

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA MECÁNICA



**IMPLEMENTACIÓN DEL CONTROL VISUAL BASADO EN
IMAGEN PARA REGULAR LA POSICIÓN DE LA
CÁMARA, DE UN MÓDULO PTU – CAM TOF,
RESPECTO DE UN OBJETO DEFORMABLE**

TESIS

**PARA OPTAR EL TÍTULO PROFESIONAL DE:
INGENIERO MECATRÓNICO**

ELABORADO POR:

MARK WILLIAM SANCHEZ PALACIOS

PROMOCIÓN 2009 - I

LIMA - PERÚ

2015

Digitalizado por:

**Consortio Digital del
Conocimiento MebLatam,
Hemisferio y Dalse**

TABLA DE CONTENIDOS

PRÓLOGO	1
CAPÍTULO 1.	
Introducción	3
1.1. Antecedentes	5
1.2. Planteamiento del Problema	7
1.3. Justificación	8
1.4. Objetivos	9
1.5. Alcances	10
1.6. Limitaciones	11
CAPÍTULO 2.	
Descripción y Modelamiento del PTU–CAM	12
2.1. Descripción del Módulo	12
2.2. Cinemática	16
2.2.1. Movimiento de Cuerpo Rígido	16
2.2.2. Convención Denavit-Hartenberg	17
2.2.3. Cinemática de Velocidades	19
2.2.4. Cinemática Inversa de Velocidades	21
2.3. Dinámica	23
2.3.1. Dinámica del Mecanismo	23
2.3.2. Dinámica del Actuador	26
2.4. Control Desacoplado de Articulaciones	29
2.5. Modelo del Módulo	30
CAPÍTULO 3.	
Visión por Computador	32
3.1. Modelamiento de cámara	32
3.1.1. Parámetros Intrínsecos	35
3.1.2. Parámetros de Distorsión	35
3.2. Calibración de cámara	37
3.3. Cámara Time-of-Flight	39
3.4. Procesamiento de Imágenes	41
3.4.1. Segmentación de Color	42
3.4.2. Clasificación de Regiones	44

CAPÍTULO 4.	
Control Basado en Visión	48
4.1. Enfoques del Control Basado en Visión	49
4.2. Control Visual Basado en Imagen	52
4.2.1. Matriz de Interacción	53
4.2.2. Ley de Control	55
4.2.3. Compensación al Movimiento del Objeto	56
4.2.4. Transformación de Velocidades	57
4.3. Calibración Cámara-Robot	58
CAPÍTULO 5.	
RESULTADOS	66
5.1. Simulación	66
5.1.1. Respuesta del IBVS	70
5.1.2. Comparación entre el IBVS y el control proporcional	81
5.2. Implementación	93
CONCLUSIONES Y TRABAJOS FUTUROS	108
BIBLIOGRAFÍA	111
APÉNDICE A.	
Movimiento de Cuerpos Rígidos	116
A.1. Matriz de Rotación	116
A.2. Matriz Antisimétrica	118
A.3. Derivada de la Matriz de Rotación	119
A.4. Velocidad Lineal	120
A.4.1. Movimiento de un Punto Visto desde la Referencia	120
A.4.2. Punto Fijo Visto desde la Referencia en Movimiento	120
A.5. Velocidad Angular	121
A.6. Transformación de Velocidad	122

LISTA DE FIGURAS

1.1. Robot ADEPT Viper S850 utilizando IBVS.	4
1.2. PTU D48E en aplicaciones industriales.	4
1.3. Cámaras TOF comerciales de bajo costo.	5
1.4. Visual Servoing en la cirugía laparoscópica.	6
1.5. Arbusto de ajés Párikas.	8
2.1. Dibujo 3D del PTU-CAM.	13
2.2. Paquetes de instrucciones en el AX-12.	13
2.3. Cámara SoftKinect DepthSense DS325.	14
2.4. Esquema de conexión entre dispositivos.	15
2.5. Representación del movimiento de un cuerpo rígido.	16
2.6. Representación de los parámetros DH.	18
2.7. Asignación de los sistemas coordenados en el PTU-CAM.	19
2.8. Simulación de la Cinemática del PTU-CAM.	20
2.9. Modelo del motor DC.	27
2.10. Respuesta (12V) del modelo de motor DC del AX-12.	28
2.11. Control desacoplado de una articulación.	30
2.12. Modelo completo del PTU-CAM.	31
2.13. Dinámica inversa de la estructura del PTU-CAM.	31
3.1. Principio de una cámara Pinhole o Estenopeica.	33
3.2. Modelo de cámara pinhole.	34
3.3. Distorsión radial en la cámara.	36
3.4. Distorsión tangencial en la cámara.	36
3.5. Esquema de calibración interna de una cámara.	38
3.6. Tipos de modulación en una cámara TOF.	41
3.7. Espacio de color CIE $L^*a^*b^*$	43
3.8. Algoritmo de segmentación del color verde.	43
3.9. Morfología (dilatación y erosión).	44
3.10. Conectividad de píxeles y Etiquetado de regiones.	45
3.11. Algoritmo de Identificación de la región de interés.	46
3.12. Obtención de la posición de un objeto de color verde deformable en el Frame de la cámara.	46
3.13. Imagen filtrada con el algoritmo de identificación de la región de interés, para diferentes tonos de iluminación artificial.	47

4.1. Ubicación del sistema de visión.	51
4.2. Control ver y mover estático/dinámico basado en posición.	51
4.3. Control visual directo basado en imagen.	51
4.4. Control visual híbrido.	51
4.5. Sistema IBVS multifrecuencia.	52
4.6. Movimiento de una cámara respecto de un punto fijo.	53
4.7. Formulación $AX = ZB$	59
4.8. Formulación $AX = XB$	59
4.9. Plantilla de calibración ($AX = ZB$).	63
4.10. Disposición del entorno de calibración ($AX = ZB$).	64
4.11. Simulación de la Calibración $AX = ZB$ en el PTU-CAM.	65
5.1. Tiempo promedio de procesamiento y control.	68
5.2. Respuesta real y de simulación al escalón del modelo PTU-CAM. (a) y (b) Posición angular de las articulaciones. (c) y (d) Velocidad angular de las articulaciones.	68
5.3. Respuesta de simulación al escalón del modelo PTU-CAM. (a) Tensión eléctrica en los actuadores de las articulaciones. (b) Co- rriente eléctrica en los actuadores de las articulaciones. (c) Torques generados por los eslabones.	69
5.4. Diagrama de bloques del sistema con el IBVS.	70
5.5. Simulación del sistema utilizando el IBVS para un objeto fijo y diferentes ganancias. (a) Frame de la cámara. (b) Profundidad del objeto respecto a la cámara. (c) y (d) Errores en las características visuales (señales de control). (e) y (f) Velocidades angulares de las articulaciones.	71
5.6. Simulación del sistema utilizando el IBVS ($K = 10$) para un objeto fijo. (a) Vista del espacio de tareas con el objeto (rojo) y la estima- ción de las características visuales (azul), sobre el plano $X = 0.5$. (b) Frame de la cámara. (c) Vista del plano $X = 0.5$	72
5.7. Respuesta simulada del IBVS ($K = 10$) para un objeto fijo. (a) Error de las características visuales. (b) Profundidad del objeto respecto a la cámara. (c) y (d) Velocidad de la cámara respecto del objeto y expresado en la cámara. (e) y (f) Velocidad absoluta del último eslabón (gripper) respecto de la base del módulo.	74
5.8. Respuesta simulada del IBVS ($K = 10$) para un objeto fijo. (a) y (b) Velocidad angular deseada (azul) y actual (rojo) de las articu- laciones. (c) y (d) Posición angular deseada (azul) y actual (rojo) de las articulaciones.	75
5.9. Trayectoria del centro de masa de un objeto (ají) con movimiento pendular variable.	76
5.10. Simulación del sistema utilizando el IBVS ($K = 10$) para un objeto con movimiento pendular variable. (a) Vista del espacio de tareas con el objeto (rojo) y la estimación de las características visuales (azul), sobre el plano $X = 0.5$. (b) Frame de la cámara. (c) Vista del plano $X = 0.5$	77

5.11. Simulación del sistema utilizando el IBVS ($K = 10$) para un objeto con movimiento pendular variable. (a) Error de las características visuales. (b) Profundidad del objeto respecto a la cámara. (c) y (d) Velocidad de la cámara respecto del objeto y expresado en la cámara. (e) y (f) Velocidad absoluta del último eslabón (gripper) respecto de la base del módulo.	79
5.12. Simulación del sistema utilizando el IBVS ($K = 10$) para un objeto con movimiento pendular variable. (a) y (b) Velocidad angular deseada (azul) y actual (rojo) de las articulaciones. (c) y (d) Posición angular deseada (azul) y actual (rojo) de las articulaciones.	80
5.13. Diagrama de bloques del sistema con el control proporcional.	81
5.14. Simulación del sistema utilizando el control proporcional para un objeto fijo y diferentes ganancias, que proporcionan los mismos tiempos de establecimiento que sus similares del IBVS. (a) Frame de la cámara. (b) Profundidad del objeto respecto a la cámara. (c) y (d) Error de las características visuales (señales de control). (e) y (f) Velocidad angular de las articulaciones.	83
5.15. (a)–(f) Simulación de la comparación de los error de las características visuales (señales de control) entre el control proporcional y el IBVS para un objeto fijo y diferentes ganancias que proporcionan los mismos tiempos de establecimiento.	84
5.16. Simulación del sistema utilizando el IBVS ($K = 10$) y el control proporcional ($K = 0.0183$) para un objeto fijo. (a) Vista del espacio de tareas con el objeto (rojo) y la estimación de las trayectorias de las características visuales en el IBVS (azul) y el <i>control proporcional</i> (verde), sobre el plano $X = 0.5$. (b) Frame de la cámara. (c) Vista del plano $X = 0.5$	85
5.17. Respuesta simulada del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto fijo. (a) y (b) Error de las características visuales. (c) y (d) Velocidad angular deseada de las articulaciones. (e) y (f) Velocidad angular actual de las articulaciones.	87
5.18. Respuesta simulada del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto fijo. (a) y (b) Posición angular deseada de las articulaciones. (c) y (d) Posición angular actual de las articulaciones.	88
5.19. Respuesta simulada del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto con movimiento pendular variable. (a) Vista del espacio de tareas con el objeto (rojo) y la estimación de las trayectorias de las características visuales en el IBVS (azul) y el <i>control proporcional</i> (verde), sobre el plano $X = 0.5$. (b) Frame de la cámara. (c) Vista del plano $X = 0.5$	89

5.20. Respuesta simulada del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto con movimiento pendular variable. (a) y (b) Error de las características visuales. (c) y (d) Velocidad angular deseada de las articulaciones. (e) y (f) Velocidad angular actual de las articulaciones.	91
5.21. Respuesta simulada del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto con movimiento pendular variable. (a) y (b) Posición angular deseada de las articulaciones. (c) y (d) Posición angular actual de las articulaciones.	92
5.22. Diagrama de bloques del programa del IBVS.	93
5.23. Entornos gráficos del control IBVS.	94
5.24. Disposición del PTU-CAM y el objeto fijo.	94
5.25. Respuesta real y simulada del sistema utilizando el IBVS ($K = 10$) para un objeto fijo. (a) Vista del espacio de tareas con objeto (rojo) y la estimación de las características visuales, sobre el plano $X = 0.5$. (b) Frame de la cámara. (c) Vista del plano $X = 0.5$. (d) Profundidad del objeto respecto a la cámara.	95
5.26. Respuesta real y simulada del sistema utilizando el IBVS ($K = 10$) para un objeto fijo. (a) y (b) Error de las características visuales. (c)-(f) Velocidad de la cámara respecto del objeto y expresado en la cámara.	97
5.27. Respuesta real y simulada del sistema utilizando el IBVS ($K = 10$) para un objeto fijo. (a)-(d) Velocidad absoluta del último eslabón (gripper) respecto de la base del módulo. (e) y (f) Velocidad deseada de las articulaciones.	98
5.28. Respuesta real y simulada del IBVS ($K = 10$) para un objeto fijo. (a) y (b) Velocidad angular actual de las articulaciones. (c) y (d) Posición deseada de las articulaciones. (e) y (f) Posición actual de las articulaciones.	99
5.29. Respuesta real y simulada del control proporcional ($K = 0.0183$) para un objeto fijo. (a) Vista del espacio de tareas con objeto (rojo) y la estimación de las características visuales, sobre el plano $X = 0.5$. (b) Frame de la cámara. (c) Vista del plano $X = 0.5$. (d) Profundidad del objeto respecto a la cámara.	100
5.30. Respuesta real y simulada del control proporcional ($K = 0.0183$) para un objeto fijo. (a) y (b) Error de las características visuales. (c), (d) Velocidad deseada de las articulaciones. (e) y (f) Velocidad angular actual de las articulaciones.	102
5.31. Respuesta real y simulada del control proporcional ($K = 0.0183$) para un objeto fijo. (a) y (b) Posición deseada de las articulaciones. (c) y (d) Posición actual de las articulaciones.	103

- 5.32. Respuesta real del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto fijo. (a) Vista del espacio de tareas con objeto (rojo) y la estimación de las características visuales, sobre el plano $X = 0.5$. (b) Frame de la cámara. (c) Vista del plano $X = 0.5$. (d) Profundidad del objeto respecto a la cámara. 104
- 5.33. Respuesta real del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto fijo. (a) y (b) Error de las características visuales. (c) y (d) Velocidad deseada de las articulaciones. (e) y (f) Velocidad angular actual de las articulaciones. 106
- 5.34. Respuesta real del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto fijo. (a) y (b) Posición deseada de las articulaciones. (c) y (d) Posición actual de las articulaciones. . . . 107

LISTA DE TABLAS

2.1. Especificación del AX-12	14
2.2. Especificación de la cámara DS325	15
2.3. Definición de los parámetros DH	18
2.4. Parámetros DH del PTU-CAM	18
2.5. Definición de los parámetros físicos de un eslabón	24
2.6. Parámetros físicos del PTU-CAM	26
2.7. Definición de los parámetros del motor DC	28
2.8. Parámetros de motor DC del AX-12	29
3.1. Parámetros de cámara	37
3.2. Ventajas y desventajas de los tipos de estimación TOF	40
3.3. Áreas de las máscaras obtenidas del entorno filtrado para diferentes tonos de iluminación artificial.	47
4.1. Parámetros de cámara DS325 (marco de profundidad)	62
4.2. Parámetros de cámara-robot del PTU-CAM	64
5.1. Error RMS de la simuladas del IBVS y del control proporcional para un objeto fijo.	86
5.2. Error RMS de la simuladas del IBVS y del control proporcional para un objeto con movimiento pendular variable.	90
5.3. Error RMS de la implementación del IBVS y del control proporcional para un objeto fijo.	105
A.1. Propiedades de la matriz de rotación	116
A.2. Rotaciones elementales de la matriz de rotación	117
A.3. Representaciones de orientación a matrices de rotación equivalentes	117
A.4. Matriz de rotación a representaciones de orientación equivalentes	118
A.5. Propiedades de la matriz antisimétrica	119

PRÓLOGO

La tesis presenta el uso del *control visual basado en imagen* para regular la posición de la cámara, de un módulo “Pan-Tilt cámara TOF”, respecto de un objeto deformable de color verde. El módulo está constituida de dos servomotores Dynamixel AX-12 y una cámara TOF (Time of Flight o Tiempo de Vuelo) SoftKinetic DepthSense DS325.

