t) = \alpha |s_2|^\rho \quad \rho = 1/2 \tag{3.6.21}$$

Finalmente, se definen las ganancias de control de la siguiente manera [51]

$$W_1 > 2\alpha \quad W_2 > W_1 \frac{5\alpha W_1 + 4\alpha^2}{2(W_1 - 2\alpha)} \tag{3.6.22}$$

para finalizar así con el diseño del controlador. Los cotas de  $W_1$  y  $W_2$  provienen del análisis de estabilidad de (3.6.20) y (3.6.19), el cual será desarrollado a continuación.

## Análisis de Estabilidad del controlador por STIA

Definimos una función de Lyapunov para la dinámica de  $s_2$  como sigue

$$V_2(s_2) = \Psi^T P \Psi \tag{3.6.23}$$

donde

$$\Psi^T = \begin{bmatrix} |s_2|^{1/2} \text{sign}(s_2) & s_2 \end{bmatrix} \quad \text{y} \quad P = \frac{1}{2} \begin{bmatrix} 4W_2 + W_1^2 & -W_1 \\ -W_1 & 2 \end{bmatrix}.$$

La derivada de (3.6.23) nos queda [51]

$$\dot{V}_2 = -\frac{1}{|s_2|^{1/2}} \Psi^T Q \Psi + \frac{\lambda_m}{|s_2|^{1/2}} q_1^T \Psi \tag{3.6.24}$$

donde

$$\begin{aligned}
Q &= \frac{W_1}{2} \begin{bmatrix} 2W_2 + W_1^2 & -W_1 \\ -W_1 & 1 \end{bmatrix}, \\
q_1^T &= \begin{bmatrix} \left( 2W_2 + \frac{W_1^2}{2} \right) & \frac{-W_1}{2} \end{bmatrix},
\end{aligned}$$

y

$$q_2^T = [-W_1 \quad 2].$$

la cual bajo las condiciones (3.6.22) es definida negativa, esto es  $\dot{V}_2 < 0$ . Por lo tanto,  $s_2$  y  $\dot{s}_2$  convergen a cero en tiempo finito.

El primer bloque es idéntico al de CMDIA, y en la sección 3.5.1 demostramos como es estable, y que  $s_1$  converge a una vecindad de cero dada por

$$\Omega = \frac{\ln\left(\frac{[k_1 + \beta_1(x_1, t)]}{[k_1 - \beta_1(x_1, t)]}\right)}{2\varepsilon_1},$$

y que  $\dot{s}_1$  converge asintóticamente a cero.

Entonces, debido a que  $\dot{s}_1 \rightarrow 0$  y  $\dot{s}_2 \rightarrow 0$  obtenemos lo siguiente

$$x_{2,1} = -\lambda_u, \quad b(y_1)u_1 = -\lambda_m$$

y el sistema de error en lazo cerrado puede ser reformulado como

$$\begin{aligned} \dot{e}_1 &= -c_1 e_1 + e_2 \\ \dot{e}_2 &= -c_2 e_2 \end{aligned} \tag{3.6.25}$$

Si  $c_1 > 0$  y  $c_2 > 0$ , entonces una solución del sistema (3.6.25) tiende a cero, resultando que

$$\lim_{t \rightarrow \infty} e_1(t) = 0$$

y el objetivo de control es alcanzado.

### 3.8. Control por Modos Deslizantes Discretos (MDD)

El control por modos deslizantes ha sido ampliamente estudiado y bien establecido para sistemas en tiempo continuo, pero la mayoría de las estrategias de control en tiempo real son implementadas de forma discreta, casi siempre, por medio de un dispositivo digital programable. Esto llevó a muchos investigadores a prestar más atención al diseño de controladores de modos deslizantes discretos.

Debido a la velocidad finita de muestreo, algunas propiedades válidas para sistemas continuos no son apropiadas para sistemas discretos. Aún más importante, un controlador desarrollado para sistemas continuos podría volverse inestable si se implementa de manera directa en un sistema digital. Por lo tanto, un rediseño de la estrategia de control por modos deslizantes para sistemas muestreados o discretos es necesario.

Existen dos enfoques para diseñar modos deslizantes discretos. Uno se enfoca en conservar el término discontinuo en la ley de control por MDD [52; 53]. El otro utiliza el principio del control equivalente, junto con un observador de perturbaciones, para obtener una señal continua de control [54; 55].

## 3.9. Control por Modos Deslizantes Integrales Discretos (MDID)

El uso de una señal discontinua de control en sistemas discretos genera un efecto indeseable, en la respuesta en lazo cerrado, conocido como *chattering discreto*. Por esta razón, el control equivalente es usado en este trabajo para diseñar un controlador por Modos Deslizantes Integrales Discreto (MDID) libre de *chattering*. Además de evitar el efecto ya mencionado, se obtiene una ley de control robusta a perturbaciones desde el instante inicial y con ganancia de control reducida. Esto se verá en detalle a continuación.

### 3.9.1. Algoritmo de MDID

El objetivo de este algoritmo de control es lograr el seguimiento de referencia de la pose del efector final del robot. Antes de iniciar con el diseño del controlador, recordemos el modelo discreto de cinemática diferencial obtenido en la sección 3.2.2, el cual quedó definido como

$$\begin{aligned} \mathbf{x}_{1,k+1} &= \mathbf{x}_{1,k} + T\mathbf{u}_k \\ \mathbf{x}_{2,k+1} &= \mathbf{x}_{2,k} + TJ(\mathbf{x}_{1,k})\mathbf{u}_k + \mathbf{d}_k(\mathbf{x}_{1,k},k) \\ \mathbf{y}_k &= \mathbf{x}_{2,k}. \end{aligned} \quad (3.8.1)$$

donde  $\mathbf{x}_{1,k} \in \mathbb{R}^n$  y  $\mathbf{x}_{2,k} \in \mathbb{R}^6$  son las variables de estado, que representan a la *velocidad angular* de las uniones y a la pose del efector final, respectivamente,  $\mathbf{y}_k$  es la salida del sistema,  $J(\mathbf{x}_{1,k}) \in \mathbb{R}^{6 \times n}$  es el jacobiano del robot,  $\mathbf{u}_k$  es el término de control,  $\mathbf{d}_k(\mathbf{x}_{1,k},k)$  son las perturbaciones del sistema, y  $T$  es el período de muestreo.

Ahora, el primer paso es definir la variable de error  $\mathbf{e}_k \in \mathbb{R}^6$  como

$$\mathbf{e}_k = \mathbf{x}_{2,k} - \mathbf{y}_{k,\text{ref}} \quad (3.8.2)$$

con  $\mathbf{y}_{k,\text{ref}}$  como la referencia a seguir por la salida del sistema, es decir, la pose del efector final. Usando (3.8.2) y (3.8.1) podemos obtener la dinámica de la variable de error como

$$\mathbf{e}_{k+1} = \mathbf{x}_{2,k} + TJ(\mathbf{x}_{1,k})\mathbf{u}_k + \mathbf{d}_k(\mathbf{x}_{1,k},k) - \mathbf{y}_{k+1,\text{ref}}, \quad (3.8.3)$$

y suponemos que no conocemos  $y_{k+1,ref}$ , tal como sería en la mayoría de las aplicaciones reales. Entonces, combinamos los términos desconocidos en uno solo  $\lambda_k(x_{1,k}, k)$  para re-definir (3.8.3) como

$$e_{k+1} = x_{2,k} + TJ(x_{1,k})u_k + \lambda_k(x_{1,k}, k) \quad (3.8.4)$$

donde  $\lambda_k(x_{1,k}, k) = d_k(x_{1,k}, k) - y_{k+1,ref}$ .

El siguiente paso es diseñar la superficie deslizante  $s_k \in \mathbb{R}^n$  como

$$s_k = Pe_k + z_k \quad (3.8.5)$$

donde  $P \in \mathbb{R}^{n \times 6}$  es una matriz de diseño y  $z_k \in \mathbb{R}^n$  es la variable integral que será definida después.

Bajo la filosofía de los Modos Deslizantes Integrales, definimos el control en dos términos de la forma

$$u_k = u_{k0} + u_{k1} \quad (3.8.6)$$

donde  $u_{k0}$  es la parte nominal del control y  $u_{k1}$  será diseñado para rechazar el término de perturbación  $\lambda_k(x_{1,k}, k)$ .

Entonces, usando (3.8.6), la dinámica del error se convierte en

$$e_{k+1} = x_{2,k} + TJ(x_{1,k})u_{k0} + TJ(x_{1,k})u_{k1} + \lambda_k(x_{1,k}, k) \quad (3.8.7)$$

y usando (3.8.5) y (3.8.7) la proyección del sistema en el sub-espacio  $s_k$  puede ser obtenida como

$$s_{k+1} = P(x_{2,k} + TJ(x_{1,k})u_{k0} + TJ(x_{1,k})u_{k1} + \lambda_k(x_{1,k}, k)) + z_{k+1}. \quad (3.8.8)$$

Ahora, la dinámica para la variable integral  $z_k$  se elige como

$$z_{k+1} = -P[x_{2,k} + TJ(x_{1,k})u_{k0}], \quad (3.8.9)$$

lo que simplifica (3.8.8) en

$$s_{k+1} = P(\lambda_k(x_{1,k}, k) + TJ(x_{1,k})u_{k1}). \quad (3.8.10)$$

Para generar un modo deslizante sobre  $s_k = 0$  y rechazar la perturbación, la segunda parte del control  $u_{k1}$  se selecciona con la forma

$$u_{k1} = u_{k1eq}, \quad u_{k1eq} = -J^+(x_{1,k})[\lambda_k(x_{1,k}, k)/T] \quad (3.8.11)$$

donde el control equivalente  $u_{k1eq}$  es obtenido de  $s_{k+1} = 0$  en (3.8.10) y  $J^+(x_{1,k})$  es la pseudo-inversa de  $J(x_{1,k})$ . Este control conduce la trayectoria del sistema sobre la superficie deslizante  $s_k = 0$  en un periodo de muestreo; y para asegurar la

ocurrencia de esto desde  $k=0$ , la condición inicial para (3.8.9) se elige como  $z_0 = -P\mathbf{e}_0$ .

Sin embargo, el control (3.8.11) no puede ser implementado debido a que  $\lambda_k(x_{1,k}, k)$  es desconocido. Por otro lado, este término puede ser estimado, con algunas suposiciones de continuidad, por su valor previo  $\lambda_{k-1}$  [54]. Considerando una muestra antes de la ecuación (3.8.10), podemos obtener la estimación  $\hat{\lambda}_k$  de la siguiente forma

$$\hat{\lambda}_k = \lambda_{k-1} = P^+ s_k - TJ(x_{1,k-1})u_{(k-1)}. \quad (3.8.12)$$

Sustituyendo  $\lambda_k$  por su estimación  $\hat{\lambda}_k$  en (3.8.11) resulta en

$$u_{k1} = -J^+(x_{1,k})[\lambda_{k-1}/T] = -J^+(x_{1,k})\left[\frac{1}{T}P^+ s_k - J(x_{1,k-1})u_{(k-1)}\right]. \quad (3.8.13)$$

Este término de control se encargará de eliminar la perturbación  $\lambda_k(x_{1,k}, k)$ .

Finalmente, para introducir una dinámica deseada en la variable de error definimos el control nominal como

$$u_{k0} = -\frac{J(x_{1,k})^+}{T}[x_{2,k} - K\mathbf{e}_k] \quad (3.8.14)$$

donde  $K \in \mathbb{R}^{6 \times 6}$  es una matriz Schur, con lo que terminamos el diseño del controlador.

## Análisis de Estabilidad del control por MDID

De las ecuaciones (3.8.10) y (3.8.13) se obtiene

$$\begin{aligned} s_{k+1} &= P(\lambda_k - \lambda_{k-1}) \\ &= P\chi_k \end{aligned} \quad (3.8.15)$$

donde  $\chi_k = \lambda_k - \lambda_{k-1}$ , por lo que se puede concluir que  $s_k$  converge a una vecindad de cero dada por  $\|P\chi_k\|$ .

Ahora, usando (3.8.13) y (3.8.7), podemos obtener la dinámica de error en modo deslizante como

$$\mathbf{e}_{k+1} = x_{2,k} + TJ(x_{1,k})u_{k0} + \chi_k. \quad (3.8.16)$$

Se puede observar que el orden del sistema de error original es el mismo que el definido en modo deslizante (3.8.16). Esta es una característica de los modos deslizantes integrales. Usando (3.8.14), se puede convertir a (3.8.16) en

$$e_{k+1} = Ke_k + \chi_k, \quad (3.8.17)$$

si  $K$  es una matriz Schur, entonces  $e_k$  converge a una vecindad de cero delimitada por  $\|\chi_k\|$  y el objetivo de control es alcanzado.

### 3.10. Control Super-Twisting Discreto (STD)

Como se mencionó en 3.5.1, el algoritmo Super-Twisting (ST) se usa ampliamente para sustituir controles discontinuos en plantas continuas, lo cual permite evitar el efecto *chattering*. En esta sección se propondrá el algoritmo ST para sustituir controles discontinuos en plantas discretas, y evitar otro indeseable fenómeno conocido como *chattering discreto*, sin tener que usar el *control equivalente* y, además, aprovechar las ventajas que el algoritmo ST ofrece. Dicho algoritmo de control se describirá a continuación.