La finalidad de controlar visualmente el módulo es crear un dispositivo dinámico para obtener información visual relevante del entorno de trabajo. Las aplicaciones de este nuevo dispositivo dinámico y versátil son muchas, entre ellos se distingue la videovigilancia automática y el control e interacción con mecanismo robóticos. Siendo más específico, se puede utilizar en el reconocimiento de frutos para una cosecha automatizada.

Para observar el rendimiento del *control visual basado en imagen* se compara con un *control proporcional*, que es simple y sencillo de implementar, en la regulación de la posición de la cámara del módulo respecto de un objeto *fijo* y otro de *movimiento pendular variable*. Este último se realiza sólo en simulación y con la intención de recrear a un objeto dinámico en un rango limitado; por ejemplo, el movimiento de un fruto en su arbusto producido por el viento.

La tesis consta de cinco Capítulos, Conclusiones, Bibliografía y Apéndice. El Capítulo 1 se describe la *introducción* de la tesis, donde se menciona los antecedentes, planteamiento del problema, justificación, objetivos, alcance y limitaciones. El Capítulo 2 se describe el módulo y se encuentra el modelo matemático del

módulo para simular el controlador, utilizando los concepto de robótica como la *cinemática*, mediante la convención Denavit-Hartenberg; *dinámica*, mediante la formulación de Euler-Lagrange, para el mecanismo y los actuadores; y el *control desacoplado de articulaciones* para los actuadores. El Capítulo 3 se encuentra el modelo matemático de la cámara y un procedimiento para encontrar la posición del objeto deformable, en características visuales, mediante algoritmos básicos de procesamiento de imagen. El Capítulo 4 se describe los conceptos del *control visual basado en imagen* que permite al módulo regular la posición de la cámara respecto del objetivo deformable; y la calibración *cámara-robot* que permite acoplar la cámara al mecanismo Unidad Pan-Tilt (PTU). Por último, el Capítulo 5 presenta los resultados de la simulación y la implementación del *control visual basado en imagen* para un objeto deformable de color verde y su comparación con el *control proporcional*.

CAPÍTULO 1

Introducción

A menudo los robots necesitan de diversos sensores para obtener información de su entorno y poder interactuar con él. La *cámara* digital es un sensor muy versátil, mediante el cual se puede obtener mucha información del entorno de trabajo en comparación a los sensores convencionales. El uso de información visual procedente de una o varias cámaras para controlar el movimiento del efector final¹ de un robot respecto de la tarea a realizar, se le conoce como *control basado en visión* o *visual servoing* (ver Figura 1.1). Un caso particular de este control es el denominado *control visual basado en imagen* o IBVS (*image-based visual servoing*), el cual utiliza las características visuales de un objeto (p. ej., puntos, segmentos, círculos, esquinas, etc.) obtenidos por un sistema de *visión por computador* o *vision artificial*, como señal de referencia, para regular la posición y orientación del efector final del robot respecto del objeto fijo [1]. El IBVS, pese a ser diseñado para tareas de regulación respecto de un objeto fijo, puede utilizarse para objetos con movimiento dinámico desconocido (dentro de su visión de trabajo) haciéndolo muy práctico y robusto para diversa tareas de reconocimiento.

El PTU cámara, conocido como *cámara activa* y de aquí en adelante como PTU-CAM, es una unidad de movimiento panorámico y de inclinación (2 GDL²),

¹El *efector final*, es la herramienta o manipulador que está acoplado al extremo del brazo robótico.

²El *Grado De Libertad* (GDL) es el número de parámetros de un sistema que pueden variar de forma independiente.

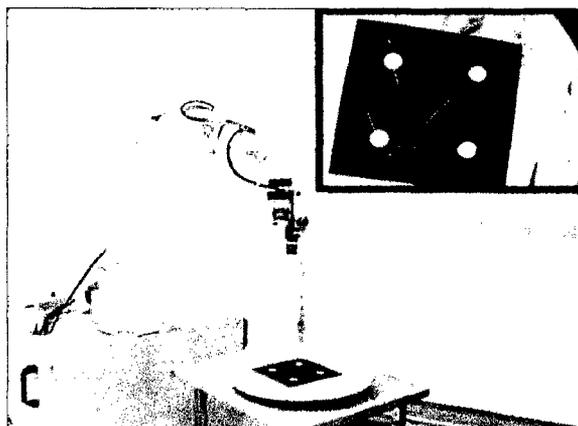


Figura 1.1: Robot ADEPT Viper S850 utilizando IBVS.

en el cual tiene adjunta en su extremo una o varias cámaras. Éste módulo es muy utilizado para la ubicación y seguimiento de objetos dinámicos. La Figura 1.2 muestra un PTU industrial.

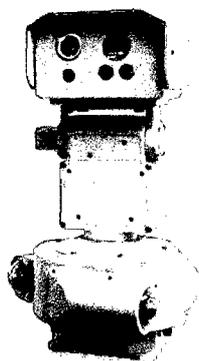


Figura 1.2: PTU D48E en aplicaciones industriales.

La cámara TOF (Tiempo de vuelo) es un dispositivo electrónico que adquiere información visual y de profundidad del entorno de trabajo. Ésta tiene, por lo general, una cámara de color y una de profundidad, y utiliza la calibración estereoscópica para sincronizar ambos marcos. La tecnología TOF tiene como principio la medición del tiempo que tarda la luz en viajar desde un emisor, ubicado cerca a la cámara de profundidad, hasta el objeto y luego volver al sensor de profundidad, con éste se calcula la distancia intermedia. Existen pocas cámaras TOF comerciales de bajo costo, entre las más conocidas están la *Microsoft*

Kinect, *Creative SENZ3D* y la *DepthSense DS325* (ver Figura 1.3).

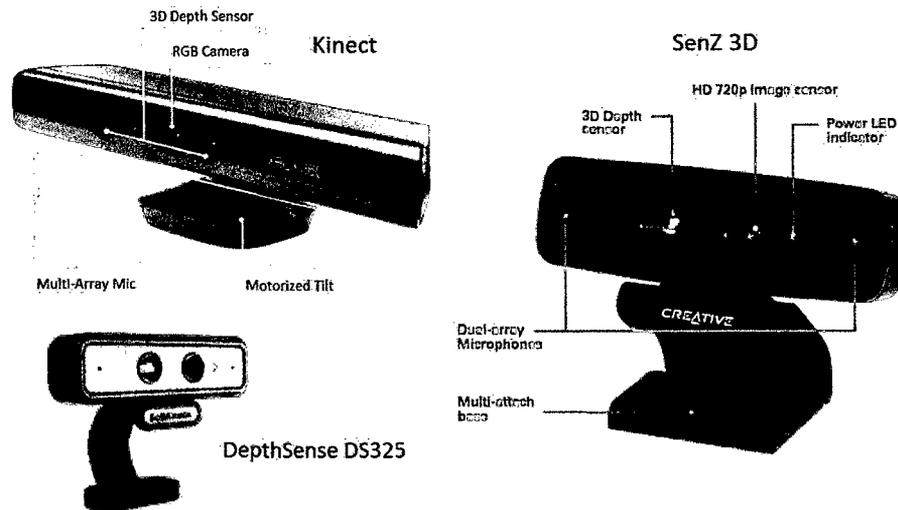


Figura 1.3: Cámaras TOF comerciales de bajo costo.

En las siguientes secciones se procede a revisar los *antecedentes* que relacione el PTU, cámara digital y el IBVS.

1.1. Antecedentes

La principal aplicación del *control basado en visión* están en las industrias que utilizan brazos robóticos, cuyos movimientos son controlados para evitar obstáculos, montar y desmontar piezas mecánicas, realizar trabajos de pintura o de soldaduras, entre otros. También, se utiliza en aplicaciones médicas [2], por ejemplo, la *cirugía laparoscópica* necesita realizar pequeñas incisiones en la pared abdominal para introducir pequeños instrumentos y la cámara laparoscópica, de manera que el cirujano pueda operar con solo mirar las imágenes obtenidas por la cámara (ver Figura 1.5). Otras aplicaciones están en robots móviles como el *vehículo submarino no tripulado* o UUV (*Unmanned Underwater Vehicle*) [3], el *vehículo terrestre no tripulado* o UGV (*Unmanned Ground Vehicle*) [4], y el *vehículo aéreo no tripulado* o UAV (*Unmanned Aerial Vehicle*) [5].

Una de las características del IBVS es el termino relacionado a la función

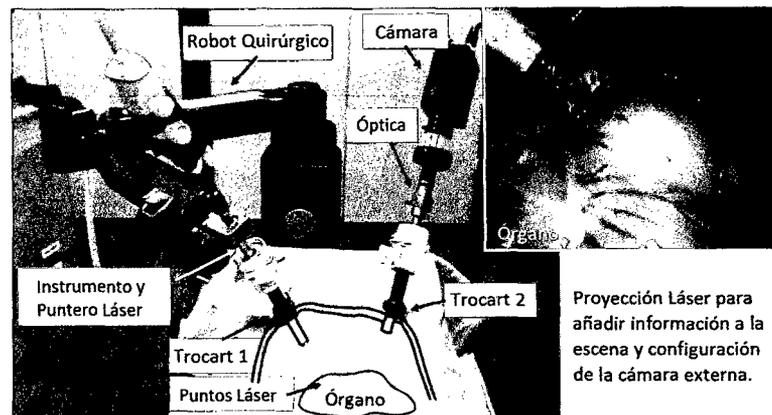


Figura 1.4: Visual Servoing en la cirugía laparoscópica.

tarea que realiza el robot con el objeto, denominado *matriz de interacción*. Esta matriz depende, entre otros parámetros asequibles, de la profundidad del objeto respecto de la cámara, el cual es complicado estimarlo en sistemas monoculares. En [6] se especifica el problema de la tarea en términos de una regulación de la posición de la cámara respecto de un objeto con movimiento estimable (profundidad) y predecible (filtro de kalman) respecto al sistema coordinado global; y también propone un estimador para realizar la labor de seguimiento. Un estudio más complejo puede verse en [7], el cual utiliza FLC³ para tratar la señal de control hacia las articulaciones del robot. Siguiendo con los sistemas monoculares, tenemos a [8] que realiza una estimación de la profundidad con un objeto fijo. Otro estudio más completo se ve en [9], donde se utiliza dos cámaras activas para obtener información de la profundidad del objetivo. También, en [10] se realiza una estimación en los parámetros de la función tarea como la *matriz de interacción* y la velocidad del objeto. Por otro lado, en [11] se realiza un estudio de diferentes estimadores del movimiento del objeto para ser adicionado al IBVS como compensador.

El uso del PTU puede observarse en [12], donde se utiliza como un asistente de manejo en automóviles. En [13] se utiliza para fines filmográficos, ayudando

³Controlador lógico difuso o FLC (*Fuzzy Logic Controller*).

al operario a reducir la maniobrabilidad de un sistema de enfoque con varios *grados de libertad*. En [14] y [15] se observa la ayuda del PTU a robots móviles para ubicar mejor los objetivos dinámicos. También, en [16] muestra el control cinemático simplificado del PTU, obteniendo un movimiento poco adecuado para la regulación. Por último, en [17] muestra el efecto de seguir un objeto con una cámara fija y una activa.

1.2. Planteamiento del Problema

El PTU-CAM es un mecanismo serial de 2 GDL para regular la posición de la cámara respecto de un objeto o realizar seguimiento a éste. Modelos más sencillo suelen usar un sistema monocular y un *controlador proporcional*, de forma que relacione los errores de las coordenadas (horizontal y vertical), en el marco de la imagen, con las velocidades angulares de las articulaciones. Sin embargo, éste puede sufrir problemas de sobre impulso en el seguimiento o regulación por objetos con movimiento desconocido, puesto que trabaja de forma directa en el marco de la imagen y no observa la cinemática del módulo.

El IBVS depende principalmente de la profundidad del objeto respecto de la cámara y éste puede ser calculado por un sistema estereoscopio (dos o más cámaras). Sin embargo, para cámaras de bajo costo, el cálculo del mapa de profundidad, a partir de la disparidad, suele ser muchas veces complejo y tedioso o no encontrarse concordancia entre píxeles de los marcos de las cámara, debido a cambios dinámicos del entorno como luminosidad, sombras y sincronización entre las cámaras; produciendo un mapa incompleto.

Por tanto, con la implementación del IBVS en el PTU-CAM se reduce el sobreimpulso en la tarea de regulación de la posición de la cámara respecto del objeto, realizando un movimiento suavizado y ahorrativo en mantenimiento. También, con el uso de la cámara *DepthSense DS325* proporciona un mapa de profundidad

completo y sincronizado con la cámara de color, dentro de su rango de trabajo.

1.3. Justificación

Como es de conocimiento público, el ají Páprika (ver Figura 1.5) es nuestro producto Capsicum de bandera (Perú), cuya producción se incrementa cada año a nivel nacional e internacional. Lugares como Lambayeque, Arequipa o Ica, tienen un buen clima para el crecimiento natural y óptimo del fruto. La cosecha del mismo se realiza mediante maquinaria pesada que destroza los arbustos en una sola pasada, teniendo que aplicar previamente aceleradores de crecimiento para homogeneizar el crecimiento, perdiendo las diversas floraciones naturales de la planta y elevando el precio del producto. Una forma de aprovechar estas floraciones naturales y tener un precio muy competitivo es cosechando manualmente, pero muchas veces no hay suficiente personal para cosechar grandes hectáreas, debido a las condiciones hostiles donde se desarrolla el producto, como elevadas temperaturas, problemas ergonómicos, problemas respiratorios por el polvo del lugar, entre otros.



Figura 1.5: Arbusto de ajíes Páprikas.

Para automatizar el proceso de cosecha de ajíes Páprikas, como medida a la falta de mano de obra en la cosecha manual, se debe utilizar principalmente

un dispositivo que obtenga información del ambiente de cosecha y actuadores recolectores de frutos (p. ej., brazos robóticos). El primer dispositivo mencionado es donde se enfoca esta tesis (PTU-CAM), principalmente en el tema de *control visual basado en imagen* y dejando en menor grado el procesamiento de imagen, ya que éste dependerá de la estrategia de cosecha y la forma del fruto y de su arbusto.

El fruto (color rojo vino) tiene un movimiento variable respecto a la rama que lo contiene, producido por vientos de la zona y vibraciones por maquinaria pesada. Esto provoca un cambio en su geometría al momento de la adquisición de las características visuales. Por tanto, se elige un objeto deformable esférico de plástico y de color verde como simulación real de un fruto. El color verde se elige para evitar cualquier ruido al momento de filtrar junto al color de la piel humana, puesto que se pretende interactuar con la misma.

1.4. Objetivos

Implementar la técnica de *control visual basado en imagen* en un módulo PTU-CAM TOF para obtener un movimiento suavizado en la tarea de regulación de la posición de la cámara respecto de un objeto deformable de color verde, en comparación al *control proporcional*.

Objetivos Específicos

- Obtener un modelo matemático del módulo PTU-CAM para simular el IBVS y el *control proporcional*.
- Obtener un procedimiento para obtener las características visuales de un objeto deformable de color verde.
- Obtener un método de calibración *cámara-robot* para acoplar la cámara DS325 y el mecanismo del PTU.

1.5. Alcances

La obtención y simulación del modelo mecánico del PTU–CAM se realiza con base a su estructura y su actuador. Para no extender demasiado y no desviarse del objetivo de la tesis, se toma los parámetros del motor DC para el AX–12 de la disertación [18], ya que en ésta se realiza un estudio más completo. Además, se realiza el modelo de cámara sin distorsión para no hacer más complejo la simulación.

Los parámetros visuales constantes de la *matriz de interacción*, que utiliza el IBVS, se obtiene de la cámara DS325 calibrada de fabrica, los cuales están optimizados para diferentes configuraciones internas y mejor funcionamiento. También, para encontrar la relación mecánica entre el PTU y la cámara DS325, se realiza una calibración adicional ligada a los antes mencionados conocido como calibración *cámara–robot*.

Para observar el rendimiento del IBVS se realiza la comparación con un *control proporcional* respecto a un objeto *fijo* y otro de *movimiento pendular variable*. La simulación del conjunto se realiza en *Simulink* de *Matlab*.

El Software, para la implementación real, utiliza la librería *Qt C/C++*, SDK⁴ DS325, USB2Dynamixel, y para el procesamiento de imagen la librería *OpenCV*. También, en éste se visualiza las gráficas de las posiciones angulares actuales y el error en características visuales. Además, la imagen 3D sincronizada y procesada de la cámara DS325 y el solido del modelo dibujado en SolidWorks, utilizan OpenGL⁵ y GLSL⁶ reescritas para Qt.

⁴Software Developer Kit o Herramientas de Desarrollo de Software

⁵Open Graphics Library, es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D.

⁶Es el acrónimo de OpenGL Shading Language (Lenguaje de Sombreado de OpenGL), también conocido como GLslang, una tecnología parte del API estandar OpenGL, que permite especificar segmentos de programas gráficos que serán ejecutados sobre el GPU.

1.6. Limitaciones

Esta tesis se limita a realizar algoritmos básicos de *visión por computador* para obtener características visuales de un objeto deformable de color verde. Por tanto, no se discute algoritmos más complejos para el reconocimiento. Trabajos futuros pueden mejorar éste o inclusive realizar aplicaciones con rostros de personas o vehículos en pleno tránsito.

Debido al rango limitado de sensado de profundidad por parte de la cámara DS325, se excluye a objetos que estén fuera de su rango.

CAPÍTULO 2

Descripción y Modelamiento del PTU–CAM

En este Capítulo se realiza una breve *descripción* de los principales elementos que conforman el PTU–CAM. Luego se procede a encontrar el modelo matemático a través de su *cinemática, dinámica y control*, considerando su estructura como un pequeño brazo robótico de 2 GDL, cuyos eslabones están conectados serialmente por articulaciones de rotación.

2.1. Descripción del Módulo

El PTU–CAM se conforma principalmente por tres elementos: el servomotor AX–12, la cámara 3D DS325 y el dispositivo de comunicación USB2Dynamixel. Entre ellos, el servomotor y la cámara son los que conforman la estructura mecánica (ver Figura 2.1).

El servomotor AX–12 está compuesto por un motor DC con caja reductora, un módulo de controlador interno (control de posición, velocidad, corriente, temperatura, entre otros) con amplificador de potencia, y un módulo de comunicación Half Duplex¹ UART², para los paquetes de instrucciones (ver Figura 2.2). Además, cuenta con accesorios mecánicos que permiten conectarse entre si. La Tabla 2.1 muestra las principales especificaciones de Hardware del AX–12.

La cámara 3D DS325 (ver Figura 2.3) fue creado por SoftKinect, cuyo sensor

¹Método o protocolo de envío de información bidireccional pero no simultáneo.

²Universal Asynchronous Receiver-Transmitter.

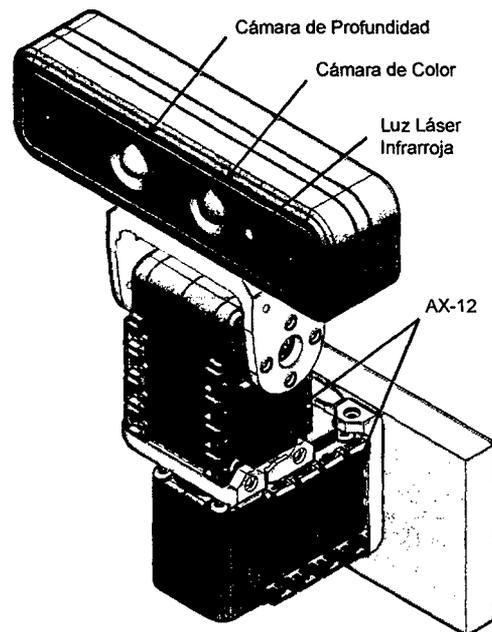


Figura 2.1: Dibujo 3D del PTU-CAM.

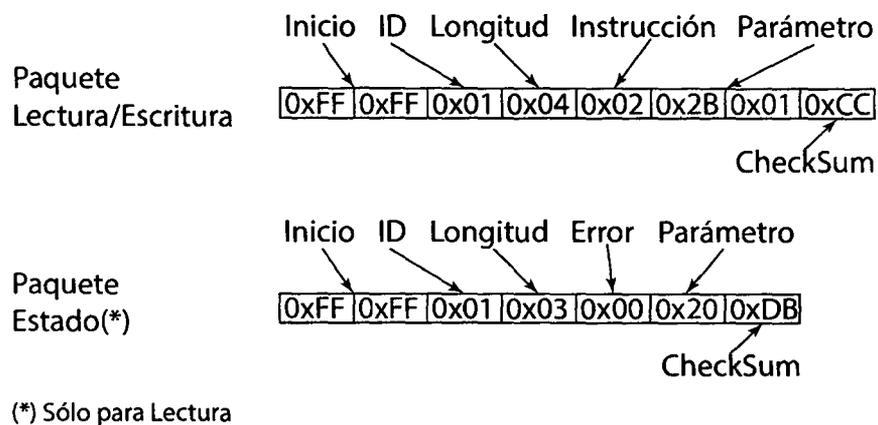


Figura 2.2: Paquetes de instrucciones en el AX-12.

de profundidad DepthSense (patentado) es de tipo TOF (Time-of-Flight). Este sensor mide el tiempo que tarda la luz infrarroja, emitida por un LED³, en rebotar en el objeto y llegar hasta el sensor de profundidad (tiempo de vuelo). Además, el DepthSense transforma los datos de posición TOF en imágenes 3D de color, así como la confianza en escala de grises y el mapa de profundidad. También, la

³Sigla de la expresión inglesa Light-Emitting Diode, "diodo emisor de luz".

Tabla 2.1: Especificación del AX-12

Especificación	Característica
Masa	53.5 g
Dimensión	32 × 50 × 40 mm
Resolución	0.29°
Relación de engranaje	254:1
Torque de parada	1.5 Nm (a 12 V, 1.5 A)
Velocidad sin carga	59 RPM (a 12 V)
Ángulo de trabajo	0° ~ 300°
Temperatura de trabajo	-5°C to 70°C
Voltaje	9 ~ 12 V (recomendado 11.1 V)
Señal de comando	Paquete digital
Tipo de protocolo	Serial Asíncrono Half-duplex (8 bit, 1 stop, No Parity)
Enlace físico	TTL Level Multi Drop (Daisy)
Identificación de dispositivo	254 ID (0 ~ 253)
Velocidad de comunicación	7343 bps ~ 1 Mbps
Realimentación	Posición, Temperatura, etc.
Material	Ingeniería de plástico

cámara tiene un conector USB que simplifica la comunicación con el computador.

La Tabla 2.2 muestra las principales especificaciones de Hardware del DS325.

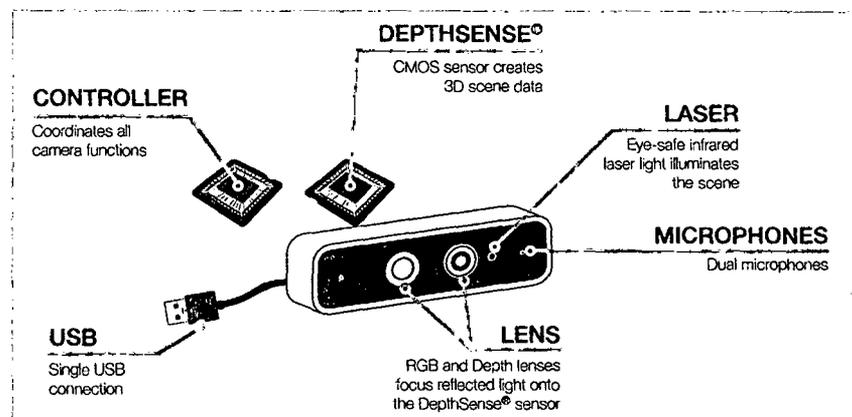


Figura 2.3: Cámara SoftKinect DepthSense DS325.

El dispositivo de comunicación externo USB2Dynamixel permite conectar los servomotores AX-12 al computador. Este dispositivo convierte el protocolo de comunicación UART a USB y viceversa, permitiendo una conexión con el

Tabla 2.2: Especificación de la cámara DS325

Especificación	Característica
Tecnología	CAPD Time-of-Flight
Número de Píxeles de Profundidad	320 × 240 (QVGA)
Campo de Visión de Profundidad	74° × 58° × 87° (H×V×D)
Velocidad de Cuadro	25 fps – 30 fps QVGA y QQVGA 50 fps – 60 fps QVGA y QQVGA
Rango Nominal de Operación	0.15 m – 1.0 m
Resolución de Profundidad	< 1.4 cm @1 m
Tipo de Iluminación	Láser Difusa
Luz Ambiental	Interior Típico
Número de Píxeles de RGB	HD 720p
Campo de Visión de Profundidad	63.2° × 49.3° × 75.2° (H×V×D)
Micrófonos	2
Acelerómetro	3 ejes
Conexión	simple UBS
Temperatura de Operación	10°C to 40°C
Poder	<= 2.5 W
Tamaño	10.5 × 3.0 × 2.3 cm

computador y ser manejado por éste mediante su SDK, el cual está escrito para diferentes entornos de programación (p. ej., Visual C++, Matlab, LabView, etc.). En [19] existe abundante información sobre el funcionamiento del AX-12 y el USB2Dynamixel. La Figura 2.4 muestra el esquema de conexión entre los dispositivos del PTU-CAM.

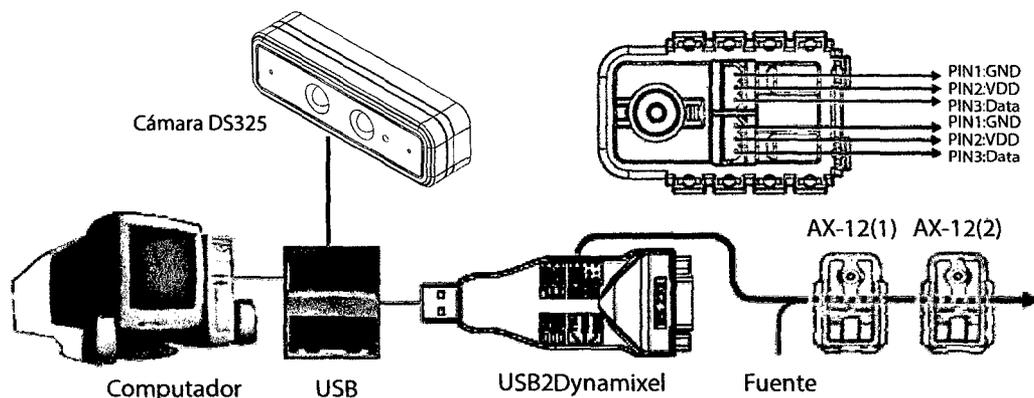


Figura 2.4: Esquema de conexión entre dispositivos.