#### 3.10.1. Algoritmo STD

Igual que en el control por MDID, considere el modelo discreto de cinemática diferencial del manipulador

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + Tu_k \\ x_{2,k+1} &= x_{2,k} + TJ(x_{1,k})u_k + d_k(x_{1,k}, k) \\ y_k &= x_{2,k}. \end{aligned} \quad (3.9.1)$$

donde  $T$  es el período de muestreo,  $x_{1,k} \in \mathbb{R}^n$  es el vector de velocidades angulares de las uniones del robot,  $x_{2,k} \in \mathbb{R}^6$  es la pose del efector final del robot,  $u_k$  es la señal de control,  $J_k(x_{1,k}) \in \mathbb{R}^{6 \times n}$  es el jacobiano del robot,  $y_k$  es la salida del sistema y  $d_k(\bullet)$  es un vector de perturbaciones.

Iniciamos con el diseño del algoritmo definiendo la superficie deslizante como el error de seguimiento

$$s_k = x_{2,k} - y_{k,ref} \quad (3.9.2)$$

con  $y_{k,ref}$  como la discretización de la referencia a seguir cuyas derivadas son desconocidas pero acotadas. Podemos obtener la dinámica de  $s_k$  usando (3.9.2) y (3.9.1) como

$$s_{k+1} = x_{2,k} + TJ(x_{1,k})u_k + d_k(x_{1,k}, k) - y_{k+1,ref} \quad (3.9.3)$$

y suponiendo  $y_{k+1,ref}$  es desconocida, podemos obtener una representación de como sigue

$$s_{k+1} = x_{2,k} + TJ(x_{1,k})u_k + \lambda(x_k, k) \quad (3.9.4)$$

que donde  $\lambda(x_k, k) = d_k(x_k, k) - y_{k+1,ref}$  es el nuevo término de perturbaciones.

Diseñando la señal de control  $u_k$  con los principios del algoritmo ST (ver sección 2.4.4) obtenemos

$$u_k = -\frac{J(x_{1,k})^+}{T} \left[ x_{2,k} + L \|s_k\|^{1/2} \text{sigm}(s_k, \varepsilon_1) - u_{2,k} - s_k \right] \quad (3.9.5)$$

$$u_{2,k+1} = -W \text{sigm}(s_k, \varepsilon_2) + u_{2,k}$$

donde  $\text{sigm}(\bullet)$  es una función sigmoidea cuya pendiente se define por  $\varepsilon_1$  y  $\varepsilon_2$ , respectivamente, y  $L > 0$  y  $W > 0$  son parámetros de diseño del controlador.

Usando (3.9.5) y (3.9.4) obtenemos el siguiente sistema en lazo cerrado

$$s_{k+1} = -L \|s_k\|^{1/2} \text{sigm}(s_k, \varepsilon_1) + u_{2,k} + s_k + \delta(x_k, T) \quad (3.9.6)$$

$$u_{2,k+1} = -WT \text{sigm}(s_k, \varepsilon_2) + u_{2,k}$$

donde  $L$  y  $W$  serán diseñados para asegurar la estabilidad de (3.9.6).





## 4. Resultados en Simulación

Los controladores diseñados en este trabajo se probaron en simulación, algunos usando el modelo dinámico obtenido y otros el modelo cinemático diferencial. Además los controladores obtenidos en tiempo discreto se implementaron también en tiempo real y se compararon con un controlador estándar PID discreto (PID) para evaluar su desempeño. Esto será tratado en esta sección y se mostrarán los resultados obtenidos por estas pruebas.

### 4.1. Parámetros del Modelo Dinámico.

Para simular el desempeño de los controladores dinámicos propuestos en este trabajo, se utilizó el modelo dinámico obtenido por *álgebra geométrica conformal*, el cual definimos en la sección 3.2.3 como

$$\delta m L_{cm} \ddot{\mathbf{q}} + \mathbf{V}^T \mathbf{m} (\mathbf{V} \ddot{\mathbf{q}} + \dot{\mathbf{V}} \dot{\mathbf{q}} + \mathbf{a}) = \boldsymbol{\tau} \quad (4.1.1)$$

El manipulador cuenta con 2 GDL y los parámetros que definen su modelo dinámico (4.1.1) son los siguientes:

$m_1 = 10$  kg.,  $m_2 = 1$  kg.,  $\ell_1 = 1$  m.,  $\ell_2 = 1$  m.,  $\ell_{c,1} = 0.5$  m.,  $\ell_{c,2} = 0.4$  m., donde  $\ell_i$ ,  $M_i$  y  $\ell_{c,i}$  son la longitud, masa y distancia entre el centro de masas y eje de rotación del  $i^{\text{ésimo}}$  eslabón, respectivamente.

Además, es necesario definir los ejes de actuación del robot. En este caso, todas las uniones son rotativas por lo que  $D_i = L_i$  para  $i = \{1, 2\}$  las cuales se definen como  $L_1 = e_{12}$  y  $L_2 = e_{12} + \ell_1 e_2 e_\infty$ . Estos parámetros pueden ser apreciados en la siguiente figura.

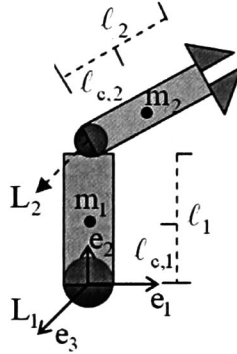


Figura 34. Manipulador de 2 grados de libertad y sus parámetros.

Finalmente, los términos de perturbación y de referencia, para las posiciones angulares, son

$$\lambda = \left[ 4, 2 \sin\left(\frac{t}{2}\right) \right]^T \quad x_{1ref} = \left[ 2 \sin(2t), 2 + 3 \cos(t) \right]^T \quad (4.1.2)$$

con lo que el modelo queda completamente definido. A continuación se presentarán los resultados en simulación de los controladores propuestos en lazo cerrado con el modelo dinámico definido por los parámetros de esta sección.

## 4.2. Control por MDIA

Los parámetros utilizados en simulación del controlador por MDIA, desarrollado en la sección 3.5 están definidos como

$$k_1 = 6 \quad c_1 = 3 \quad \varepsilon_1 = 10 \quad k_2 = 30 \quad c_2 = 3. \quad (4.2.1)$$

Los resultados de la simulación pueden ser apreciados en las siguientes figuras. La Figura 35 muestra la posición angular para ambas uniones del manipulador. Se puede apreciar que el desempeño del control por MDIA diseñado es satisfactorio, ya que los objetivos de control son alcanzados al lograr el seguimiento de trayectorias y rechazar las perturbaciones definidas en (4.1.2).

Los errores de seguimiento se pueden observar en la Figura 36, y se aprecia que dichas variables convergen a una vecindad de cero en tiempo finito. También se puede ver, en la parte derecha de la misma figura, el retrato de fase generado por las variables de error de ambas uniones. Se aprecia la convergencia asintótica al origen del retrato de fase y el comportamiento oscilatorio de las variables. En la Figura 37 se pueden observar las señales de control, esfuerzos de torsión, para cada una de las uniones. Se aprecia la componente de alta frecuencia en el transitorio de las variables de control, debido al uso de la función signo en el segundo bloque del sistema, y la desaparición de dicha componente en el estado estacionario.

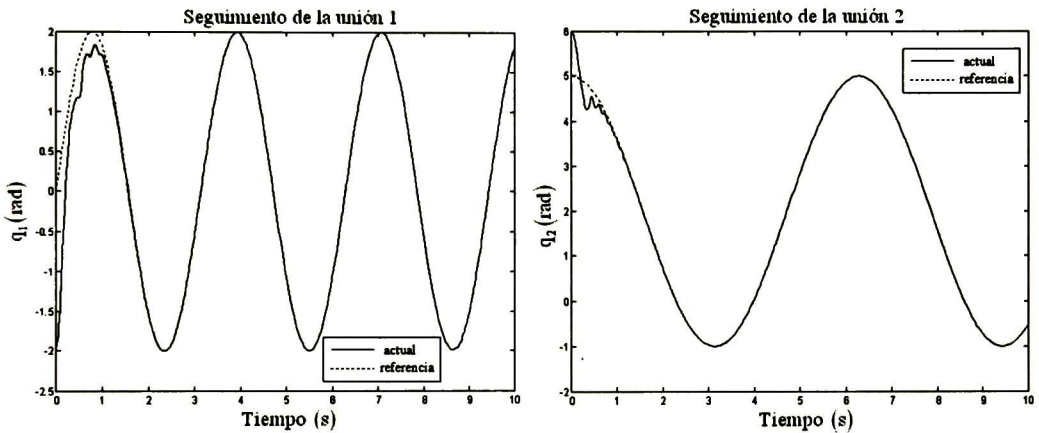


Figura 35. MDIA: Respuesta de seguimiento para la 1<sup>era</sup> unión (izq.) y la 2<sup>da</sup> (der.)

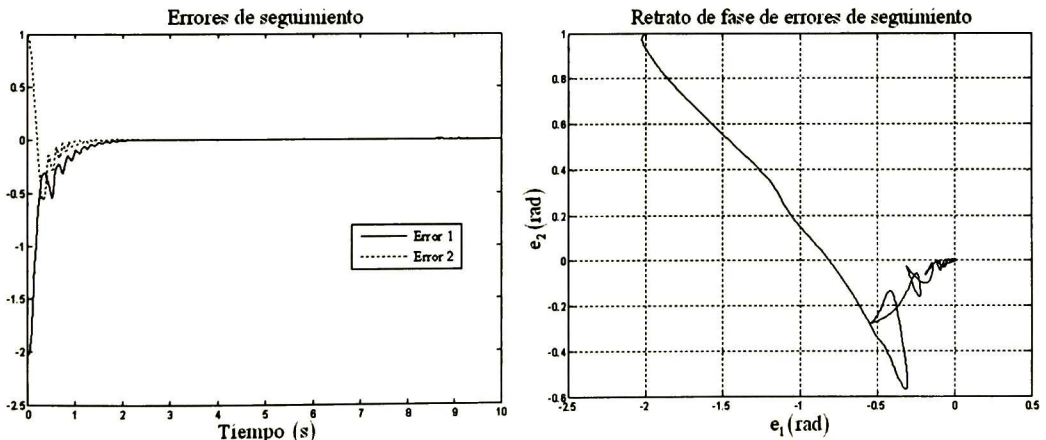


Figura 36. MDIA: Variables de error (izq.) y retrato de fase de errores (der.)

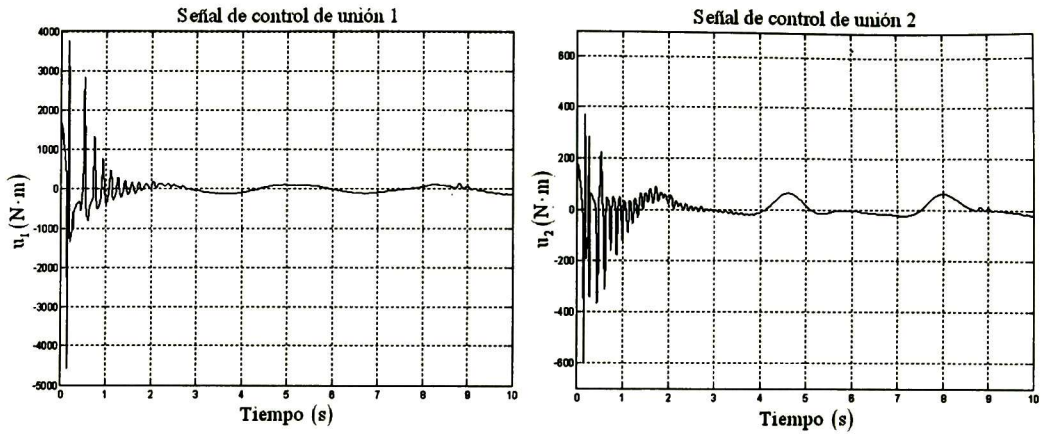


Figura 37. MDIA: Señales de control para la primera unión (izq.) y para segunda (der.)

### 4.3. Control por STA

Los parámetros utilizados en la simulación del controlador por STA, desarrollado en la sección 3.6, fueron diseñados como

$$k_1 = 6 \quad \sigma_1 = 12 \quad \varepsilon_1 = 6 \quad k_2 = 10 \quad \sigma_2 = 30. \quad (4.3.1)$$

Las siguientes figuras muestran los resultados de la simulación para este controlador. En la Figura 38 se muestra la posición angular para ambas uniones del manipulador. Además de rechazar las perturbaciones y lograr el objetivo de control, lo hace de una manera más suave que el control por MDIA, debido al uso del algoritmo *Super-Twisting* en su ley de control.

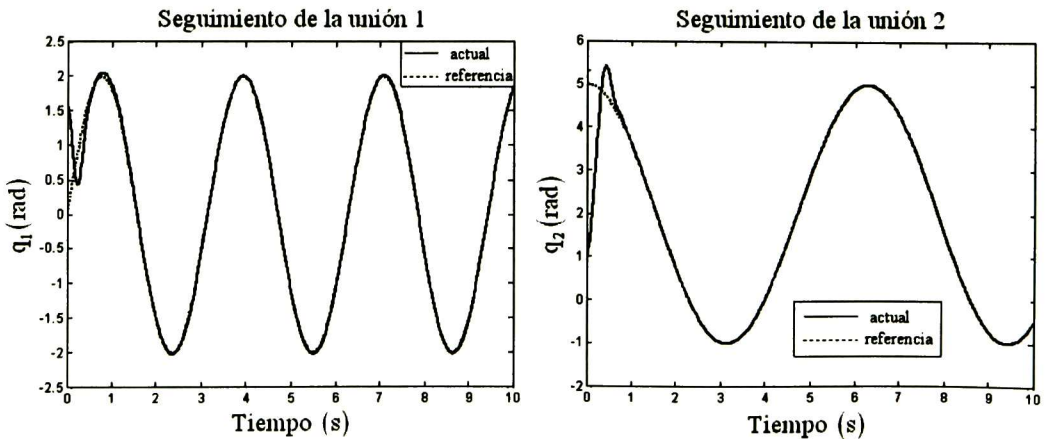


Figura 38. STA: Respuesta de seguimiento para la 1<sup>era</sup> unión (izq.) y la 2<sup>da</sup> (der.)