2.2. Cinemática

La cinemática estudia el movimiento sin considerar las fuerzas que lo produce [20], es decir, solo el cambio de las *propiedades geométricas*⁴ del robot en función del tiempo [21].

2.2.1. Movimiento de Cuerpo Rígido

La base de la formulación de la *cinemática* es la representación del movimiento de un cuerpo rígido. Éste se define por su ubicación, representado por un *vector de posición* $p := p(t) \in \mathbb{R}^3$, y su orientación, representado por una *matriz de rotación* $R := R(t) \in SO(3)$ ⁵, en función del tiempo (ver Figura 2.5); y se formula por

$${}^j r = {}^j R_i {}^i r + {}^j p_i, \quad (2.1)$$

donde ${}^i r$ representa el vector posición de un punto perteneciente al cuerpo rígido respecto al sistema coordenado $\{S_i\}$ adjunto a éste, ${}^j p_i$ y ${}^j R_i$ representa el vector posición y la matriz de rotación respectivamente del origen del sistema coordenado $\{S_i\}$ respecto de otro $\{S_j\}$, y ${}^j r$ representa el vector posición del punto respecto de $\{S_j\}$.

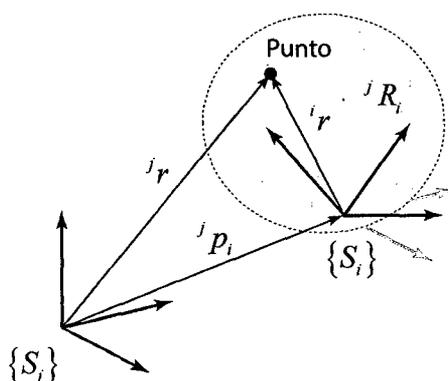


Figura 2.5: Representación del movimiento de un cuerpo rígido.

⁴En robótica, se conoce como *modelo geométrico* a la relación entre la posición y orientación del extremo del robot y las variables asociadas a sus articulaciones en un instante de tiempo.

⁵Grupo ortogonal especial del conjunto de todas las orientaciones en 3 dimensiones.

El Apéndice A.1 muestra otras representaciones de la orientación de un cuerpo rígido.

La ecuación (2.1) puede representarse matricialmente como

$$\begin{pmatrix} {}^j r \\ 1 \end{pmatrix} = \begin{pmatrix} {}^j R_i & {}^j p_i \\ 0_{1 \times 3} & 1 \end{pmatrix} \begin{pmatrix} {}^i r \\ 1 \end{pmatrix}, \quad (2.2)$$

donde a la matriz de orden 4×4 se le denomina *matriz de transformación homogénea* (MTH) y se representa por

$$T = \begin{pmatrix} R & p \\ 0_{1 \times 3} & 1 \end{pmatrix}. \quad (2.3)$$

2.2.2. Convención Denavit-Hartenberg

Una forma práctica de solucionar el *problema cinemático directo* de un robot de n -eslabones, el cual determinar la ubicación y orientación del extremo del robot conocidas las variables asociadas a las articulaciones en función del tiempo, es mediante la *conversión de Denavit-Hartenberg* (convención DH). Éste define cuatro magnitudes (θ, d, a, α) conocidos comúnmente como parámetros DH (ver Tabla 2.3), que permiten encontrar la MTH de cada cada eslabón respecto de su precedente (ver Figura 2.6) y cuya obtención está dado por

$${}^{i-1}T_i = T_{Rot}(\hat{z}_{i-1}, \theta_i) T_{Trans}(\hat{z}_{i-1}, d_i) T_{Trans}(\hat{x}_i, a_i) T_{Rot}(\hat{x}_i, \alpha_i), \quad (2.4)$$

donde $T_{Rot}(\hat{z}_{i-1}, \theta_i)$ y $T_{Rot}(\hat{x}_i, \alpha_i)$ son las MTH de rotación pura respecto de sus respectivos ángulos (ver Apéndice A.2), y $T_{Trans}(\hat{z}_{i-1}, d_i)$ y $T_{Trans}(\hat{x}_i, a_i)$ son las MTH de traslación pura, a través de sus respectivos ejes. El parámetro θ es la variable asociada a una articulación, en donde gira un eslabón respecto de otro, y los demás son parámetros son fijos y característicos del eslabón mencionado [21].

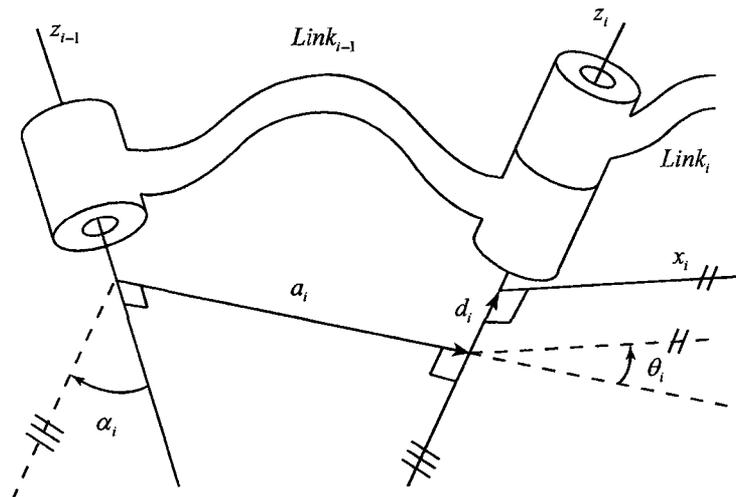


Figura 2.6: Representación de los parámetros DH.

Tabla 2.3: Definición de los parámetros DH

Parámetro	Definición
θ_i	Rotación respecto de z_{i-1}
d_i	Desplazamiento en z_{i-1}
a_i	Desplazamiento en x_i
α_i	Rotación respecto de x_i

Finalmente, la convención DH permite encontrar la MTH del sistema coordenado adjunto al extremo del robot $\{S_n\}$ respecto del sistema coordenado de referencia ubicado en la base del mismo $\{S_0\}$.

PTU-CAM

La Figura 2.7 muestra la asignación de los sistemas coordenados a cada eslabón y en la Tabla 2.4 sus respectivos parámetros DH.

Tabla 2.4: Parámetros DH del PTU-CAM

Articulación	θ_i [rad]	d_i [m]	a_i [m]	α_i [rad]
1	q_1	41.5×10^{-3}	0	$-\frac{\pi}{2}$
2	$q_2 - \frac{\pi}{2}$	0	26.5×10^{-3}	$-\frac{\pi}{2}$

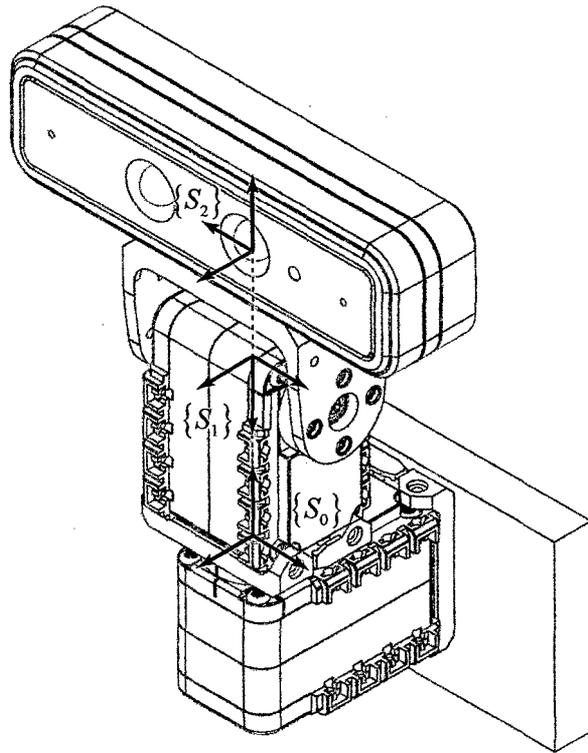


Figura 2.7: Asignación de los sistemas coordinados en el PTU-CAM.

La Figura 2.8 muestra la simulación de la cinemática directa, cuyo MTH característico es

$${}^0T_2 = \begin{pmatrix} \cos \theta_1 \sin \theta_2 & \sin \theta_1 & \cos \theta_1 \cos \theta_2 & 0.0265 \cos \theta_1 \sin \theta_2 \\ \sin \theta_1 \sin \theta_2 & -\cos \theta_1 & \sin \theta_1 \cos \theta_2 & 0.0265 \sin \theta_1 \sin \theta_2 \\ \cos \theta_2 & 0 & -\sin \theta_2 & 0.0265 \cos \theta_2 + 0.0415 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.5)$$

2.2.3. Cinemática de Velocidades

La *cinemática de velocidades* permite encontrar la velocidad total (lineal y angular) del extremo del robot de n -eslabones, a partir de las velocidades de las variables asociadas a las articulaciones [22]. Ésta relación se conoce como

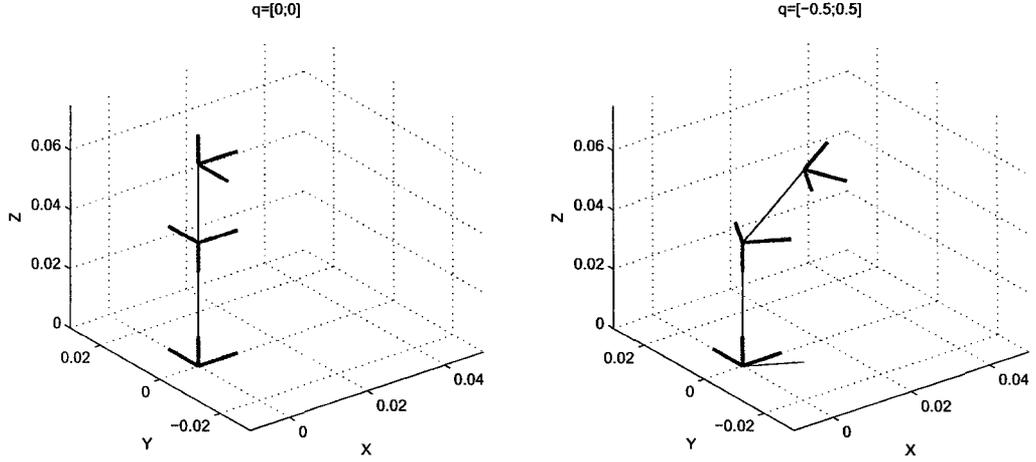


Figura 2.8: Simulación de la Cinemática del PTU-CAM.

Jacobiano del robot o simplemente *Jacobiano* [20], y se formula por

$$V_n = J\dot{q}, \quad (2.6)$$

donde $V_n := {}^0V_n = (v_n, \omega_n)^T$ representa la velocidad total (velocidad lineal $v_n := {}^0v_n$ y velocidad angular $\omega_n := {}^0\omega_n$) del extremo del robot respecto de su base, $q = (q_1, q_2, \dots, q_n)^T$ representa las variables asociadas a las articulaciones, y $J := J(q, t) = (J_v, J_\omega)^T$ es el Jacobiano (orden $6 \times n$) compuesto del *Jacobiano lineal* (J_v) y el *Jacobiano angular* (J_ω).

Considerando la *cinemática directa* obtenida por la convención DH, el *Jacobiano lineal* se obtiene por

$$J_v := J_v \Big|_{p_n} = \begin{pmatrix} \frac{\partial p_x}{\partial q_1} & \frac{\partial p_x}{\partial q_2} & \cdots & \frac{\partial p_x}{\partial q_n} \\ \frac{\partial p_y}{\partial q_1} & \frac{\partial p_y}{\partial q_2} & \cdots & \frac{\partial p_y}{\partial q_n} \\ \frac{\partial p_z}{\partial q_1} & \frac{\partial p_z}{\partial q_2} & \cdots & \frac{\partial p_z}{\partial q_n} \end{pmatrix}, \quad (2.7)$$

donde $p_n := {}^0p_n = (p_x, p_y, p_z)^T$ es el vector posición del sistema coordinado adjunto al último eslabón o extremo del robot $\{S_n\}$ respecto de su base $\{S_0\}$; y

el *Jacobiano angular* se obtiene por

$$J_\omega = \begin{pmatrix} z_{0x} & z_{1x} & \dots & z_{(n-1)x} \\ z_{0y} & z_{1y} & \dots & z_{(n-1)y} \\ z_{0z} & z_{1z} & \dots & z_{(n-1)z} \end{pmatrix}, \quad (2.8)$$

donde $z_i := {}^0z_i = (z_{ix}, z_{iy}, z_{iz})^T$ representa al eje z o la tercera columna de la matriz de rotación (0R_i) del eslabón i respecto de la base, y puede calcularse como ${}^0z_i = {}^0R_i z_0$, donde $z_0 = (0, 0, 1)^T$.

PTU-CAM

Realizando los cálculos correspondientes, el Jacobiano del módulo es

$$J = \begin{pmatrix} J_v \\ J_\omega \end{pmatrix} = \begin{pmatrix} -0.0265 \sin \theta_1 \sin \theta_2 & 0.0265 \cos \theta_1 \cos \theta_2 \\ 0.0265 \cos \theta_1 \sin \theta_2 & 0.0265 \sin \theta_1 \cos \theta_2 \\ 0 & -0.0265 \sin \theta_2 \\ 0 & -\sin \theta_1 \\ 0 & \cos \theta_1 \\ 1 & 0 \end{pmatrix}. \quad (2.9)$$

2.2.4. Cinemática Inversa de Velocidades

La *cinemática inversa de velocidades* permite encontrar las velocidades de las variables asociadas a las articulaciones a partir de la velocidad total (lineal y angular) del extremo del robot de n -eslabones [22]. Esta relación se resuelve con la inversión del Jacobiano de orden cuadrado (p. ej., robot de 6 GDL), pero al realizar este proceso puede presentar serios inconvenientes como singularidades⁶ en vecindades, aunque las técnicas basadas en *mínimos cuadrados*⁷ pueden mejorar la inversión condicionada desde el punto de vista numérico. Por lo general, el

⁶En robótica, se conoce como *singularidad* o *configuración singular* de un robot en donde la determinante de la matriz *jacobiano* es nulo, por tanto no existe su inversa.