Las variables de error de seguimiento, y el retrato de fase de las mismas, se pueden observar en la Figura 39. Es de notar la convergencia de los errores a una vecindad del origen del plano de fase de error, y la disminución drástica de oscilaciones en comparación con el control por MDIA. Las señales de control se pueden observar en la Figura 40, además de las velocidades angulares de las uniones. Se percibe una reducción de más del 50% en la magnitud de las señales de control en el período transitorio, en comparación con el control por MDIA.

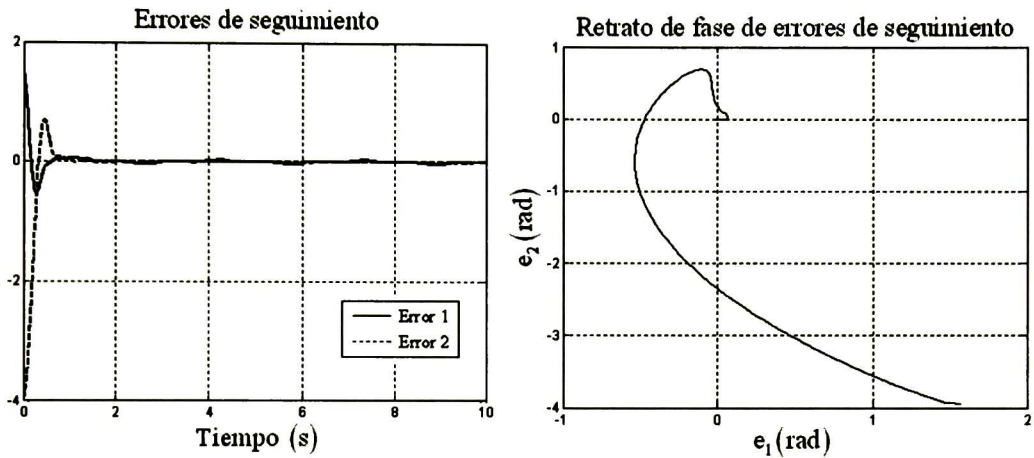


Figura 39. STA: Variables de error (izq.) y retrato de fase de errores de seguimiento (der.)

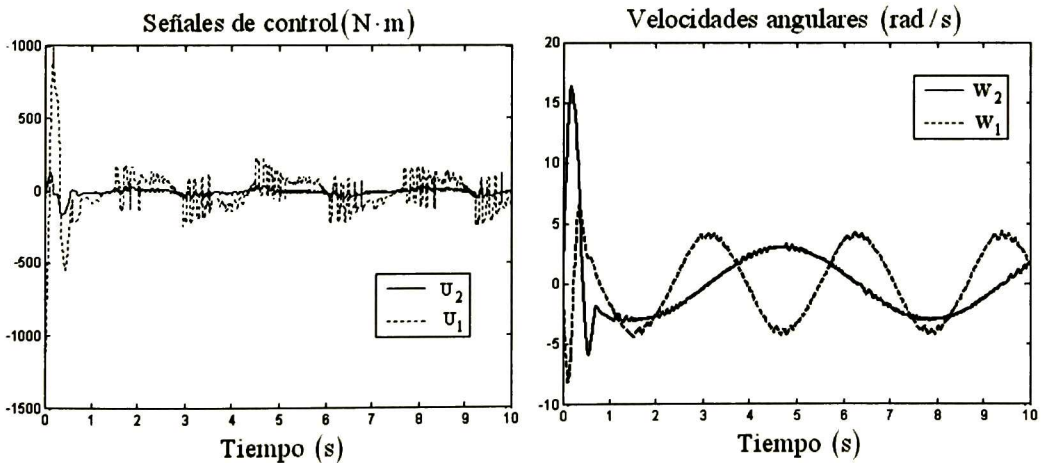


Figura 40. STA: Señales de control (izq.) y velocidades angulares (der.) de las uniones.

## 4.4. Control por STIA

Los parámetros utilizados en la simulación del controlador por STIA, desarrollado en la sección 3.7, fueron diseñados con los valores

$$k_1 = 5, \quad c_1 = 10, \quad \varepsilon_1 = 6, \quad c_2 = 10, \quad W_1 = 80, \quad W_2 = 6, \quad \rho = \frac{1}{2}. \quad (4.4.1)$$

Los resultados de la simulación pueden ser apreciados en las siguientes figuras.

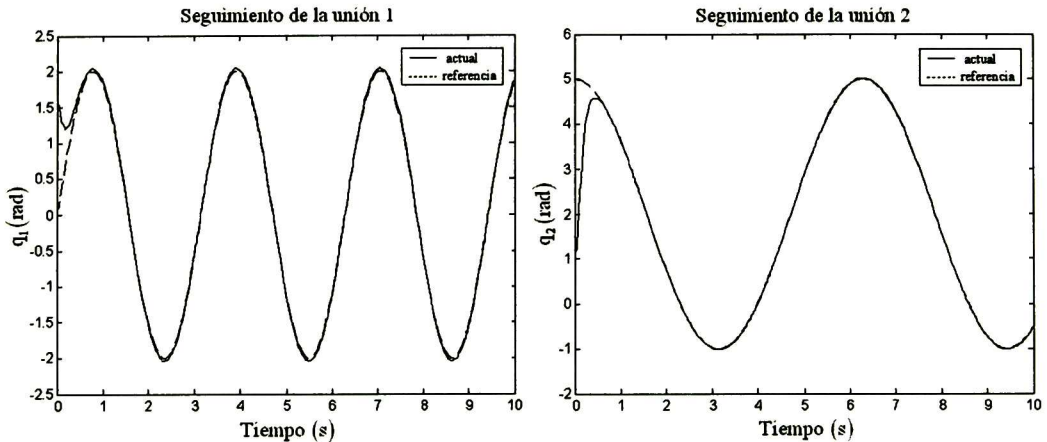


Figura 41. STIA: Respuesta de seguimiento para la 1<sup>ra</sup> unión (izq.) y la 2<sup>da</sup> (der.)

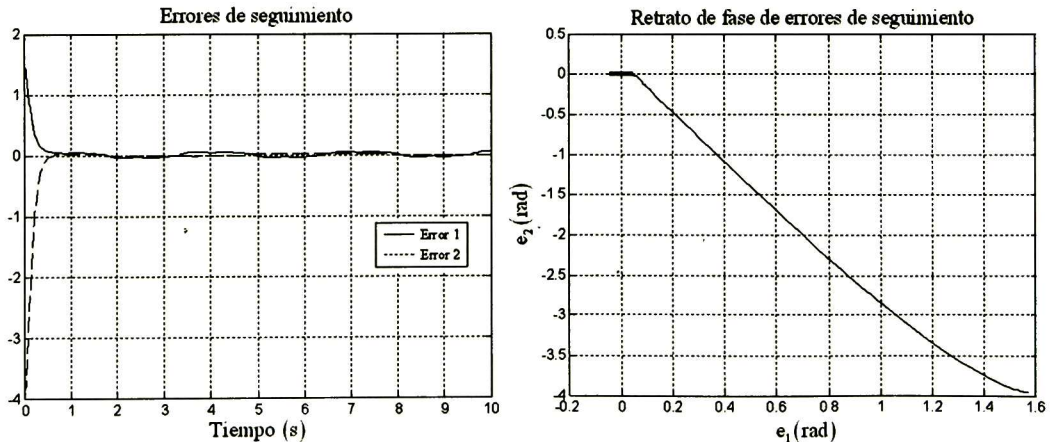


Figura 42. STIA: Variables de error (izq.) y retrato de fase de errores (der.)

En la Figura 41 se observan las posiciones angulares de ambas uniones. Se aprecia que además de cumplir con el objetivo de control, se obtiene una señal

suave y libre de *chattering*, con un tiempo de estabilización pequeño, en comparación con el control por MDIA. La Figura 42 muestra los errores de seguimiento contra el tiempo, y el retrato de fase de los mismos. Se puede notar su convergencia en tiempo finito a una vecindad de cero de forma suave y con un tiempo de estabilización pequeño, en comparación con los controles por MDIA y STA.

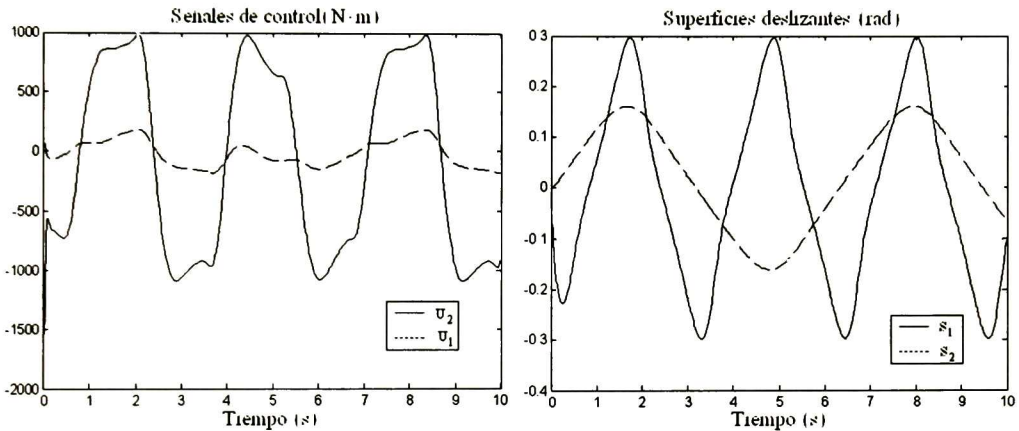


Figura 43. STIA: Señales de control (izq.) y superficies deslizantes (der.)

Las señales de control para cada una de las uniones se muestran en la Figura 43, al igual que las superficies deslizantes. Se aprecia como las superficies, desde el instante inicial, están dentro de la región de atracción del sistema en lazo cerrado.

## 4.5. Análisis Comparativo

En la Figura 44 se muestra una tabla con algunas características de la respuesta en lazo cerrado de los controladores por MDIA, STA y STIA. Se aprecia que los tres controladores propuestos logran el objetivo de control: el seguimiento de trayectorias con perturbaciones *no-matched* en el sistema. Sin embargo, se pueden apreciar algunas diferencias.

El tiempo de estabilización de la respuesta en lazo cerrado del control por MDIA es el doble que el de los controles por STA y STIA. Además, el pico máximo de la magnitud de la señal de control por MDIA es más de 4 veces mayor que para



los otros dos controladores propuestos. Las vecindades de convergencia del error son equiparables para los tres controles, sin embargo, el control por STIA es superior a los demás debido a la ausencia de alta frecuencia en la señal de control. Esta es una característica deseable para cualquier control que se desea implementar en tiempo real, ya que la presencia de alta frecuencia en las señales de control genera desgaste prematuro o incluso la descompostura de los actuadores.

	MDIA	STA	STIA
Tiempo de estabilización	2 s	1 s	1 s
Pico Máximo de Control	4,500 Nm	1,000 Nm	1,000 Nm
Alta Frecuencia en Señal de Control	Si	Si	No
Vecindad de Convergencia de Error	0.01	0.08	0.05
Rechaza Perturbaciones <i>No-Matched</i>	Si	Si	Si

Figura 44. Tabla comparativa entre los controladores MDIA, STA y STIA en simulación.

## 4.6. Parámetros del Modelo Discreto

El modelo de cinemática diferencial usado en simulación corresponde al de una unidad PTU (*pan-tilt unit*) que se encuentra en el laboratorio de Control Automático del Cinvestav, Unidad Gdl. Un PTU es un sistema de visión estéreo montado en una cadena cinemática de 2 grados de libertad (*pan* y *tilt*) como se muestra en la Figura 45.

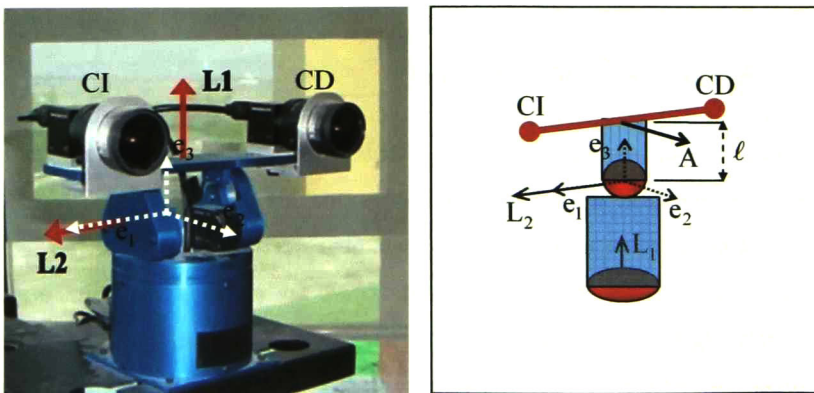


Figura 45. Unidad *Pan-Tilt* con sus ejes de rotación (izq.) y su esquema geométrico (der.)