⁷Técnica de análisis numérico para determinar la curva que mejor describa un conjunto de datos independientes.

Jacobiano tiene dimensión rectangular por lo que su inversión no es posible, pero existen métodos para representar su “inversa” como el popularmente conocido *Moore-Penrose pseudoinversa* o simplemente *pseudoinversa*, el cual define, para $J \in \mathbb{R}^{m \times n}$ con $m = 6$ generalmente y J de rango⁸ completo, lo siguiente.

- Sí $m < n$ y $\text{rank}(J) = m$, entonces $(JJ^T)^{-1}$ existe y

$$J^+ = J^T(JJ^T)^{-1}, \quad (2.10)$$

quien representa la *pseudoinversa derecha* de J , dado que $JJ^+ = I$.

- Sí $m > n$ y $\text{rank}(J) = n$, entonces $(J^T J)^{-1}$ existe y

$$J^+ = (J^T J)^{-1} J^T, \quad (2.11)$$

quien representa la *pseudoinversa izquierda* de J , dado que $J^+ J = I$.

La *pseudoinversa* no tiene la propiedad conmutativa, por ejemplo, para (2.10) tenemos que $J^+ J \neq I$ y la inversa de (2.6) puede escribirse como

$$\dot{q} = J^+ V_n + (I - J^+ J)b, \quad (2.12)$$

donde $b \in \mathbb{R}^n$ es un vector arbitrario.

La representación (2.12) puede demostrarse fácilmente si es reemplazado en (2.6). En general, para $m < n$, $(I - J^+ J) \neq 0$, todos los vectores de la forma $(I - J^+ J)b$ se encuentran en el espacio nulo de J , es decir, si las articulaciones se mueven con una velocidad de la forma $(I - J^+ J)b$ entonces el extremo del robot permanecerá fijo. Si el objetivo es reducir al mínimo el vector velocidad asociado a las articulaciones entonces se elige $b = 0$, esto puede verse aplicando

⁸Número de filas o columnas que son linealmente independientes.

la desigualdad triangular

$$\begin{cases} \|\dot{q}\| = \|J^+V_n + (I - J^+J)b\|, \\ \|\dot{q}\| \leq \|J^+V_n\| + \|(I - J^+J)b\|. \end{cases} \quad (2.13)$$

2.3. Dinámica

La dinámica estudia el movimiento de un cuerpo rígido considerando las fuerzas y/o pares que lo produce [21].

2.3.1. Dinámica del Mecanismo

La *dinámica del mecanismo* hace referencia al movimiento de la estructura del robot serial de n -eslabones, conectados por articulaciones, producido por fuerzas y/o pares. Cada eslabón está caracterizado, adicionalmente de sus parámetros DH (ver Tabla 2.3), por sus parámetros físico: m_i , ${}^i r_c$ y I_{c_i} , cuyas definición se muestran en la Tabla 2.5. Por lo general, se suele encontrar las fuerzas y/o pares totales en cada articulación del robot a partir de la información de las posiciones, velocidades y aceleraciones de las variables asociados a las articulaciones, a esta relación se le conoce como *dinámica inversa*. Una forma de encontrar la dinámica de forma elegante, para un robot de pocos *grados de libertad*, es utilizando la formulación de Euler-Lagrange, el cual es una interpretación de la energía involucrada en un sistema de cuerpo rígido [20]. Esta formulación consta de 5 pasos, los cuales describen para cada eslabón i lo siguiente.

Paso 1. Tensor de inercia

$$\mathcal{I}_i = {}^0 R_i I_{c_i} {}^0 R_i^T. \quad (2.14)$$

Paso 2. Velocidad del centro de masa

$$V_{c_i} = (v_{c_i}, \omega_{c_i})^T. \quad (2.15)$$

Tabla 2.5: Definición de los parámetros físicos de un eslabón

Parámetro	Definición
m_i	Masa
${}^i r_{c_i}$	Vector de posición del centro de masa (c) respecto de su $\{S_i\}$
I_{c_i}	Momento de inercia del centro de masa (c) alineado con su $\{S_i\}$

- *Velocidad lineal del centro de masa*

$$v_{c_i} = \left(J_v \Big|_{r_{c_i}} \right) q_i, \quad (2.16)$$

donde $r_{c_i} := {}^0 r_{c_i}$ representa el vector de posición del centro de masa del eslabón i respecto de la base del robot y se calcula como

$$r_{c_i} = {}^0 R_i {}^i r_{c_i} + {}^0 p_i. \quad (2.17)$$

- *Velocidad angular del centro de masa*

$$\omega_{c_i} = J_{\omega_i} q_i, \quad (2.18)$$

donde $\omega_{c_i} = {}^0 \omega_i$ para un cuerpo rígido (ver Apéndice A.5).

Paso 3. Energía cinética

$$\mathcal{K} = \sum_{i=1}^n (\mathcal{K}_{t_i} + \mathcal{K}_{r_i}). \quad (2.19)$$

- *Energía cinética de traslación del centro de masa*

$$\mathcal{K}_{t_i} = \frac{1}{2} v_{c_i}^T m_i v_{c_i}. \quad (2.20)$$

- *Energía cinética de rotación del centro de masa*

$$\mathcal{K}_{r_i} = \frac{1}{2} \omega_{c_i}^T \mathcal{I}_i \omega_{c_i}. \quad (2.21)$$

Paso 4. **Energía potencial**

$$\mathcal{U} = \sum_{i=1}^n \mathcal{U}_i, \quad (2.22)$$

donde \mathcal{U}_i es la *energía potencial* del centro de masa del eslabón i , y se calcula como

$$\mathcal{U}_i = -m_i g^T p_{c_i}, \quad (2.23)$$

donde g es el vector de aceleración de la gravedad.

Paso 5. **Dinámica**

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i, \quad (2.24)$$

donde τ_i es la fuerza o par aplicada en q_i , y \mathcal{L} se le conoce como *Lagrangiano* y se calcula como

$$\mathcal{L} = \mathcal{K} - \mathcal{U}. \quad (2.25)$$

También, la *dinámica* puede escribirse de forma canónica en el espacio de articulaciones como

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau, \quad (2.26)$$

donde $D(q)$ es la *matriz de inercia* del robot, de orden $n \times n$, simétrica, definida positiva para q_i y sus elementos están asociados a los coeficientes de la variable \ddot{q}_i ; $C(q, \dot{q})\dot{q}$ es el *vector de fuerzas centrífugas y de coriolis*, de orden $n \times 1$ y sus elementos están asociados a los coeficiente de la variable \dot{q}_i ; $G(q)$ es el *vector de fuerza gravitacional*, de orden $n \times 1$ y sus elementos están asociados a los coeficientes de la constante g ; y τ es el *vector de fuerza o pares* generados en las articulaciones y es de orden $n \times 1$.

PTU-CAM

Los parámetros físicos se calculan a partir del modelo en *Solidwork* y estos se muestran en la Tabla 2.6.

Tabla 2.6: Parámetros físicos del PTU-CAM

Eslabón	$m_i \times 10^{-3}$ [Kg]	$r_c \times 10^{-3}$ [m]	$I_{c_i} \times 10^{-9}$ [Kg m ²]
1	60.65	$\begin{pmatrix} 0 \\ 17.44 \\ 1.43 \end{pmatrix}$	$\begin{pmatrix} 4841.69 & 0.01 & 0.06 \\ 0.01 & 1953.65 & -238.0 \\ 0.06 & -238.0 & 4107.58 \end{pmatrix}$
2	136.92	$\begin{pmatrix} 13.03 \\ 0.17 \\ 3.46 \end{pmatrix}$	$\begin{pmatrix} 4707.22 & -322.67 & 0.25 \\ -322.67 & 2115.74 & -0.04 \\ 0.25 & -0.04 & 5523.61 \end{pmatrix}$

2.3.2. Dinámica del Actuador

La *dinámica del actuador* permite encontrar la velocidad angular del motor DC a partir de su alimentación proporcionada.

El motor DC es un mecanismo de electromagnetismo basado en la ley de fuerza de *Lorentz*, el cual define que cualquier cable que transporte corriente dentro de un campo magnético, experimenta una fuerza mecánica proporcional a la corriente y a la fuerza del campo magnético perpendicular a éste, causando un par o torque [23].

Para modelar el motor DC (ver Figura 2.9) se considera dos aspectos: el *modelo eléctrico* (ligado al circuito eléctrico interno) y el *modelo mecánico* (ligado al rotor).

El *modelo eléctrico* del motor DC puede escribirse como

$$V_m - K_b \dot{\theta} = L_a \frac{di}{dt} + R_a i, \quad (2.27)$$

donde V_m es la fuente de voltaje de alimentación, i es la corriente en la bobina de armadura, R_a es la resistencia de la bobina de armadura, L_a es la inductancia de

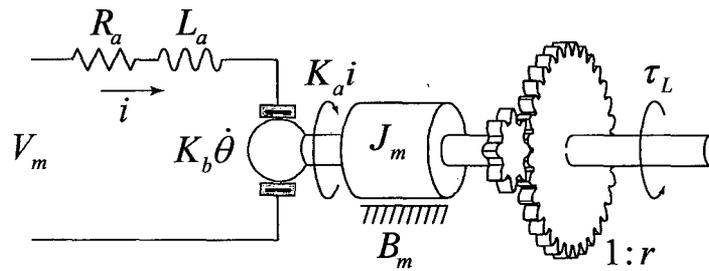


Figura 2.9: Modelo del motor DC.

la bobina de armadura, K_b es la constante proporcional entre la velocidad angular del motor DC y la fuerza contra electromotriz (EMF), y θ es la posición angular del eje del motor DC.

El *modelo mecánico* del motor DC puede escribirse como

$$J_m \ddot{\theta} + B_m \dot{\theta} + \frac{1}{r} \tau_L = K_a i, \quad (2.28)$$

donde J_m es la inercia del rotor, B_m es el coeficiente de amortiguamiento del rotor, K_a es la constante proporcional entre la corriente en la bobina de armadura y el torque generado por el motor DC, $r \gg 1$ es la relación de transmisión de la caja reductora, y τ_L es el torque neto o de carga aplicado en la salida del eje de la caja reductora.

Mediante la aplicación de la transformada de Laplace en (2.27) y (2.28), se encuentra el modelo del motor DC (ver Figura 2.9), cuya Función de Transferencia (TF) característica, considerando $K_a = K_b = K$ y $\tau_L = 0$ para simplificar, es

$$\frac{\dot{\theta}}{V_m} = \frac{K}{J_m L_a s^2 + (B_m L_a + J_m R_a) s + B_m R_a + K^2}, \quad (2.29)$$

La Tabla 2.7 muestra la definición de los parámetros de un motor DC generalmente obtenidos en diversas teorías de algoritmos de *identificación de parámetros*,

como el modelo AR⁹ o el modelo OE¹⁰.

Tabla 2.7: Definición de los parámetros del motor DC

Parámetro	Definición
K	Constante proporcional $K = K_a = K_b$
J_m	Inercia del rotor
B_m	Coefficiente de amortiguamiento del rotor
L_a	Inductancia de la bobina de armadura
R_a	Resistencia de la bobina de armadura
r	Relación de transmisión (fijo)

PTU-CAM

Los parámetros para el servomotor AX-12 son tomados de la disertación [18], en donde se realiza la identificación a través de procedimientos más detallados. La Figura 2.10 muestra la respuesta de la función de transferencia del AX-12 para una entrada constante de 12 V, y donde se observa que la velocidad obtenida es de 58.44 RPM, muy cercano al real de 59 RPM (ver Tabla 2.1).

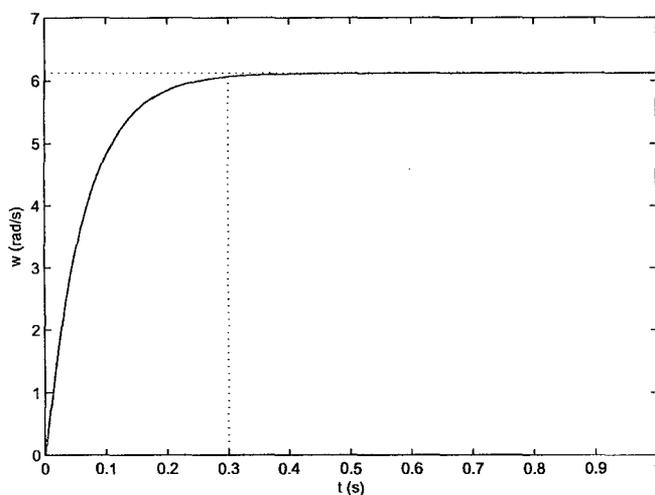


Figura 2.10: Respuesta (12V) del modelo de motor DC del AX-12.

⁹ *AutoRegresivo Model*, este es un caso especial del *ARMA Model (AutoRegressive Moving Average Model)* y especifica que la variable de salida depende linealmente de sus propios valores anteriores.

¹⁰ *Output-Error Model*, es un caso del *ARMA Model* con relación entrada/salida sin perturbación más ruido blanco aditivo en la salida.

Tabla 2.8: Parámetros de motor DC del AX-12

Parámetro	Valor
K	0.0071 V s/rad
J_m	1.4203×10^{-4} Kg m ²
B_m	0.0022 Kg m ² /s
L_a	0.0064 H
R_a	6.3029 Ω
r	254

2.4. Control Desacoplado de Articulaciones

La motivación de realizar el modelamiento del control interno de posición angular del servomotor AX-12, es obtener un modelo más completo del conjunto PTU-CAM, ya que sin éste se tendría una respuesta directa en la posición angular de las articulaciones, es decir, sin zona de establecimiento ni perturbación en el movimiento producido por los eslabones. Si bien la perturbación producido por los eslabones en el movimiento es mínimo, por lo pequeño del módulo, se quiere hacer notar este modelamiento para futuras aplicaciones con mayores dimensiones, siguiendo el mismo procedimiento.