El vector de posiciones angulares se definió como

$$\theta = [\theta_1 \quad \theta_2]^T \quad (4.5.1)$$

Debido a que el sistema solo posee uniones rotativas se tiene que  $D_i = L_i$  para  $i = \{1, 2\}$ . Entonces, para definir por completo dicho modelo, se deben definir los ejes de rotación del sistema en la posición inicial, los cuales están dados por

$$L_{1,0} = e_{12} \quad L_{2,0} = e_{23}.$$

La tarea a realizar por la PTU es el seguimiento visual de objetos, esto es, lograr que la orientación del SVE (definida en la Figura 45 como A) coincida con la orientación deseada (definida por la posición del objetivo a seguir, la cual se obtiene del procesamiento de las imágenes del SVE). Por lo tanto, es necesario definir la posición inicial de la orientación del SVE, la cual se define como

$$A_0 = e_{31} + \ell e_3 e_\infty.$$

Ahora, la cinemática diferencial de la orientación actual del SVE ( $A'$ ) se obtiene usando (3.2.17), resultando en

$$\dot{A}' = J(\theta) \dot{\theta} \quad J(\theta) = [\alpha_1 \quad \alpha_2] \quad (4.5.2)$$

donde  $\alpha_j = \frac{1}{2}(A' L'_j - L'_j A')$  para  $i = \{1, 2\}$ . Cabe notar que los términos  $A'$  y  $L_j$  se obtienen por cinemática directa, tal como se describe en la sección 3.2.1.

Finalmente, para obtener el modelo discreto de (4.5.2) se utiliza el método de discretización de Euler (como se vio en la sección 3.2.2) resultando en

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + T u_k \\ x_{2,k+1} &= x_{2,k} + T J(x_{1,k}) u_k \\ y_k &= x_{2,k} \end{aligned} \quad (4.5.3)$$

donde  $x_{1,k} = q_k$  y  $x_{2,k} = A'_k$  son las variables de estado y representan a las posiciones angulares de unión y la orientación del efector final, respectivamente,  $u_k$  es el término de control,  $J(x_{1,k})$  es la discretización del jacobiano del sistema,  $T$  es el período de muestreo y  $y_k$  es la salida del sistema.

La variable de error del sistema se define como

$$e_k = A'_k - A_{k,ref}. \quad (4.5.4)$$

Para lograr que la referencia  $A_{k,ref}$  correspondiera al comportamiento real de un objetivo en movimiento, se realizó un experimento de procesamiento de imágenes con el SVE mostrado en la Figura 45, el cual consta de dos cámaras *Flea*® de

*Point Grey Research Inc.* montadas sobre un manipulador de dos GDL. Una secuencia de imágenes del experimento puede ser observada en la Figura 46.

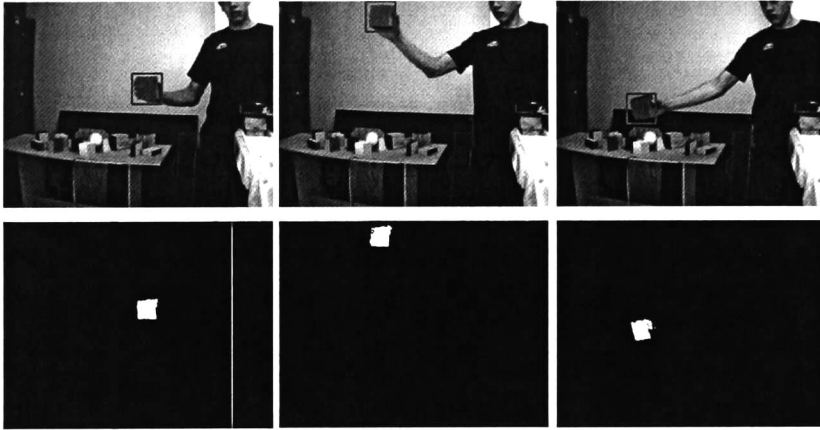


Figura 46. Secuencia de imágenes para obtener la orientación de referencia.

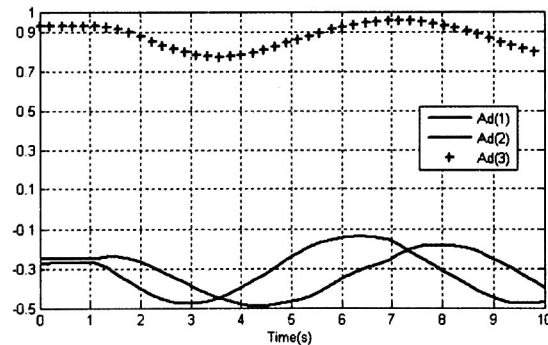


Figura 47. Componentes euclidianas de la orientación de referencia obtenida.

En la parte inferior de dicha figura se observa el resultado binario de la segmentación por color realizada al objetivo (un cubo rojo), donde los *pixeles* blancos son los que el algoritmo de segmentación definió como pertenecientes al objetivo. La parte de arriba muestra la imagen original con el objetivo encerrado en un cuadro. Las componentes euclidianas de la orientación de referencia obtenida se muestran en la Figura 47.

El vector de perturbaciones usado en las simulaciones de los controles discretos (MDID y STD) está dado por

$$d_k = [0.5 \quad 0.2 + 0.3 \sin(\pi k T) \quad -3 + 0.2 \cos(k T)]^T$$

las condiciones iniciales se definieron como

$$\theta_0 = [\pi/2 \quad , \quad \pi/3]^T$$

y el período de muestreo utilizado es de  $T = 220$  milisegundos, el cual fue elegido de acuerdo al tiempo que toma procesar las imágenes del SVE.

## 4.7. Control por MDID

La ley de control por MDID, diseñada en la sección 3.9, está definida por

$$\mathbf{u}_k = \mathbf{u}_{k0} + \mathbf{u}_{kl} \quad (4.6.1)$$

donde

$$\mathbf{u}_{kl} = -\mathbf{J}^+(\mathbf{x}_{1,k}) \left[ \frac{1}{T} \mathbf{P}^+ \mathbf{s}_k - \mathbf{J}(\mathbf{x}_{1,k-1}) \mathbf{u}_{(k-1)l} \right],$$

$$\mathbf{u}_{k0} = -\frac{\mathbf{J}(\mathbf{x}_{1,k})^+}{T} [\mathbf{x}_{2,k} - \mathbf{K}_1 \mathbf{e}_k].$$

Los parámetros de control, utilizados en simulación, que definen a (4.6.1) son

$$\mathbf{K}_1 = \frac{1}{2} \begin{bmatrix} -0.4 & 0 & 0 \\ 0 & -0.5 & 0 \\ 0 & 0 & -0.3 \end{bmatrix},$$

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Con fines de comparación, también se simuló la respuesta en lazo cerrado del sistema con un controlador Proporcional-Integral-Derivativo discreto (PID) paralelo, el cual está definido por la siguiente ecuación

$$\mathbf{u}_k = \mathbf{u}_{k-1} - \mathbf{J}_k(\mathbf{x}_{1,k})^+ \left[ \mathbf{K}_p t_s \mathbf{e}_k + \frac{\mathbf{K}_i}{t_s} (\mathbf{e}_k - 2\mathbf{e}_{k-1} + \mathbf{e}_{k-2}) + \mathbf{K}_d (\mathbf{e}_k - \mathbf{e}_{k-1}) \right]$$

Los parámetros del PID usado en simulación son

$$\mathbf{K}_p = 15,$$

$$\mathbf{K}_i = 0.5,$$

$$\mathbf{K}_d = 1.$$

Estos parámetros fueron elegidos para asegurar la estabilidad del sistema en lazo cerrado. Ahora, describiremos las figuras que muestran la respuesta en lazo cerrado del controlador propuesto y del PID.

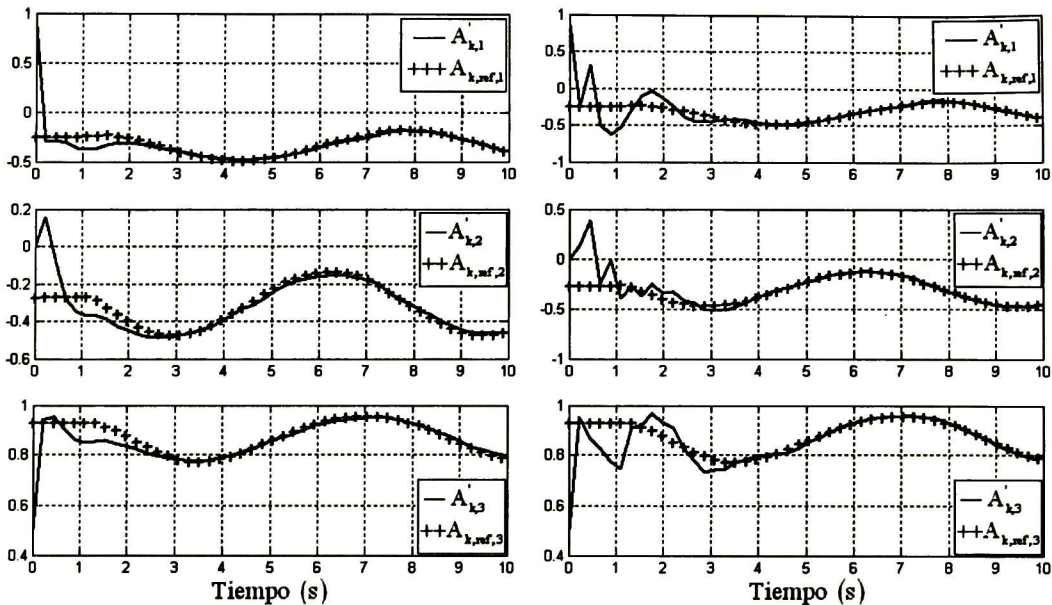


Figura 48. Orientación y referencia para MDID (izq.) y PIDD (der.)

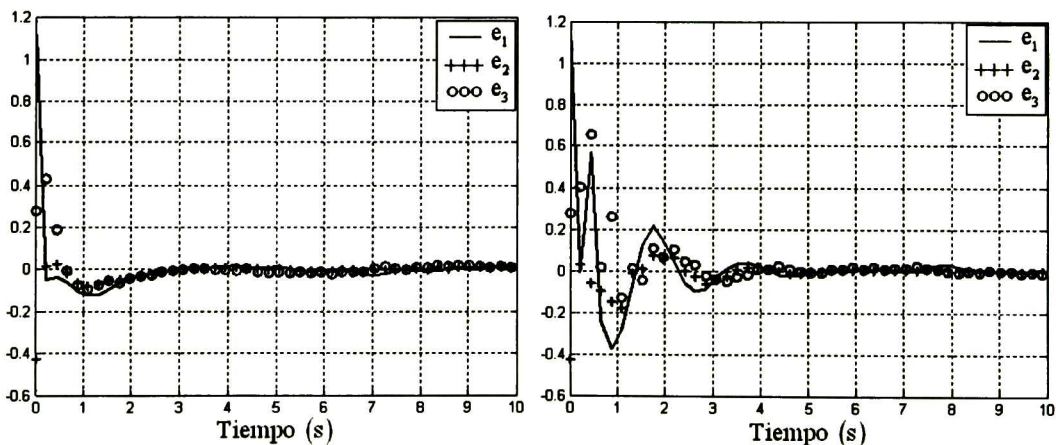


Figura 49. Variables de error para MDID (izq.) y PIDD (der.)

En la Figura 48 se muestran las componentes euclidianas del vector de orientación y sus respectivas referencias para el control MDID y PIDD. Se observa que ambos controladores cumplen con el objetivo de control, sin embargo el desempeño del control por MDID es menos oscilatorio, tiene menor tiempo de estabilización y menos sobre-impulso que el PIDD. Las variables de error para

ambos controladores se muestran en la Figura 49, donde también se puede apreciar una mejor respuesta para el controlador basado en modos deslizantes. La Figura 50 muestra las señales de control (en este caso las velocidades angulares del PTU) para ambos controladores.

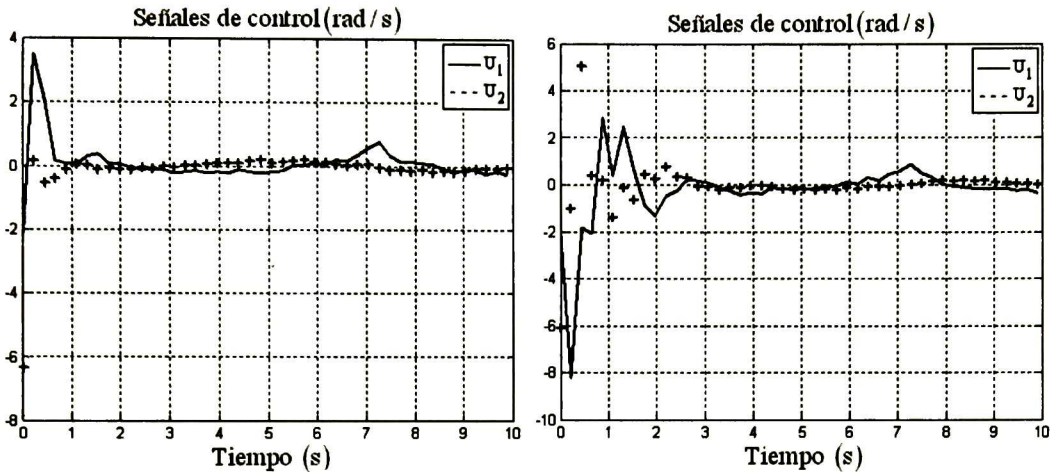


Figura 50. Señales de control para MDID (izq.) y PIDD (der.)

## 4.8. Control STD

El control STD diseñado en la sección 3.10 está definido por

$$u_k = -\frac{J(x_{1,k})^+}{T} \left[ x_{2,k} + L \|s_k\|^\rho \text{sigm}(s_k, \varepsilon_1) - u_{2,k} \right], \quad (4.6.2)$$

$$u_{2,k+1} = -WT \text{sigm}(s_k, \varepsilon_2) + u_{2,k}.$$

Los parámetros del control STD utilizado en simulación son

$$L = -0.3 \quad W = 0.2 \quad \varepsilon_1 = 40 \quad \varepsilon_2 = 6 \quad \rho = 1/2. \quad (4.6.3)$$

Las siguientes figuras muestran el comportamiento del sistema en lazo cerrado.

En la parte izquierda de la Figura 51 se muestran los vectores de orientación y sus respectivas referencias para el control STD. Se puede apreciar que se cumple el objetivo de control, ya que se realiza satisfactoriamente el seguimiento de la referencia. La convergencia de las variables de error a una vecindad de cero se puede apreciar en la parte izquierda de la Figura 51. Se nota también que el desempeño del controlador es muy superior al del PIDD (menos oscilatorio en su

etapa transitoria y menor tiempo de estabilización) y ligeramente mejor que el control por MDID, ya que converge a una vecindad menor de cero.

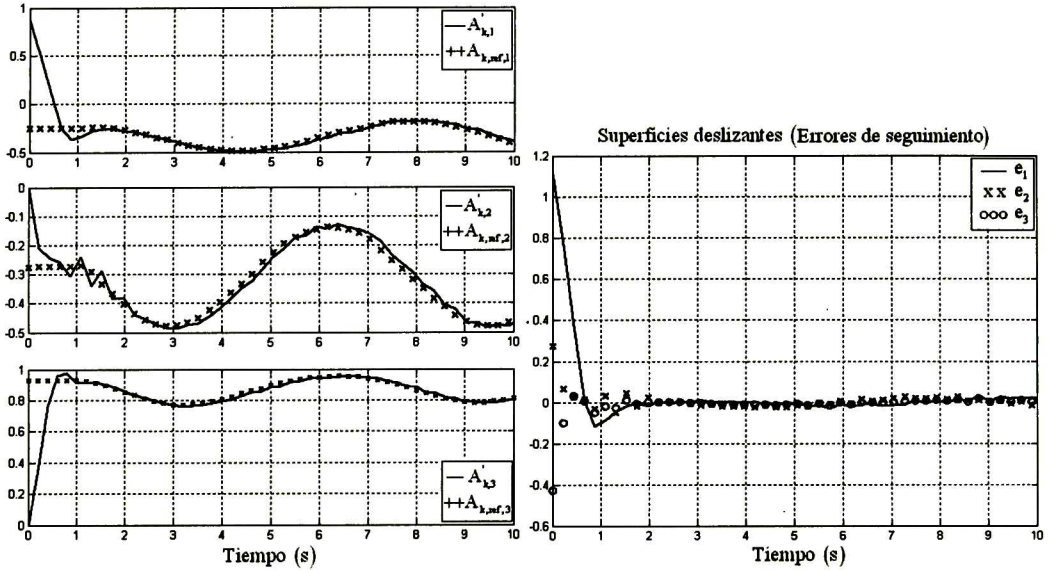


Figura 51. STD: Componentes del vector de orientación  $A_k$  y su referencia  $A_{k,ref}$  (izq.) y las superficies deslizantes (der.)

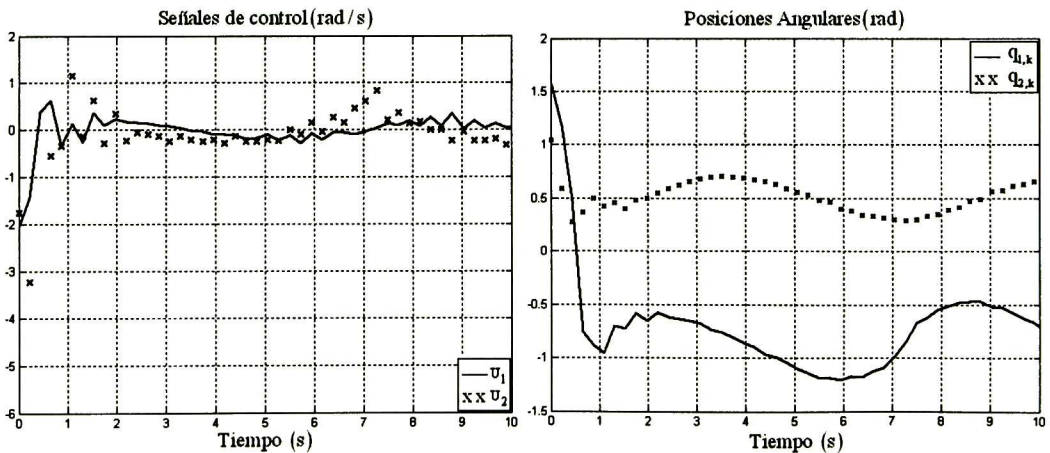


Figura 52. STD: Señales de control (izq.) y posiciones angulares (der.)

Las señales de control se observan en la parte izquierda de la Figura 52 y en la parte derecha las posiciones angulares. De nuevo, las señales de control son más

suaves que el PIDD y con un menor pico máximo respecto al control por MDID y al PIDD.

## 4.9. Análisis Comparativo

Un análisis de las secciones 4.7 y 4.8 permite concluir que los tres controladores logran el seguimiento de las trayectorias de orientación para el SVE, aún en presencia de perturbaciones. Sin embargo, existen algunas diferencias en la respuesta en simulación en lazo cerrado de los controladores PIDD, MDID y STD, las cuales se pueden observar en la tabla de la Figura 53.

Primero, los tres controladores logran la convergencia de la señal de error a una vecindad similar y pudieron rechazar el término de perturbaciones agregado en el sistema. Sin embargo, el tiempo de estabilización es menor para el control STD que para los otros dos controladores. Además, el pico máximo de la señal de control es mayor para el control PIDD, mientras que el menor valor lo mostró la respuesta del control STD. Finalmente, el PIDD fue el único controlador que mostró oscilaciones durante el período transitorio de la respuesta en lazo cerrado. Lo anterior, permite concluir que el mejor desempeño lo tuvo el control STD.

	PIDD	MDID	STD
Tiempo de Estabilización	4 s	3 s	2 s
Pico Máximo de Control	8 rad/s	6.3 rad/s	5.5 rad/s
Oscilaciones Durante el Transitorio	Si	No	No
Vecindad de Convergencia de Error	0.04	0.04	0.03
Rechaza Perturbaciones	Si	Si	Si

Figura 53. Tabla comparativa entre los controladores PIDD, MDID y STD en simulación.





## 5. Implementación en Tiempo Real

Para poder realizar la implementación en tiempo real de los controles discretos propuestos, es necesario realizar la calibración Mano-Ojo descrita en la sección 2.5.4. Dicho procedimiento obtiene la transformación entre el sistema de coordenadas fijo a una de las cámaras (usualmente la izquierda) y el sistema de coordenadas del último eslabón del sistema robótico. La Figura 54 muestra los sistemas robóticos y sistemas de visión donde se implementaron los controladores, así como los sistemas de coordenadas relacionados por la calibración Mano-Ojo en los mismos.

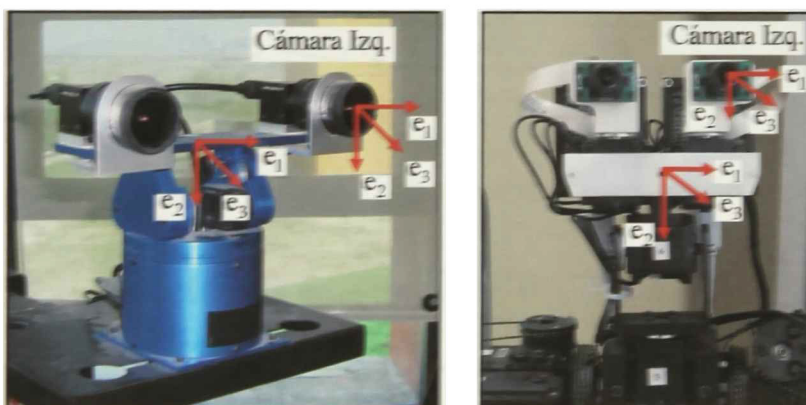


Figura 54. Sistemas de coordenadas relacionados por la calibración Mano-Ojo en el sistema robótico para MDID (izq.) y STD (der.)

El control por MDID se implementó en una unidad PTU (Figura 54, parte izquierda) y el control STD en la cabeza de un robot humanoide (Figura 54, parte derecha), ambos con dos grados de libertad. A continuación se presentan gráficas e imágenes obtenidas del comportamiento en lazo cerrado de los controles discretos diseñados.

## 5.1. Control por MDID en Tiempo Real

En la Figura 55 se observan las posiciones angulares para cada uno de las uniones del PTU, además de las variables de error para el experimento en tiempo real. Nótese la convergencia de dichas variables a una vecindad de cero. Las superficies deslizantes y las velocidades angulares (señales de control) se muestran en la Figura 56. Las componentes del vector de orientación y del vector de referencia se observan en la Figura 57. Podemos observar el desempeño satisfactorio del controlador.

En la Figura 58 se muestra una secuencia de imágenes obtenidas del experimento en tiempo real. El objetivo (en este caso un taladro) ha sido segmentado en las imágenes y encerrado en un cuadro rojo, además de que el algoritmo de control ha mantenido a dicho objetivo dentro del campo de visión del sistema estéreo, el cual era el objetivo de control.

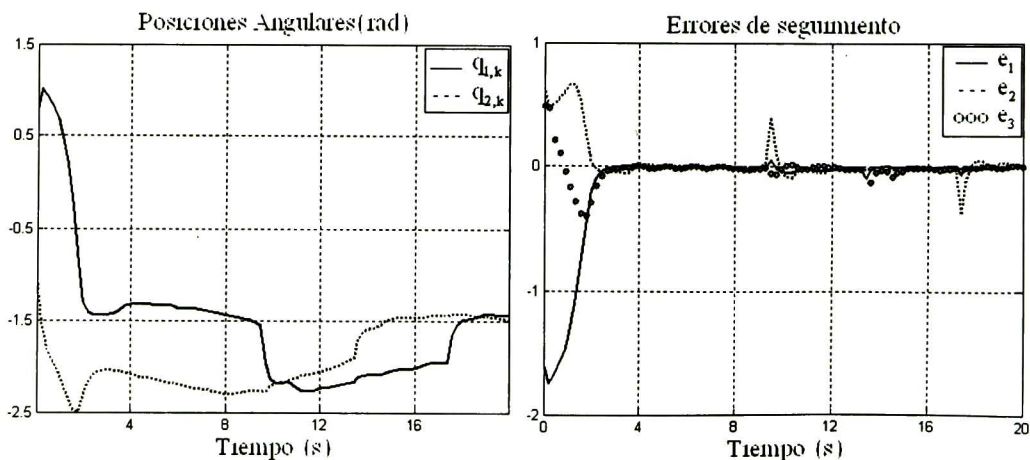


Figura 55. Posiciones angulares (izq.) y variables de error (der.) en tiempo real.

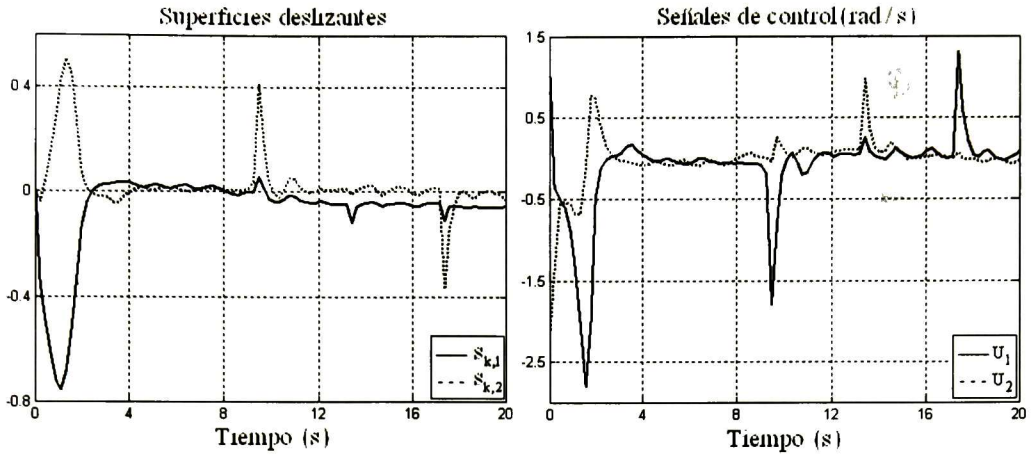


Figura 56. Superficies deslizantes (izq.) y señales de control (der.) en tiempo real.

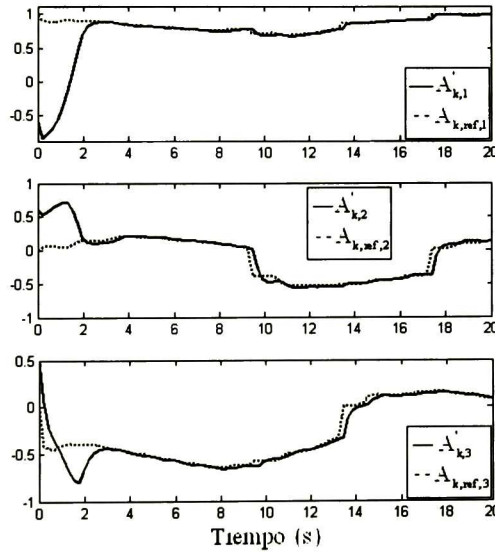


Figura 57. Componentes euclidianas de la orientación y sus referencias en tiempo real.