El servomotor AX-12 tiene un dispositivo electrónico interno que permite controlar la posición angular del eje de salida de la caja reductora, de modo que un par o torque, aplicado en el mismo, tendrá únicamente efecto sobre el movimiento de éste, es decir, su control es de forma independiente o desacoplado. Esta técnica se le conoce como *control desacoplado de articulaciones* y es común emplear un controlador PID, en donde los pares de acoplamientos serán anulados por el controlador como si fueran perturbaciones [20]. Dentro de los pares de acoplamiento puede considerarse los pares generados por los eslabones del robot (dinámica del mecanismo).

El controlador utilizado en la disertación [18], para el modelo de una *articulación o motor DC*, es el PD reducido (ver Figura 2.11), el cual es un controlador

proporcional más una alimentación de velocidad que consigue la nulidad de la velocidad final [18], y cuya ley de control está dado por

$$V_m = K_p e_q - K_v \dot{q}_i, \quad (2.30)$$

donde $e_q = q_d - q_i$ es el error de la posición angular para cada articulación, q_d la posición angular deseada, K_p la ganancia proporcional, K_v la ganancia de velocidad o derivativa, y V_m el voltaje de alimentación [24].

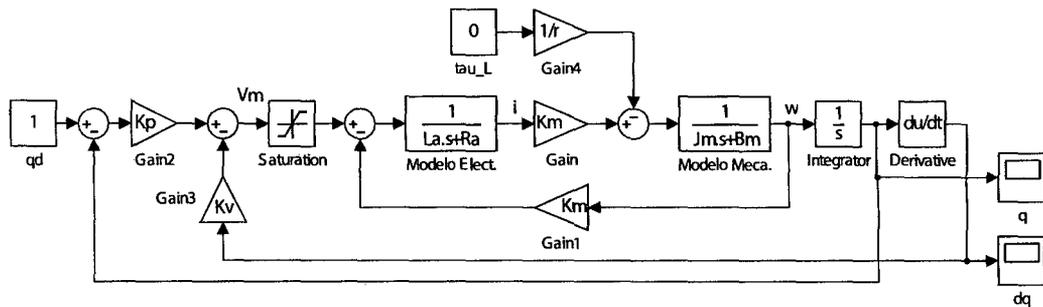


Figura 2.11: Control desacoplado de una articulación.

Los valores de K_p y K_v pueden encontrarse mediante una sintonización que satisfaga un determinado comportamiento en el modelo de la articulación. Para sintonizar estos parámetros se puede usar el *método manual*, en donde los parámetros son variados hasta que se cumplan especificaciones deseadas; o el *método de Ziegler-Nichols*, en donde los parámetros son encontrados de acuerdo a valores obtenidas en la respuesta del modelo real en lazo cerrado ante un escalón.

2.5. Modelo del Módulo

El modelo matemático del PTU-CAM, a partir de la integración de los modelos de cinemática, dinámica y el control desacoplado de articulaciones, antes encontrados; se visualiza en la Figura 2.12, donde el bloque “Dina. Inv. Cano. PTU-CAM” es la representación canónica de la dinámica inversa, encontrada en ecuación (2.26) y cuyo esquema se visualiza en la Figura 2.13; el bloque “Modelo

AX-12” es el modelo discretizado del actuador AX-12 descrita en la Figura 2.11, y el bloque “Disc. Deriv.” es la derivada discreta. El Capítulo 5 muestra los resultados de la simulación con este modelo discretizado.

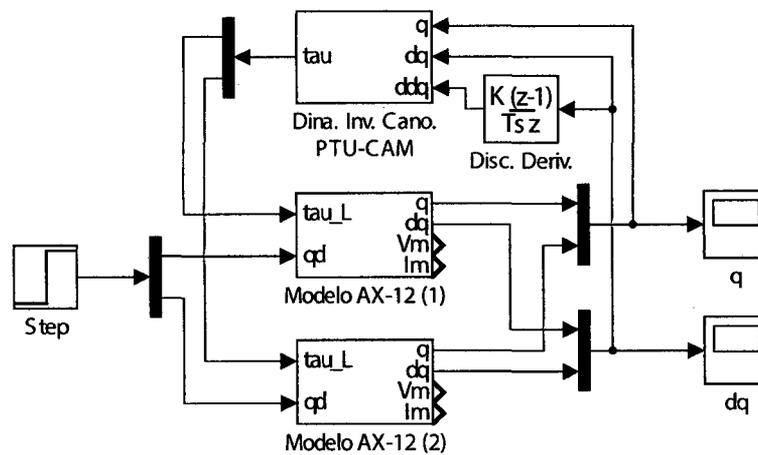


Figura 2.12: Modelo completo del PTU-CAM.

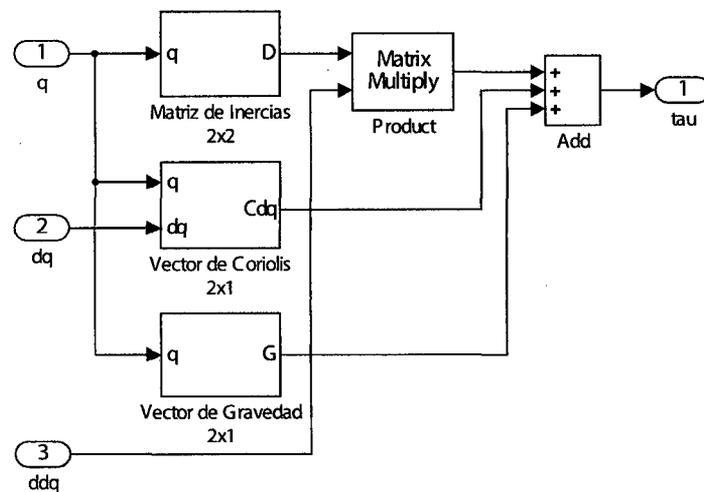


Figura 2.13: Dinámica inversa de la estructura del PTU-CAM.

CAPÍTULO 3

Visión por Computador

La *visión por computador*, también conocido como *visión artificial* o *machine vision*, tiene el propósito de extraer información del mundo físico a partir de imágenes obtenidas por una o varias cámaras digitales y entenderlo utilizando una computador. Este subcampo de la *inteligencia artificial* es muy amplio, que va desde la radiometría¹ hasta el estudio de reconstrucción tridimensional de entornos dinámicos. En [25] se encuentra información más extensa y detallada sobre éste tema.

Para no extender demasiado este Capítulo se menciona los principales fundamentos en la formación de una imagen a través del *modelamiento de cámara*, la obtención de los parámetros característicos de la cámara mediante la *calibración*, y las principales técnicas de *procesamiento de imágenes* como la conversión del espacio de color, conectividad de píxeles, morfología matemática y ubicación de regiones.

3.1. Modelamiento de cámara

El principio de trabajo de toda cámara es convertir una imagen óptica en señales eléctricas. La luz que el objeto refleja es recolectada por la lente y enfocada a un divisor de rayos (filtro de Bayer), el cual divide la luz blanca en luz roja,

¹Ciencia que estudia la medida de la radiación electromagnética.

verde y azul. Estos rayos son enviados a la matriz de sensores fotoeléctricos² de tipo CCD³, CMOS⁴ o un híbrido entre ambos, donde la luz RGB es transformada en señales RGB y convertidas en imágenes digitales por medio de un convertidor de señales análogo a digital.

El modelo más conocido para una cámara es el denominado cámara *Pinhole* o *Estenopeica* (ver Figura 3.1). En este modelo, la lente se considera como un agujero muy pequeño, como si se hubiese hecho por un alfiler (pinhole), situado en el centro focal de la lente y por donde los rayos de luz pasan a través de éste produciendo un imagen proyectada en un plano denominado *plano de imagen* [22]. Este modelo es ideal, por tanto no incluye distorsiones geométricas o imágenes borrosas causadas por la lente.

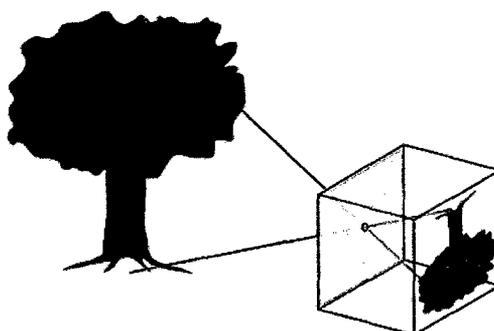


Figura 3.1: Principio de una cámara Pinhole o Estenopeica.

La Figura 3.2 muestra el esquema del modelo de cámara *pinhole*, donde los rayos convergen en el origen del sistema coordenado de cámara $\{C\}$, también conocido como *centro óptico* o *centro de proyección*, y una imagen no invertida se proyecta sobre el *plano de imagen* desplazado una distancia f sobre el eje óptico, denominado *distancia focal*. Considerando $C_r = (X, Y, Z)^T$ como el vector posición de un punto en el espacio, expresado respecto de la cámara, entonces su

²El *efecto fotoeléctrico* consiste en la emisión de electrones por sólidos (metal), líquidos o gases cuando absorben energía de la luz.

³Charge-Coupled Device.

⁴Complementary Metal-Oxide-Semiconductor.

proyección ${}^I r = (x, y)^T$, en el *plano de imagen* $\{I\}$, puede encontrarse utilizando semejanzas de triángulos como

$$\begin{cases} x = f \frac{X}{Z}, \\ y = f \frac{Y}{Z}. \end{cases} \quad (3.1)$$

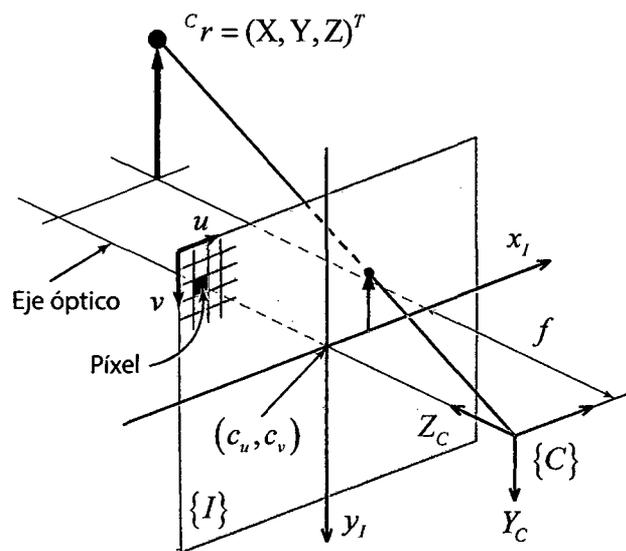


Figura 3.2: Modelo de cámara pinhole.

La imagen proyectada puede convertirse en digital relacionando ${}^I r = (x, y)^T$ con el *arreglo de sensores* o *matriz de píxeles* como

$$\begin{cases} u = \frac{x}{\rho_w} + c_u, \\ v = \frac{y}{\rho_h} + c_v, \end{cases} \quad (3.2)$$

donde (u, v) son las coordenadas de un píxel, ρ_w y ρ_h son el ancho y alto físico de cada píxel respectivamente, y (c_u, c_v) es el centro principal de la matriz de píxeles ubicado en el eje óptico.

3.1.1. Parámetros Intrínsecos

Las ecuaciones (3.1) y (3.2) pueden ser representados en forma matricial, como

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_u \\ 0 & f_y & c_v \\ 0 & 0 & 1 \end{pmatrix} C_r, \quad (3.3)$$

donde la matriz de orden 3×3 se le denomina *matriz de parámetros intrínsecos* de la cámara, $f_x = f/\rho_w$ y $f_y = f/\rho_h$ son las distancia focales en píxeles proporcionales a la distancia focal f , y λ es un escalar que convierte la proyección del punto en coordenadas (u, v) . Los parámetros f_x , f_y , c_u y c_v son conocidos como *parámetros intrínsecos* de la cámara, los cuales son constantes para una determinada cámara e invariantes respecto del movimiento de ésta.

3.1.2. Parámetros de Distorsión

El modelo de *cámara pinhole* se considera como ideal pero en realidad existen distorsiones producidas por las imperfecciones de la lente, siendo más notorio en cámara comerciales de bajo costo. Las distorsiones incluyen aberración cromática (mezcla de colores), esféricas o de astigmatismo (variación en el enfoque a través la escena), y distorsiones geométricas, donde los puntos en el plano de imagen son desplazados de donde deberían estar [1].

La *distorsión geométrica* es el efecto más común y problemático en aplicaciones de *visión por computador*, y se compone de dos tipos. El primero se denomina *distorsión radial*, el cual es producida por defectos en la construcción de la lente provocando el efecto *ojo de pez*, en donde los píxeles situados en el centro de la imagen sufren pequeñas modificaciones y los más alejados sufren aberraciones cada vez mayores (ver Figura 3.3); éste puede ser aproximado por

$$\delta r = k_1 r^3 + k_2 r^5 + k_3 r^7 + \dots, \quad (3.4)$$

donde δr es el error radial en un píxel (u, v) , r es la distancia de un píxel al centro principal (c_u, c_v) , y $k_{i=1,2,3}$ son constantes; la aproximación está bien representada por los tres primeros términos del polinomio [1]. El segundo tipo se denomina **distorsión tangencial**, el cual es producida por falta de paralelismo entre la lente de la cámara y el *array de sensores* de la misma generando imágenes con forma trapezoidal (ver Figura 3.4); éste puede ser aproximado por

$$\begin{cases} \delta t_u = 2p_1uv + p_2(r^2 + 2u^2), \\ \delta t_v = p_1(r^2 + 2v^2) + 2p_2uv. \end{cases} \quad (3.5)$$

donde δt_u y δt_v son errores tangenciales en la dirección de u y v respectivamente ($p_{i=1,2}$ son constantes).

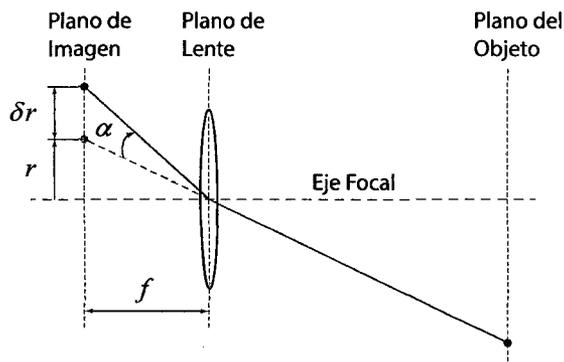


Figura 3.3: Distorsión radial en la cámara.