## 5.2. Control STD en Tiempo Real

El comportamiento en lazo cerrado del STD en tiempo real se muestra en las siguientes figuras. En la parte izquierda de la Figura 59 se observan las velocidades angulares del sistema, y en la parte derecha sus posiciones angulares.



Figura 58. Imágenes de la cámara izquierda del experimento en tiempo real para MDID.

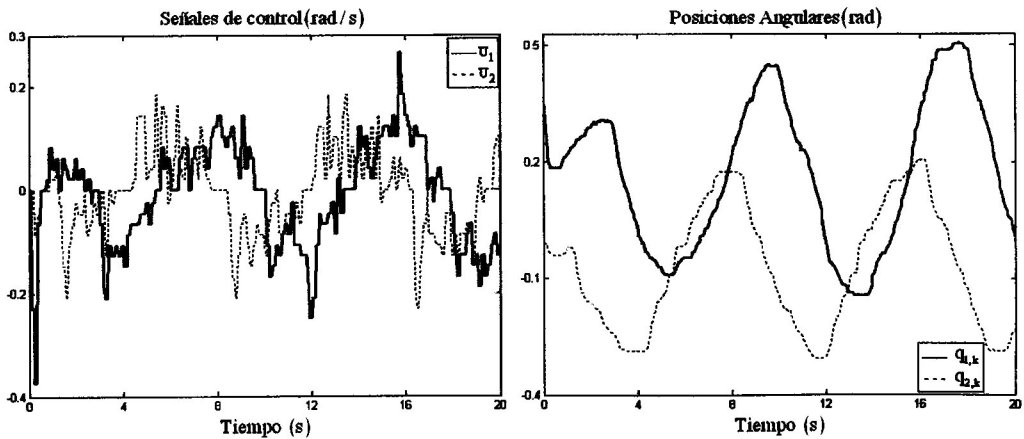


Figura 59. Señales de control (der.) y posiciones angulares (izq.) para STD en tiempo real.

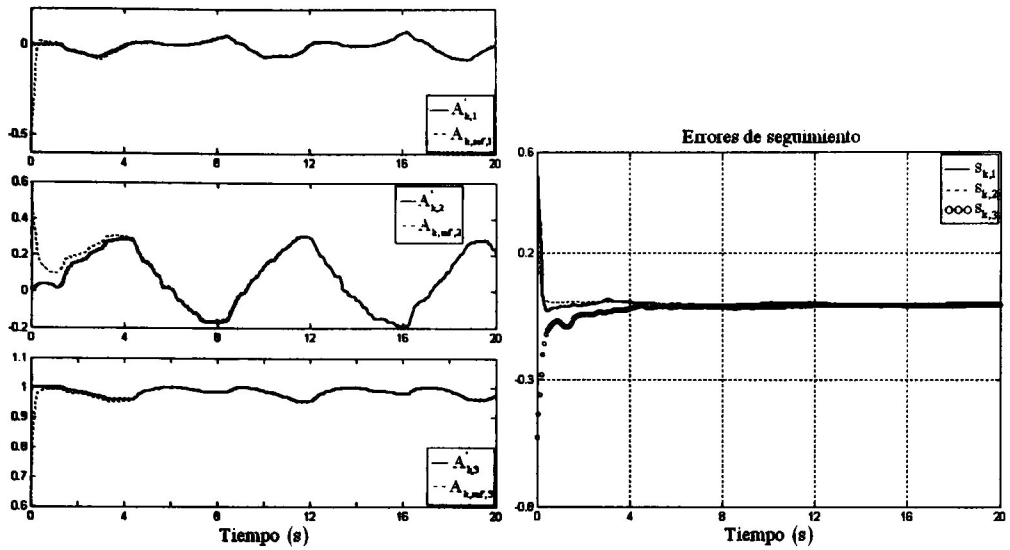


Figura 60. Orientación y referencia (izq.) y errores (der.) para STD en tiempo real.

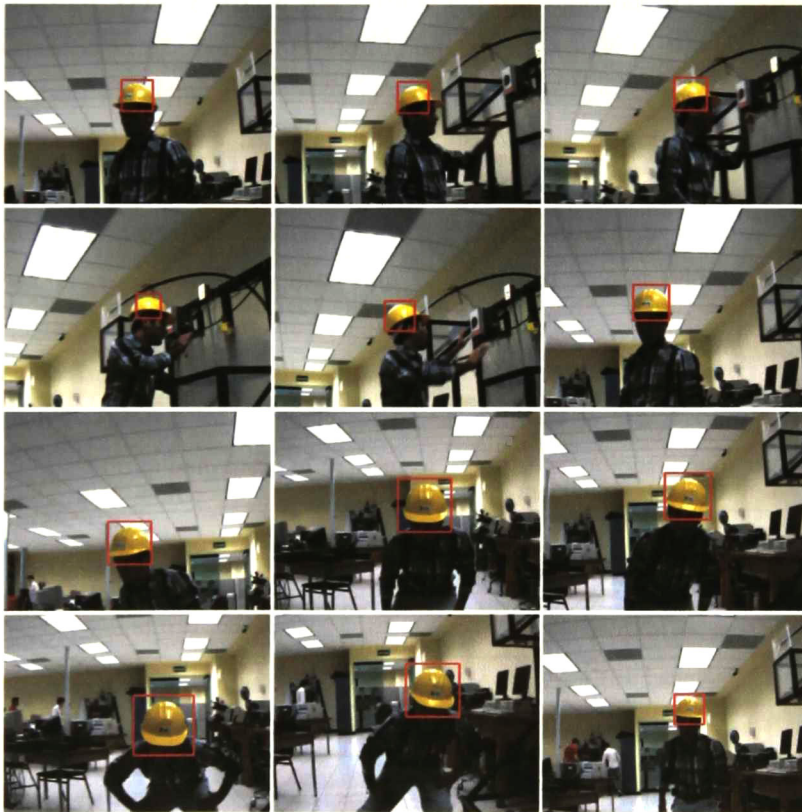


Figura 61. Imágenes de la cámara izquierda del experimento en tiempo real para STD.

Las velocidades, que son las señales de control, se mantienen con una magnitud por debajo de los 0.35 rad/s y no presentan componentes de alta frecuencia. Las componentes euclidianas de la orientación del SVE, definida por una línea en AGC, y sus referencias se aprecian en la parte izquierda de la Figura 60. En la parte derecha se observan las variables de error de seguimiento, las cuales convergen a una vecindad de cero. Note que el objetivo de control se logra, ya que el sistema en lazo cerrado realiza el seguimiento de la referencia de orientación. Finalmente, una secuencia de imágenes de la cámara izquierda del SVE del humanoide se muestra en la Figura 61. En este caso, el objetivo a seguir era un casco protector de color amarillo.

### 5.3. Análisis Comparativo

Analizando el comportamiento del control por MDID y STD, mostrados en las secciones 5.1 y 5.2, se puede concluir que ambos controladores cumplen el objetivo de control al realizar el seguimiento de trayectorias de orientación para el SVE del sistema robótico. Por otro lado, los comportamientos en lazo cerrado de ambos controladores muestran algunas diferencias importantes, las cuales se pueden observar en la tabla mostrada en la Figura 62. Primero, el pico máximo de la señal de control es mucho mayor para MDID que para STD. Además, la vecindad de convergencia de las señales de error es menor para STD. Luego, los dos controladores presentan un tiempo de convergencia a la vecindad de error muy similar y logran rechazar el término de perturbaciones del sistema. Finalmente, se puede concluir que el mejor desempeño lo mostró el controlador STD.

	MDID	STD
<b>Tiempo de estabilización</b>	3 s	3 s
<b>Pico Máximo de Control</b>	2.7 rad/s	0.38 rad/s
<b>Vecindad de Convergencia de Error</b>	0.4	0.08
<b>Rechaza Perturbaciones</b>	Si	Si

Figura 62. Tabla comparativa entre los controladores MDID y STD en tiempo real.

## 6. Conclusiones y Trabajo Futuro

En esta sección se mencionarán las conclusiones más importantes del trabajo realizado y algunos aspectos a mejorar o ampliar en el futuro.

### 6.1. Conclusiones

Durante el desarrollo de esta tesis se propusieron soluciones al problema de control de mecanismos robóticos guiados por visión. Las técnicas propuestas pertenecen principalmente a tres áreas: modelado usando Algebra Geométrica Conformal (AGC), diseño de controladores continuos por modos deslizantes y diseño de controladores discretos por modos deslizantes.

*Modelado en AGC:* Se desarrollaron los modelos de cinemática directa, cinemática inversa, cinemática diferencial y dinámica en AGC, para mecanismos robóticos seriales con uniones prismáticas o revolutivas. Se mostró que el procedimiento para obtener dichos modelos es simple y compacto. Lo anterior se debe a la facilidad con la que se pueden representar entidades geométricas (puntos, líneas, planos y esferas) y transformaciones (proyecciones, rotaciones, traslaciones y movimientos de tornillo) en el marco de la geometría conformal. Además, bajo la misma estrategia de modelado, se pueden modelar robots con configuraciones más complejas, o las interacciones entre varios mecanismos robóticos.



Otro factor importante a considerar, es el costo computacional del uso del AGC en el modelado de mecanismos robóticos. Para representar una rotación usando un *rotor* solo se definen 3 coeficientes, y para obtener una composición de 2 rotaciones solo se necesita multiplicar 9 veces y 3 sumas de tres elementos. Usando matrices se necesitan definir 9 coeficientes y la composición de 2 rotaciones requiere de 27 multiplicaciones y 9 sumas de tres elementos. Eso solo considerando la composición de 2 rotaciones. Evidentemente, se necesita un análisis más formal para obtener conclusiones precisas sobre el costo computacional entre AGC y el álgebra matricial, y sería justo incluir a los cuaterniones en este análisis, el cual se propone como trabajo futuro.

Es la opinión de quien presenta este trabajo que, conforme se vaya incrementando el uso del álgebra geométrica conformal, se apreciará mejor la potencialidad que esta tiene en la solución de problemas, tanto en el área de robótica como en otras disciplinas.

*Control Continuo por Modos Deslizantes:* Se diseñaron tres controladores continuos para los mecanismos robóticos: por modos deslizantes integrales anidados (MDIA), por algoritmo super-twisting anidado (STA), y por algoritmo super-twisting integral anidado (STIA). Los resultados se obtuvieron solo en simulación debido a la falta de un prototipo robótico adecuado para probar las señales de control de esfuerzo de torsión obtenidas por los algoritmos de control.

Dichos resultados mostraron que los tres controladores son robustos ante perturbaciones externas, variaciones paramétricas e incertidumbres de modelado, con un seguimiento satisfactorio de la referencia de salida. Sin embargo, existieron diferencias entre el comportamiento en lazo cerrado de cada uno de los algoritmos.

La más significativa es la disminución de oscilaciones en la respuesta del STA y la ausencia de ella con el control STIA, en comparación con el MDIA. Esto se debe al uso de una ley de control totalmente continua en el STIA, lo que genera una señal de control libre de *chattering*.

Otro aspecto a resaltar es que aún cuando las vecindades de convergencia son comparables, el control que utiliza el algoritmo de Super Twisting tiene un menor tiempo de convergencia y asentamiento.

El uso de la estrategia de control por modos deslizantes anidados, permite que todos los controladores diseñados rechacen las perturbaciones *matched* y *no matched* que presentaban los sistemas de error del sistema a controlar. Por otro lado, el uso de la filosofía de modos deslizantes integrales reduce la magnitud de las ganancias del controlador.

Lo anterior permite que el algoritmo que conjuga las tres estrategias de control (modos deslizantes anidados, integrales y algoritmo *super-twisting*) tenga el mejor desempeño en lazo cerrado.

*Control Discreto por Modos Deslizantes:* Finalmente, se diseñaron dos controladores discretos para el modelo de cinemática diferencial: control por modos deslizantes integrales discretos (MDID) y control *super-twisting* discreto (STD). El desempeño de estos dos controladores se obtuvo en simulación y en tiempo real. También se simuló la respuesta de un control PID discreto (PIDD) con fines de comparación. Los resultados en simulación nos permiten concluir que el uso de modos deslizantes robustece la respuesta de los controles por MDID y STD, en comparación con el control PIDD. Además, mejora la respuesta en factores como tiempo de asentamiento, suavidad y magnitud de la señal de control, y vecindad de convergencia de las variables de error. Esto se ve reflejado en el desempeño de dichos controladores en los experimentos en tiempo real.

Cabe mencionar, que la aplicación de los algoritmos de control propuestos puede ser extendida a manipuladores seriales de  $n$  grados de libertad y/o a mecanismos robóticos más complejos.

Basado en las conclusiones anteriores, se puede resumir que la conjunción del modelado por AGC y el control por modos deslizantes, es una buena opción para resolver un problema tan complejo como es el control de mecanismos robóticos.

## 6.2. Trabajo Futuro

Como trabajo futuro se pueden mencionar los siguientes puntos:

- Mejorar la etapa de localización con un algoritmo más robusto.
- La demostración de estabilidad del control STD en lazo cerrado.

- Realizar un análisis comparativo sobre el costo computacional del uso del AGC y otras alternativas como álgebra matricial y cuaterniones.
- Implementar en tiempo real los algoritmos de control continuo diseñados.
- Abordar y solucionar el problema de retardos en el sistema al realizar implementaciones en tiempo real.
- Implementar en mecanismos más complejos, y en tiempo real, los controladores propuestos.

## 7 Publicaciones

### **En Revista:**

- *Luis E. González-Jiménez, Alexander G. Loukianov y Eduardo Bayro-Corrochano*, “Discrete Integral Sliding Mode Control In Visual Object Tracking”, *IJHR*, 2010, pp. 491-509.

### **En Conferencia:**

- *Luis E. González-Jiménez, Alexander G. Loukianov y Eduardo Bayro-Corrochano*, “Integral Nested Sliding Mode Control for Robotic Manipulators”, *IFAC World Conference*, Seúl, Korea, 2008, pp. 9899-9904.
- *Luis E. González-Jiménez, Alexander G. Loukianov y Eduardo Bayro-Corrochano*, “Discrete Integral Sliding Mode Control In Visual Object Tracking Using Differential Kinematics”, *CIARP '09*, Jalisco, México, 2009, pp. 843-850.
- *Luis E. González-Jiménez, Alexander G. Loukianov y Eduardo Bayro-Corrochano*, “Integral Nested Super-Twisting Algorithm for Robotic Manipulators”, *IROS 2010 (International Conference on Intelligent Robots and Systems)*, Taipei, Taiwan, Octubre de 2010, pp. 3580-3585.
- *Luis E. González-Jiménez, Eduardo Bayro-Corrochano y Alexander G. Loukianov*. “Fully Nested Super-Twisting Algorithm for Uncertain Robotic

Manipulators”, ICRA 2011 (International Conference on Robotics and Automation), Shanghai, China, Mayo de 2011, pp. 5807-5812.

- *Luis E. González-Jiménez, Oscar Eleno Carbajal-Espinosa y Eduardo Bayro-Corrochano* “Geometric Techniques for the Kinematic Modeling and Control of Robotic Manipulators”, ICRA 2011 (International Conference on Robotics and Automation), Shanghai, China, Mayo de 2011, pp. 5831-5836.
- *Oscar Eleno Carbajal-Espinosa, Luis E. González-Jiménez y Eduardo Bayro-Corrochano*, “Modeling and Control of a Humanoid Arm using Conformal Geometric Algebra and Sliding Modes”, Humanoids 2011 (11th IEEE-RAS International Conference on Humanoid Robots), Bled, Eslovenia, Octubre de 2011.

## 8. ANEXOS

### 8.1. Teorema de Estabilidad de Lyapunov

La estabilidad de un sistema no lineal, en el sentido de Lyapunov, utiliza el concepto de la energía del sistema para formular condiciones que definan dicha estabilidad. Para ilustrar lo anterior considere el sistema no autónomo

$$\dot{x} = f(t, x) \quad (7.1.1)$$

donde  $f: [0, \infty) \times D \rightarrow \mathbb{R}^n$  es una función continua por partes en  $t$  y es Lipschitz en  $x$  sobre  $[0, \infty) \times D$ , y  $D \subset \mathbb{R}^n$  es un dominio que contiene al origen  $x=0$ . Entonces, el origen de (7.1.1) es un punto de equilibrio en  $t=0$  si

$$f(t, 0) = 0, \quad \forall t \geq 0. \quad (7.1.2)$$

Esta afirmación puede hacerse para cualquier punto de equilibrio que no esté en el origen, si existe una transformación que traslade ese punto de equilibrio al origen.

Los conceptos de estabilidad para el punto de equilibrio  $x=0$  del sistema (7.1.1) son:

**Estable.** Si para cada  $\varepsilon > 0$ , existe un  $\delta = \delta(\varepsilon, t_0) > 0$  tal que

$$\|x(t_0)\| < \delta \Rightarrow \|x(t)\| < \varepsilon, \quad \forall t \geq t_0 \geq 0 \quad (7.1.3)$$

**Uniformemente Estable.** Si para cada  $\varepsilon > 0$ , existe un  $\delta = \delta(\varepsilon) > 0$  independiente de  $t_0$ , tal que se satisface (7.1.3).

**Inestable.** Si no es estable.

**Asintóticamente Estable.** Si es estable y existe una  $c = c(t_0)$  tal que si  $t \rightarrow \infty$  entonces  $x(t) \rightarrow 0$  para todo  $\|x(t_0)\| < c$ .

**Uniformemente Asintóticamente Estable.** Si es uniformemente estable y existe una  $c > 0$ , independiente de  $t_0$ ; tal que si  $t \rightarrow \infty$  entonces  $x(t) \rightarrow 0$  para todo  $\|x(t_0)\| < c$ , uniformemente en  $t_0$ .

**Globalmente Uniformemente Asintóticamente Estable.** Si es uniformemente estable y para cada par de números positivos  $\varepsilon$  y  $c$ , existe una  $T = T(\varepsilon, c) > 0$  tal que

$$\|x(t)\| < \varepsilon, \quad \forall t \geq t_0 + T(\varepsilon, c), \quad \forall \|x(t_0)\| < c. \quad (7.1.4)$$

Ahora definimos la estabilidad del sistema (7.1.1) de la siguiente manera:

Sea  $x = 0$  un punto de equilibrio del sistema y  $D \subset \mathbb{R}^n$  un dominio conteniendo a  $x = 0$ . Sea  $V : [0, \infty) \times D \rightarrow \mathbb{R}^n$  una función continuamente diferenciable tal que

$$W_1(x) \leq V(t, x) \leq W_2(x) \quad (7.1.5)$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x) \leq -W_3(x) \quad (7.1.6)$$

$\forall t \geq 0, \forall x \in D$  donde  $W_1(x), W_2(x)$  y  $W_3(x)$  son funciones continuas positivas en  $D$ . Entonces  $x = 0$  es un punto de equilibrio **uniformemente asintóticamente estable**.

Si las condiciones (7.1.5) y (7.1.6) se satisfacen globalmente (para todo  $x \in \mathbb{R}^n$ ) y  $W_1(x)$  es radialmente no acotada, entonces el punto de equilibrio  $x = 0$  es **globalmente uniformemente asintóticamente estable**.

Ahora, suponga que las condiciones (7.1.5) y (7.1.6) se satisfacen con

$$W_1(x) \geq k_1 \|x\|^c \quad W_2(x) \geq k_2 \|x\|^c \quad W_3(x) \geq k_3 \|x\|^c$$

para algunas constantes positivas  $k_1, k_2, k_3$  y  $c$ . Entonces  $x = 0$  es un punto de equilibrio **exponencialmente estable**. Si además lo anterior se cumple globalmente, entonces es **globalmente exponencialmente estable**. A la función  $V$ , usada en la última forma que establecimos la estabilidad del sistema (7.1.1), se le conoce como **función de Lyapunov**.

## 8.2. Discretización de Euler

Es el método que consiste en transformar un *sistema continuo* a uno discreto expresado en *ecuaciones de diferencias*, mediante la definición

$$\dot{x}(t) \triangleq \frac{x(kT+T) - x(kT)}{T} \quad (7.1.7)$$

donde  $T$  es el período constante de discretización.

Durante este proceso, el *álgebra diferencial* se transforma en *álgebra de diferencias*. Por ejemplo, considere el siguiente sistema de ecuaciones diferenciales

$$\dot{x}(t) = f(x, u), \quad (7.1.8)$$

cuya discretización por Euler, utilizando (7.1.7), resultaría en las siguientes ecuaciones de diferencias

$$x(kT+T) = x(kT) + Tf(x(kT), u(kT)). \quad (7.1.9)$$

La condición necesaria para llevar a cabo este tipo de discretización se conoce como condición ZOH (*zero-order hold*), la cual está expresada por

$$u(t) = u(kT) \quad \forall t \in [kT, kT+T), \quad (7.1.10)$$

o en otras palabras, que  $u(t)$  permanezca constante entre períodos de muestreo. Existe otra forma de discretización, la cual consiste en obtener la expansión en series de Taylor del sistema (7.1.8), resultando en

$$x(kT+T) = x(kT) + \sum_{\alpha=1}^{\infty} \frac{T^{\alpha}}{\alpha!} x^{(\alpha)} \Big|_{kT} \quad (7.1.11)$$

Se puede apreciar que la discretización por Euler es un caso especial de (7.1.11), cuando se toma en cuenta solo el primer elemento de la sumatoria, es decir, cuando  $\alpha=1$ . A la ecuación (7.1.11) se le conoce como la *discretización exacta* de (7.1.8), y a (7.1.9) como *discretización aproximada*.



### 8.3. Glosario

**Algoritmo.** (del latín, *dixit algorithmus* y éste del matemático persa al-Jwarizmi) Es un conjunto ordenado y finito de operaciones que permite encontrar la solución de un problema. Es decir, que un algoritmo es un método para encontrar la solución a algún problema.

**Chattering (Castañeteo).** Es el efecto causado por implementaciones de controladores basados en *Modos Deslizantes*, caracterizado por la aparición de *oscilaciones de frecuencia y magnitud finita* en la respuesta del sistema en lazo cerrado. Estas oscilaciones son generadas por la *conmutación a alta frecuencia* del controlador por Modos Deslizantes que excita dinámicas no modeladas del sistema pero presentes en el lazo cerrado de control.

**Energía Cinética.** Es la energía que posee un cuerpo por encontrarse en movimiento y está definida por

$$E_p = \frac{1}{2}mv^2$$

donde  $m$  es la masa del cuerpo y  $v$  su velocidad.

**Energía Potencial.** Es la energía almacenada en un sistema, o una medida del trabajo que un sistema puede entregar. La energía potencial de un cuerpo se define como la energía que es capaz de generar un trabajo como consecuencia de la posición del mismo.

**Espacio de Estados.** Es una manera de describir sistemas de ecuaciones diferenciales. Esta representación tiene la siguiente estructura

$$\begin{aligned}\dot{X} &= f(X,t) + b(X,t)U \\ y &= h(X)\end{aligned}$$

donde  $X$  es el vector de estados,  $U$  es el vector de entradas,  $y$  es el vector de salidas y  $f(X,t)$  es una función de estados y tiempo al igual que  $b(X,t)$ .

**Fuerza Centrífuga.** Es la que se produce cuando se obliga a un cuerpo a girar. Para que un cuerpo gire hay que aplicar una fuerza hacia el centro, llamada fuerza centrípeta. La fuerza centrífuga es la fuerza en oposición a la fuerza centrípeta.

**Fuerza de Coriolis.** También denominada efecto de Coriolis, descrita en 1835 por el científico francés Gaspard-Gustave Coriolis, es una fuerza ficticia o aparente que sirve para explicar el movimiento anómalo que describe un objeto que se mueve dentro de un sistema de referencia no inercial en rotación.

**Función Lipschitz.** La continuidad Lipschitz es una forma fuerte de continuidad uniforme en funciones, y se define como un límite a la rapidez con la que puede cambiar de valor dicha función.

Formalmente, sea dos espacios métricos  $(X, d_X)$  y  $(Y, d_Y)$ , una función  $f: X \rightarrow Y$  se dice una *función Lipschitz*, si existe una constante real  $K > 0$  tal que

$$d_Y(f(x_1), f(x_2)) \leq K d_X(x_1, x_2)$$

para todo  $x_1, x_2 \in X$ . En ese caso, a  $K$  se le conoce como *constante de Lipschitz* para  $f$ .

**Matriz de Transformación Homogénea.** Es una matriz de dimensiones  $(4 \times 4)$  que transforma un vector de posición, expresado en coordenadas homogéneas, de un sistema de coordenadas hasta otro.

**Perturbaciones Matched.** Sea el sistema dinámico siguiente

$$\dot{x} = f(x) + B(x)u + h(x, t)$$

donde  $x$  es el vector de estados,  $f(x)$  y  $B(x)$  definen el sistema,  $u$  es la señal de control y  $h(x, t)$  es el vector de perturbaciones del sistema. Se dice que  $h(x, t)$  cumple la condición *matching* si se cumple

$$h(x, t) \in \text{span}\{B(x)\}. \quad (7.2.1).$$

Esto significa que la señal de control  $u$  puede afectar todas las componentes de  $h(x, t)$  por medio de la matriz de control  $B(x)$ , es decir, las perturbaciones se encuentran dentro del sub-espacio de control. A las perturbaciones que cumplen con (7.2.1) se les llama *perturbaciones matched*.

**Sistema MIMO.** (del inglés *Multi Input-Multi Output*) se trata de sistemas que tienen múltiples entradas y salidas.

**Sistema No Lineal.** Sistemas cuyo comportamiento no es expresable como la suma de los comportamientos de sus descriptores. Más formalmente, un sistema físico, matemático o de otro tipo cuyas ecuaciones de movimiento, evolución o comportamiento no se pueden expresar con una ecuación lineal. En particular, el

comportamiento de sistemas no lineales no está sujeto al principio de superposición, como lo es el de un sistema lineal.

**Superficie Deslizante.** Es una trayectoria definida en función de los estados de un sistema, que representa una trayectoria en el espacio de estados, por la cual se desea que se "deslice" el comportamiento del sistema. Esto se logra por medio de un control discontinuo, generalmente ocasionado por la inclusión de una función signo.

**Superficie Pseudo-deslizante.** Es el análogo de la superficie deslizante cuando la función signo se aproxima por medio de una función sigmoidea.