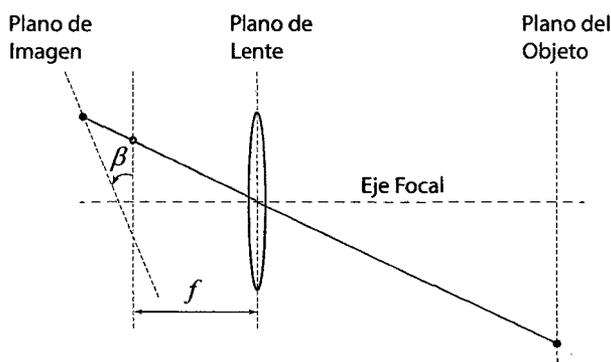


Figura 3.4: Distorsión tangencial en la cámara.

Juntando (3.4) y (3.5) puede escribirse la corrección a la *distorsión geométrica*, como

$$\begin{cases} \hat{u} = u + u(k_1r^2 + k_2r^4 + k_3r^6) + 2p_1uv + p_2(r^2 + 2u^2), \\ \hat{v} = v + v(k_1r^2 + k_2r^4 + k_3r^6) + p_1(r^2 + 2v^2) + 2p_2uv, \end{cases} \quad (3.6)$$

donde (\hat{u}, \hat{v}) son las coordenadas del píxel corregido. Los parámetros $k_{i=1,2,3}$ y $p_{i=1,2}$ son conocidos como **parámetros de distorsión** de la cámara, los cuales son constantes para un determinada cámara e invariantes respecto del movimiento de la misma.

En resumen, los parámetros que caracteriza el modelo de una cámara puede observarse en la Tabla 3.1. Además, la resolución de cámara, el cual es el número de píxeles que existe en un imagen digital, es un dato que proviene directo fábrica.

Tabla 3.1: Parámetros de cámara

Tipo	Símbolo	Definición
Intrínseco	(f_x, f_y)	Distancias focales
Intrínseco	(c_u, c_v)	Centro principal de imagen
Distorsión radial	(k_1, k_2, k_3)	Coefficientes
Distorsión tangencial	(p_1, p_2)	Coefficientes

3.2. Calibración de cámara

La calibración es el proceso para determinar los parámetros *intrínsecos* y de *distorsión* de la cámara (ver Tabla 3.1). Se suele utilizar un patrón (plantilla de calibración) con el que se plantea una serie de ecuaciones relacionando la ubicación y orientación de cada punto característico respecto de la cámara. Los parámetros que representan la ubicación y orientación del sistema coordinado adjunto a la plantilla de calibración respecto de la cámara, son conocidos comúnmente como **parámetros extrínsecos** de la cámara.

Generalmente se utiliza un *tablero de ajedrez* en donde se distingue varios puntos ordenados entre si, respecto del sistema coordenado adjunto a este $\{T\}$, que ayuden a reducir el numero de ecuaciones planteadas (ver Figura 3.5). Estos puntos pueden ser reconocidos manualmente en la imagen o utilizando el algoritmo de Harris [26].

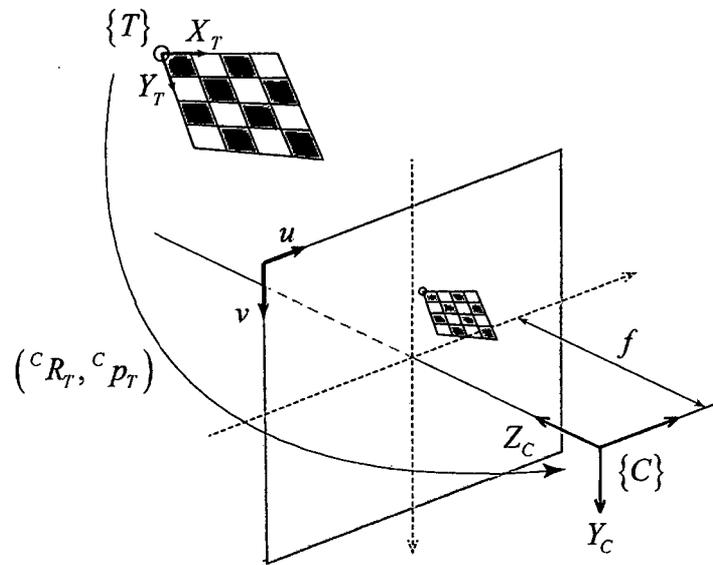


Figura 3.5: Esquema de calibración interna de una cámara.

De la Figura 3.5 se puede plantear la ecuación

$${}^C r = {}^C R_T {}^T r + {}^C p_T, \quad (3.7)$$

donde ${}^C R_T$ y ${}^C p_T$ son la matriz de rotación y el vector de posición del tablero $\{T\}$ respecto de la cámara $\{C\}$ respectivamente, ${}^T r$ es el vector posición de un punto en el tablero (esquina de un cuadrado) expresado en $\{T\}$, y ${}^C r$ es el vector de posición del mismo punto expresado en $\{C\}$.

La matriz de rotación puede ser represento por 3 parámetros, por ejemplo, la representación de Rodrigues (ver Tabla A.4) reduce a 6 los *parámetros extrínsecos* de la cámara. Teniendo 9 parámetros fijos incógnitas (4 intrínsecos y

5 de distorsión) más 6 parámetros extrínsecos por postura del tablero respecto de la cámara, entonces el número total de incógnitas sería $6N_I + 9$, donde N_I es el número de imágenes tomadas por la cámara de cada posturas del tablero respecto de la misma. También, las ecuaciones (3.6), (3.3) y (3.7) proporcionan 2 ecuaciones independientes por punto del tablero, con coordenadas (u, v) en la imagen, teniendo $2N_T N_I$ ecuaciones independientes, donde N_T es el número de puntos del tablero representados en la imagen. Por tanto, para encontrar estos parámetros debe cumplirse la condición

$$2N_I(N_T - 3) \geq 9. \quad (3.8)$$

En [1] y [22] muestran ecuaciones más desarrolladas y métodos numéricos para encontrar los parámetros incógnitas. También, se puede utilizar la herramienta para calibración de cámara (Camera Calibration Toolbox) de *Matlab* o la librería de procesamiento de imagen *OpenCV*.

3.3. Cámara Time-of-Flight

La cámara TOF es una tecnología nueva y con mucho precisión para obtener información de la escena 3D en tiempo real. La tecnología TOF se basa en el principio de medición del tiempo que tarda la luz en viajar desde la cámara hasta un objeto y volver (tiempo de vuelo), con esta información se calcula la distancia intermedia. Se trata de una cámara tridimensional al estilo de la estereovisión y los escáneres de láser, pero más económica y adaptable.

Generalmente, existen dos tipos de estimación TOF: el de impulso y la de modulación continua. En el contexto de *modulación de impulsos*, solo un único impulso de luz es emitida, lo que implica altas exigencias en la exactitud de la detección para determinar el tiempo exacto de retardo (velocidad de la luz $c = 3 \times 10^8$ m/s). Por otro lado, la *modulación continua* estima el desplazamiento

de fase entre la señal emitida y su respuesta, siendo el sistema es menos exigente, pero el rango de la medición se limita por la periodicidad de la señal emitida. La Tabla 3.2 muestra las ventajas y desventaja de cada estimación.

Los sensores TOF obtienen dos tipos de medidas: un mapa de la distancia (generalmente mono canal de 16 bit) y un mapa de la intensidad de la luz (generalmente en tres canales de 24 bit), los cuales son sincronizados y calibrados para brindar una imagen 3D de color.

Tabla 3.2: Ventajas y desventajas de los tipos de estimación TOF

Modulación	Ventajas	Desventajas
Impulsos	<ul style="list-style-type: none"> • Medición directa de tiempo de vuelo. • Pulsos de luz de alta frecuencia limitan la influencia de la iluminación de fondo. • La dirección de iluminación y observación son colineales. 	<ul style="list-style-type: none"> • Requiere alta precisión en la medida de tiempo. • La medición del pulso de luz de retorno es inexacta, debido a la dispersión de la luz. • Dificultad para generar impulsos de luz cortos con aumento rápido y tiempos de caída. • Las fuentes de luz que pueden utilizarse (p. ej., láseres) sufren bajas tasas de repetición de impulsos.
Onda Continua	<ul style="list-style-type: none"> • Variedad de fuentes de luz disponibles y no requiere pulsos cortos/fuertes. • Aplicable para diferentes técnicas de modulación (aparte de la frecuencia) • Rango y amplitud de imágenes simultáneas. 	<ul style="list-style-type: none"> • Tiempo de integración necesario para reducir el ruido. • Las tasas de fotogramas limitadas por el tiempo de integración. • Desenfoque de movimiento causado por largo tiempo de integración.

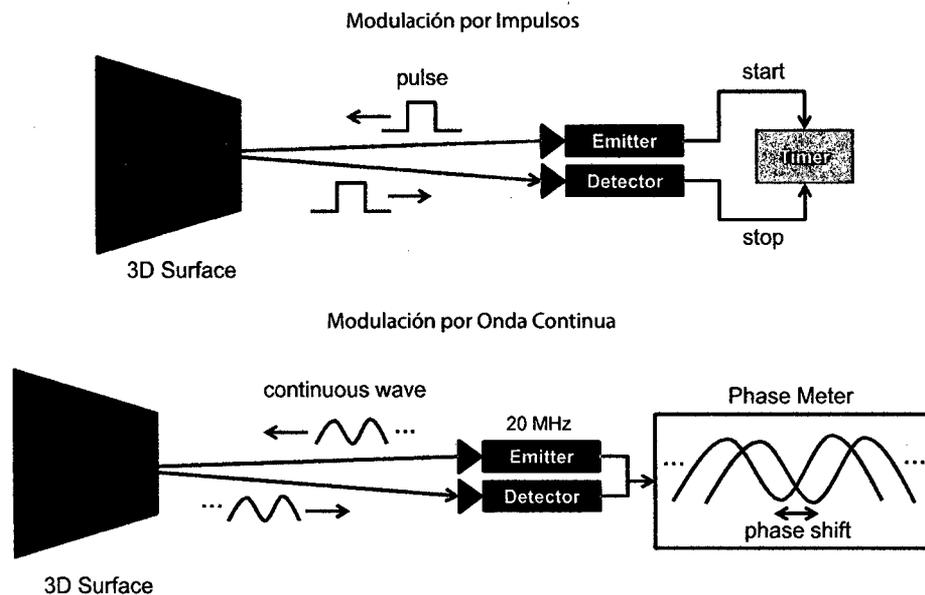


Figura 3.6: Tipos de modulación en una cámara TOF.

3.4. Procesamiento de Imágenes

El *procesamiento de imágenes* se encarga de mejorar los aspectos y hacer notar detalles deseados en un imagen con el uso técnicas y algoritmos para su posterior interpretación. En [27] se muestra los principios de diversas técnicas de *procesamiento de imágenes*.

Una imagen digital simple es la representación matricial bidimensional de una escena captada por una cámara. Ésta se compone por un número determinado de píxeles (resolución) donde cada uno representa, en valor numérico entero, la intensidad de luz de una porción de la escena. Por lo general, este valor numérico tiene un longitud de 8 bits de almacenamiento en el computador, es decir, puede tomar 256 valores (0: color negro, 255: color blanco). Esta imagen se le denomina *imagen de nivel de gris*.

Un tipo especial de imagen, que se obtiene a partir de una *imagen de nivel gris*, es la denominada *imagen binaria*, la cual esta formada por dos valores: 0 para el negro y 1 para el blanco. El proceso para obtener esta imagen se le conoce

como *binarización* y consiste en reducir el valor de intensidad de la luz a niveles de 0 y 1, usando un valor umbral (threshold) de división en el nivel de gris. Esta *imagen binaria* se utiliza como *mascara de imagen* y define regiones de interés.

La imagen obtenida por una cámara de color suele ser en tonalidades rojo, verde y azul (RGB), es decir, *imagen de color*, donde cada una de estas tonalidades de color primario se conoce como *canal* y es una *imagen de nivel gris* que representa un color primario. Por tanto, podemos decir que una imagen de color esta formado por 3 canales RGB y tiene una longitud de 24 bits.

3.4.1. Segmentación de Color

Antes de segmentar el color verde deseado, para esta tesis, se debe realizar la transformacion del espacio de color RGB al espacio de color CIE⁵ LAB (CIE $L^*a^*b^*$). Este espacio se estableció con la intención de reproducir con mayor precisión la gama de colores que el ojo humano es capaz de percibir, basándose en que los receptores del ojo humano perciben el color a razón de la luminosidad (L^*) y los espacios de color rojo-verde (a^*) y amarillo-azul (b^*). La principal ventaja de utilizar este espacio esférico de color (ver Figura 3.7) es que las componentes de color (a^*, b^*) son independiente de la luminosidad (L^*), muy variante en muchos entorno dinámicos, permitiendo que los colores no sean alterados. Este estándar es muy utilizado en la gestión de color, en lazo cerrado, de muchos dispositivos digitales [28]. Por tanto, usando el espacio CIE LAB, los colores que más se acerquen a los primarios serán reconocidos y segmentados con mayor facilidad.

Para convertir el espacio de RGB a CIE LAB primero se debe convertir a un espacio de color intermedio como el CIE XYZ, conocido como valores triestímulo y se utilizan de manera que las coordenadas de cualquier luz real sean siempre positivos. En [29] se muestran las ecuaciones de conversión con mayor descripción.

La Figura 3.8 muestra el procedimiento de segmentación del color verde para

⁵Comission Internationale de l'Éclairage (Comisión internacional de iluminación).