**Tensor de Inercia.** Un tensor es una entidad geométrica que combina los conceptos de escalar, vector y operador lineal, de tal forma que sean independientes al sistema de coordenadas en el que estén expresados. Por lo tanto, el Tensor de Inercia es un conjunto de momentos y productos de inercia de un sistema mecánico, expresados en tensores y definido respecto a un sistema de referencia inercial.

**Teorema de Steiner.** Es una fórmula que nos permite calcular el momento de inercia de un sólido rígido respecto de un eje de rotación que pasa por un punto O, cuando conocemos el momento de inercia respecto a un eje paralelo al anterior y que pasa por el centro de masas.

## Bibliografía

1. International Federation of Robotics. *International Federation of Robotics*. [En línea] 21 de Agosto de 2011. [www.IFR.org](http://www.IFR.org).
2. **Canudas de Wit, C., Siciliano, B., y Bastin, G.** *Theory of Robot Control*. UK : Springer-Verlag, 1996.
3. *Output-Feedback PID Controllers for Robot Manipulators with Jacobian Uncertainty*. **Xiafu, H., y Peng, C.** Kaohsiung, Taiwan : s.n., 2008.
4. *Passivity analysis of a motion control for robot manipulators with dynamic friction*. **Canudas De Wit, C., y Kelly Martinez, R.** 1, 2007, Asian Journal of Control, Vol. 9.
5. **Behal, A., Dixon, W., Dawson, D., y Xian, B.** *Lyapunov-Based Control of Robotic Systems*. s.l. : CRC Press, Taylor and Francis, 2009. p. 389.
6. *Robust Tracking of Perturbed Nonlinear Systems by Nested Sliding Mode Control*. **Adhami-Mirhosseini, A., y Yazdanpanah, M.J.** 2005. International Conference on Control and Automation. pp. 44-48.
7. *Variable Structure Position Control of an Industrial Robotic Manipulator*. **Mendes, M. F., Kraus, W., y De Pieri, E. R.** 2002, Journal of the Brazilian Society of Mechanical Sciences, pp. 169-176.
8. *An Efficient Calculation of Computed Torque Control of Flexible Manipulators*. **Khalil, W., y Boyer, F.** 1995. Proceedings of the IEEE International Conference on Robotics and Automation. Vol. 1, pp. 609-614.
9. *Trajectory Control of Robotic Manipulator Using neural Networks*. **Ozaki, T., Susuki, T., Furuhashi, T., Okuma, S., y Uchikawa, Y.** 6, 1992, IEEE Transactions on Industrial Electronics, Vol. 39, pp. 555-570.
10. *A Fuzzy Sliding Mode Controller for Robust Tracking of Robotic Manipulators*. **Choi, S. B., y Kim, J. S.** 2, 1997, Mechatronics, Vol. 7, pp. 199-216.
11. *Adaptive Manipulator Control: A case study*. **Slotine, J. E., y Li, W.** 11, 1988, IEEE Transactions on Automatic Control, Vol. 33, pp. 995-1003.
12. **Craig, J. J.** *Adaptive Control of Mechanical Manipulators*. s.l. : Addison-Wesley, 1988.
13. **Stoten, D. P.** *Model Reference Adaptive Control of Manipulators*. UK : Research Studies Press LTD, 1990.
14. **Spong, M. W., Hutchinson, S., y Vidyasagar, M.** *Robot Modeling and Control*. s.l. : John Wiley & Sons, 2006.

15. *Exponential control of a rotational motion of a rigid body using quaternions*. **EL-Gohary, A., y Elazab, E. R.** s.l. : Elsevier Science Inc., 2003, Applied Mathematics and Computation, Vol. 137, pp. 195–207.
16. *Tracking Control for Robot Manipulators with Kinematic and Dynamic Uncertainty*. **Braganza, D., Dixon, W. E., Dawson, D. M., y Xian, B.** Sevilla, España : s.n., 2005. Proceedings of the 44th IEEE Conference on Decision and Control.
17. **Jazar, R. N.** *Theory of Applied Robotics: Kinematics, Dynamics, and Control*. [ed.] New York. 2. s.l. : Springer, 2010. p. 883.
18. *Modeling and Control of a Humanoid Arm using Conformal Geometric Algebra and Sliding Modes*. **Carbajal-Espinoza, O., González-Jiménez, L., y Bayro-Corrochano, E.** Bled, Eslovenia : s.n., 2011. Humanoids 2011 (11th IEEE-RAS International Conference on Humanoid Robots).
19. **Bayro-Corrochano, E., Falcón Morales, L. E., y Zamora Esquivel, J.** Visually Guided Robotics Using Conformal Geometric Computing. *Mobile Robots: Perception and Navigation*. 2007, pp. 19-44.
20. **Bayro-Corrochano, E.** Robot Perception and Action Using Conformal Geometric Algebra. *Applications in Pattern recognition, Computer Vision, Neural Computing and Robotics*. Heidelberg : Springer Verlag, 2005, pp. 405-458.
21. *Real-time tracking meets online grasp planning*. **Kragic, D., Miller, A. T., y Allen, P. K.** Seúl, Korea : s.n., 2001. International Conference on Robotics and Automation (ICRA). pp. 2460–2465.
22. *A Combined Corner and Edge Detector*. **Harris, C., y Stephens, M.** Manchester, UK : s.n., 1988. Alvey Vision Conference. pp. 147–151.
23. *Good Features to Track*. **Shi, J., y Tomasi, C.** Seattle, USA : s.n., 1994. International Conference on Computer Vision and Pattern Recognition. pp. 593–600.
24. *Object Recognition from Local Scale-Invariant Features*. **Lowe, D. G.** Corfu, Grecia : s.n., 1999. International Conference on Computer Vision. pp. 1150–1517.
25. *Color Indexing*. **Swain, M. J., y Ballard, D. H.** 1991, International Journal of Computer Vision, Vol. 7, pp. 11-32.
26. *Color Histogram Algorithms for Visual Robot Control*. **Buessler, J. L., Urban, J. P., Hermann, G., y Kihl, H.** 2005, International Journal of Robotics and Automation, pp. 86-93.
27. *Spatiograms Versus Histograms for Region-Based Tracking*. **Birchfield, S.T., y Rangarajan, S.** 2005, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1158-1163.

28. *Applications of Grassmann's Extensive Algebra*. Clifford, W. K. s.l. : The John Hopkins University Press, 1878, American Journal of Mathematics, Vol. 1, pp. 350-358.
29. Li, H., Hestenes, D., y Rockwood, A. Generalized Homogeneous Coordinates for Computational Geometry. [book auth.] Gerald Sommer. *Geometric Computing with Clifford Algebras*. 1. s.l. : Springer, 2000, pp. 27-58.
30. *Motor Algebra Approach for Computing the Kinematics of Robot Manipulators*. Bayro-Corrochano, E., y Kähler, D. 2000, Journal of Robotics Systems, pp. 495-516.
31. Bernal-Marín, Miguel. *Mapas Tridimensionales para Navegación Robótica*. Departamento de Ingeniería Eléctrica y Ciencias Computacionales, CINVESTAV del IPN, Unidad Gdl. Zapopan, Jalisco : s.n., 2007. Tesis de Maestría.
32. Zamora-Esquivel, Julio. *Manipulación Robótica de Objetos Guiada por Visión Usando Álgebra Geométrica*. Departamento de Ingeniería Eléctrica y Ciencias Computacionales, CINVESTAV del IPN, Unidad Gdl. Zapopan, Jalisco : s.n., 2006. Tesis Doctoral.
33. Utkin, V., Guldner, J., y Shi, J. *Sliding Mode Control in Electromechanical Systems*. UK : Taylor and Francis , 1999.
34. *Contact Force Regulation in Wire Actuated Pantographs Via Variable Structure Control and Frequency-Domain Techniques*. Pisano, A., y Usai, E. 2008, International Journal of Control, pp. 1747-1762.
35. *Blood Glucose Regulation Using Higher Order Sliding Mode Control*. Kaveh, P., y Shtessel, Y. 2008, International Journal of Robust and Nonlinear Control, pp. 557-569.
36. Boiko, I. *Discontinuous Control Systems*. Boston : Birkhauser, 2008.
37. *Frequency Characteristics of Levant's Differentiator and Adaptive Sliding Mode Differentiator*. Kobayashi, S., y Furuta, K. 10, 2007, International Journal of System Science, Vol. 38, pp. 825 – 832.
38. *Homogeneity Approach to High-Order Sliding Mode Design*. Levant, A. 2005, Automatica, pp. 823–830.
39. *Second-Order Sliding Modes Observer for Mechanical Systems*. Davila, J., Fridman, L., y Levant, A. 11, 2005, IEEE Transactions on Automatic Control, Vol. 50, pp. 1785–1789.
40. *Exact State Estimation for Linear Systems with Unknown Inputs Based on Hierarchical Super-Twisting Algorithm*. Bejarano, J. F., Fridman, L., y Poznyak, A. 18, 2007, International Journal on Robust and Nonlinear Control, Vol. 17, pp. 1734-1753.

41. *Super Twisting Algorithm-Based Step-by-Step Sliding Mode Observers for Nonlinear Systems with Unknown Inputs*. Floquet, T., y Barbot, J. P. 10, 2007, International Journal of system Science, Vol. 38, pp. 803 – 815.
42. Hartley, R., y Zisserman, A. *Multiple View Geometry in Computer Vision*. UK : Cambridge University Press, 2000.
43. Zhang, Z. *Flexible Camera Calibration by Viewing a Plane from Unknown Orientations*. s.l. : Microsoft Research, 1998. Reporte Técnico MSR-TR-98-71.
44. González R., y Wood. R. *Digital Image Processing*. Cambridge, MA : MIT Press, 1990.
45. *Real Time Versatile Robotics Hand/Eye Calibration using 3D Machine Vision*. Tsai, R.Y., y Lenz, R.K. 1988. IEEE International Conference on Robotics and Automation. Vol. 1, pp. 554-561.
46. *The Dual Quaternion Approach to Hand-Eye Calibration*. Daniilidis, K., y Bayro-Corrochano, E. Viena, Austria : s.n., 1996. 13th International Conference on Pattern Recognition (ICPR'96). pp. 318-322.
47. *An Improved Geometric Radial Basis Function Network for Hand-Eye Calibration*. Vazquez-Santacruz, E., y Bayro-Corrochano, E., San Jose, California, USA : s.n., 2011. International Joint Conference on Neural Networks.
48. *Parallel forward Dynamics: A geometric approach*. Zamora-Esquivel, J., y Bayro-Corrochano, E. Taipei : s.n., 2010. International Conference on Intelligent Robots and Systems (IROS). pp. 2377-2382.
49. Levant, A., y Fridman, L. Higher Order Sliding Modes. [book auth.] W., y Barbot, J. Perruqueti. *Sliding Mode Control in Enginnering*. s.l. : Marcel Dekker, 2002, pp. 53-101.
50. *Sliding Order and Sliding Accuracy in Sliding Mode Control*. Levant, A. 6, 1993, International Journal of Control, Vol. 58, pp. 1247– 1263.
51. *A Lyapunov Approach to Second-Order Sliding Mode Controllers and Observers*. Moreno, J., y Osorio, M. Cancún, México : s.n., 2008. Conference on Decision and Control (CDC). pp. 2856-2861.
52. *Discrete-Time Variable Structure Control Systems*. Gao, W., Wang, Y., y Homaifa, A. 2, 1995, IEEE Transactions on Industrial Electronics, Vol. 42, pp. 117–122.
53. *Robust Discrete-Time Chattering Free Sliding Mode Control*. Golo, G., y Milosavljevic, C. 1, 2000, Systems and Control Letters, Vol. 41, pp. 19-28.
54. *An  $O(T^2)$  Boundary Layer in Sliding Mode For Sampled Data Systems*. Su, W. C., Drakunov, S., y Ozguner, U. 3, 2000, IEEE Transactions on Automatic Control, Vol. 45, pp. 482–485.
55. Utkin, V. Sliding Mode Control in Discrete-Time and Difference Systems. *Variable Structure and Lyapunov Contro*. New York : Springer-Verlag, 1994, pp. 87–107.

56. **Poznyak, A.** *Advanced Mathematical Tools fo Automatic Control Engineers: Deterministic Systems.* Amsterdam : Elsevier Science, 2008. p. 213.



"2011, Año del Turismo en México"

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Control Robusto de Mecanismos Robóticos Guiados por Visión

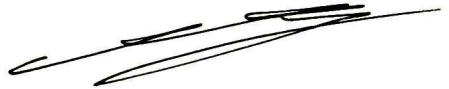
del (la) C.

Luis Enrique GONZÁLEZ JIMÉNEZ

el día 07 de Diciembre de 2011.



Dr. Eduardo José Bayro Corrochano  
Investigador CINVESTAV 3D  
CINVESTAV Unidad Guadalajara



Dr. Alexander Georgievich Loukianov  
Investigador CINVESTAV 3C  
CINVESTAV Unidad Guadalajara



Dr. José Javier Ruíz León  
Investigador CINVESTAV 3B  
CINVESTAV Unidad Guadalajara



Dr. Antonio Ramírez Treviño  
Investigador CINVESTAV 3A  
CINVESTAV Unidad Guadalajara



Dra. Nancy Arana Daniel  
Catedrático  
Universidad de Guadalajara



Dr. Carlos Alberto López Franco  
Profesor Investigador  
Universidad de Guadalajara



CINVESTAV - IPN  
Biblioteca Central



SSIT0010824