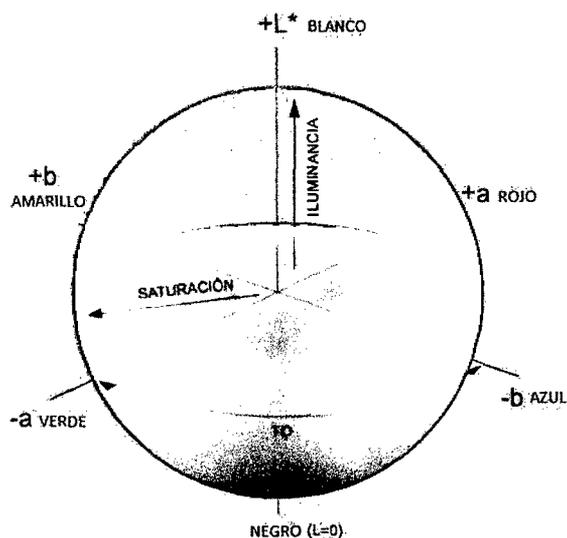


Figura 3.7: Espacio de color CIE $L^*a^*b^*$.

encontrar la máscara de imagen deseada, usando el espacio de color CIE LAB y técnicas morfológicas (dilatación y erosión), que ayudan a eliminar pequeños grupos de píxeles aislados de la región de interés, considerados como ruido (ver Figura 3.9).

```

Data: Imagen RGB  $I_{RGB}$  de  $w \times h$ 
Result: Máscara binaria  $I_{mask}$ 

 $I_{LAB} \leftarrow \text{Conversion}(I_{RGB});$ 
 $[I_L, I_a, I_b] \leftarrow \text{SeparacionCanal}(I_{LAB});$ 
for  $i \leftarrow 0$  to  $w - 1$  do
  | for  $j \leftarrow 0$  to  $h - 1$  do
  | | if  $a_{min} < I_a(i, j) < a_{max} \cap b_{min} < I_b(i, j) < b_{max}$  then
  | | |  $I_{mask}(i, j) \leftarrow 1;$ 
  | | | else
  | | | |  $I_{mask}(i, j) \leftarrow 0;$ 
  | | | end
  | | end
  | end
end

 $I_{mask} \leftarrow \text{Erosion}(I_{mask});$ 
 $I_{mask} \leftarrow \text{Dilatacion}(I_{mask});$ 

```

Figura 3.8: Algoritmo de segmentación del color verde.

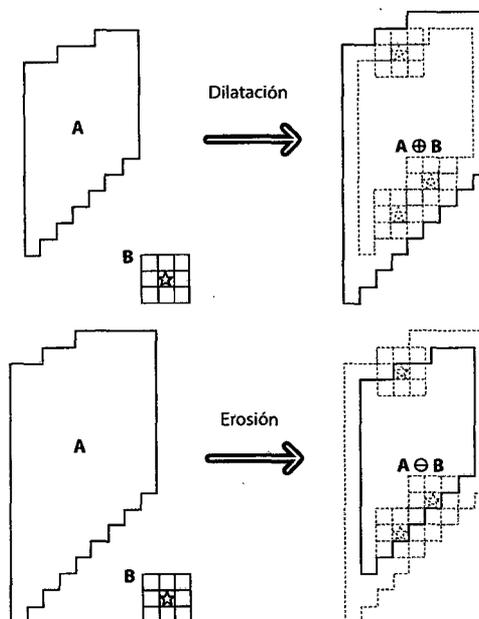


Figura 3.9: Morfología (dilatación y erosión).

3.4.2. Clasificación de Regiones

Se dice que un píxel es de *vecindad*-4 si no se considera sólo los píxeles diagonales, o de *vecindad*-8 si se considera todos sus vecinos, excepto los que están en el extremo de la imagen. La conectividad entre dos píxeles, de una imagen binaria, define la existencia de un camino mutuo entre ambos, a través de su tipo de vecindad [22]. Por tanto, una región en la imagen está formada por dos o más píxeles que tenga algún tipo de conectividad entre ellos (ver Figura 3.10).

La clasificación de regiones o grupos de píxeles, en una máscara de imagen, se hace mediante el proceso de etiqueto de cada una, para luego quedarse con la región de interés y eliminar las sobrantes (ver Figura 3.10). Una forma práctica es encontrando la región que tenga mayor *área* respecto de los demás, suponiendo que ésta representa al único objeto importante en la escena. Para encontrar la ubicación del centroide de la región de interés en la imagen, características visuales, se puede utilizar *cálculos de momentos* en ésta región [22]. El momento de

orden ij , de una región en la imagen, se define por

$$m_{ij} = \sum_{u,v} u^i v^j \mathcal{I}(u, v), \quad (3.9)$$

donde $\mathcal{I}(u, v)$ representa al valor de la intensidad del píxel (u, v) y para una imagen binaria (máscara) es la unidad.

A partir de (3.9) se puede obtener los momentos m_{01} , m_{10} y m_{00} , donde m_{00} representa al *área* de la región. Por tanto, el centroide de una región (\bar{u}, \bar{v}) en la imagen se encuentra por

$$\begin{cases} \bar{u} = \frac{m_{10}}{m_{00}}, \\ \bar{v} = \frac{m_{01}}{m_{00}}. \end{cases} \quad (3.10)$$

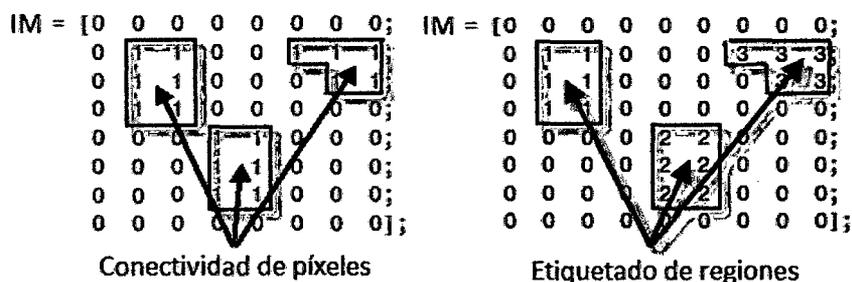


Figura 3.10: Conectividad de píxeles y Etiquetado de regiones.

Finalmente, la Figura 3.11 muestra el procedimiento de la identificación de la región de interés, a través de la clasificación de regiones.

PTU-CAM

El resultado de la implementación de los Algoritmos 3.8 y 3.11, que permiten ubicar el centroide de un objeto de color verde en el Frame de la cámara, se muestra en la Figura 3.12. Los valores iniciales de los algoritmos son contantes e iguales para cualquier imagen de entrada. Además, la Figura 3.13 muestra el resultado del algoritmo de identificación de la región de interés para diferentes tonos de iluminación artificial, y en la Tabla 3.3 sus respectivos valores numéricos, donde se observa que el algoritmo puede reconocer el 63% del objeto para una baja

Data: Máscara binaria I_{mask} de $w \times h$
Result: Centro de la región de interés (\bar{u}, \bar{v})

```

 $I_{label} \leftarrow \text{ProcesoEtiquetado}(I_{mask});$ 
 $n \leftarrow \text{NumeroRegion}(I_{label});$ 
 $region \leftarrow \text{SeparacionRegion}(I_{label});$ 
for  $i \leftarrow 1$  to  $n$  do
   $A = \text{Area}(\text{region}(i));$ 
  if  $A < a_{min}$  then
     $a_{min} = A;$ 
     $[\bar{u}, \bar{v}] = \text{MomentoCentral}(\text{region}(i));$ 
  end
end

```

Figura 3.11: Algoritmo de Identificación de la región de interés.

iluminación artificial (40%).

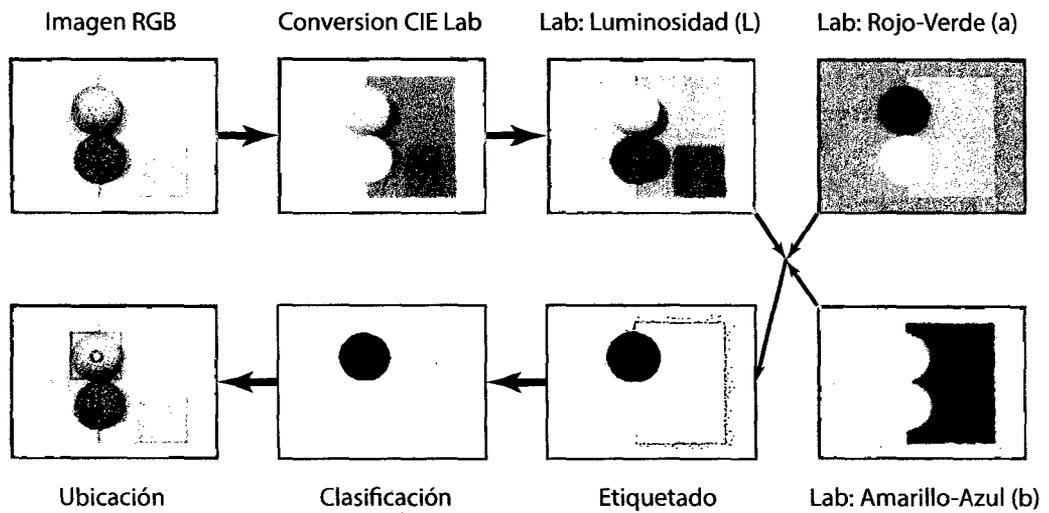


Figura 3.12: Obtención de la posición de un objeto de color verde deformable en el Frame de la cámara.

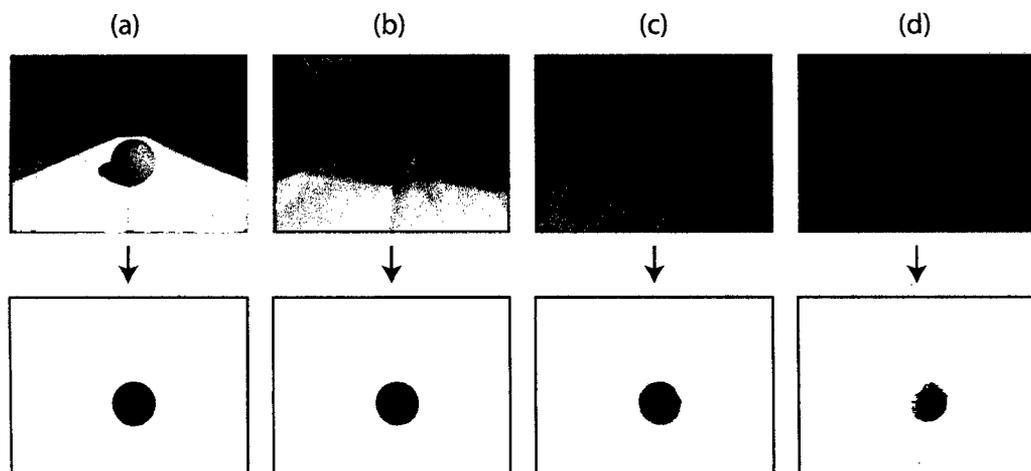


Figura 3.13: Imagen filtrada con el algoritmo de identificación de la región de interés, para diferentes tonos de iluminación artificial.

Tabla 3.3: Áreas de las máscaras obtenidas del entorno filtrado para diferentes tonos de iluminación artificial.

Frame	Iluminación (%)	Área (píxel)
a	100	10822
b	90	10643
c	70	10124
d	40	6867

CAPÍTULO 4

Control Basado en Visión

El *control basado en visión* se refiere al uso de la información obtenido en un sistema de *visión por computador* para controlar los movimientos de un mecanismo y realizar tareas correspondientes [30].

En este campo se distingue a los robots que utilizan los sistemas de *visión por computador* para obtener información visual única que active una secuencia de trabajo del mismo, comúnmente denominado *control visual en lazo abierto*; y aquellos que utilizan la información visual para realimentar su controlador principal, comúnmente conocido como *control visual en lazo cerrado*, *control servovisual* o *visual servoing*. Éste último ofrece mejores ventajas en relación al de *lazo abierto*, principalmente por ser independiente de la precisión de los sensores de visión y la del robot. Sin embargo, cerrar el lazo de control aumenta las posibilidades de inestabilidad, complicando en gran medida su desarrollo [31].

Este Capítulo realiza un pequeño resumen de los *enfoques del control basado en visión*, luego se revisa de forma más completa el *control basado en imagen*, y por último, la *calibración cámara-robot*, que permite encontrar la relación mecánica entre ambos.

4.1. Enfoques del Control Basado en Visión

Las técnicas de *control basado en visión* se enfocan principalmente en la *ubicación* y el *uso de la información* del sistema de visión por computador [1], los cuales son descritos a continuación.

1. *Ubicación* (ver Figura 4.1)

- ***Cámara externa al robot***, también conocida como *eye-to-hand*, se refiere que la cámara no tiene ninguna relación mecánica con el robot, pero es conocida la ubicación de ésta respecto de la base del robot. Por lo general, esta cámara no presenta ningún movimiento respecto al robot (cámara fija) y está situada de manera que los objetivos situados en el espacio de trabajo del robot se encuentren dentro de su campo de visión.
- ***Cámara en el extremo del robot***, también conocida como *eye-in-hand*, se refiere que la cámara este adjunta (soldada) al extremo del robot para que éste defina su campo de visión.

2. *Uso de la información*

- ***Señal de entrada***
 - ***Ver y mover estático***, también conocida como *static look-then-move*, es una configuración secuencial de control en lazo abierto donde el sistema de visión primero observa y reconoce, y luego envía la información al controlador interno del robot para que realice el movimiento correspondiente (ver Figura 4.2).
 - ***Ver y mover dinámico***, también conocida como *dynamic look-then-move*, es una configuración en lazo cerrado donde la información visual, obtenida por el sistema de visión por computador,

es tratada como realimentación por un controlador, sujeto a un entrada de referencia, previo al controlador interno del robot (ver Figura 4.2).

- ***Control visual directo***, también conocida como *direct visual servoing*, es una configuración en lazo cerrado donde la información visual, obtenida por el sistema de visión por computador, es tratada como realimentación por un solo controlador, sujeto a una entrada de referencia, que actúa directamente en las articulaciones del robot (ver Figura 4.3).
- ***Característica extraída***
 - ***Control visual basado en posición***, también conocida como *position-based visual servoing* (PBVS), se refiere que las características extraídas por el sistema de visión por computador son traducidas a una localización o trayectoria en el espacio cartesiano, para ser usada como entrada a la configuración de control correspondiente (ver Figura 4.2).
 - ***Control visual basado en imagen***, también conocida como *image-based visual servoing* (IBVS), se refiere que las características extraídas por el sistema de visión por computador están en el espacio de la imagen (p. ej. puntos, líneas, curvas, círculos, entre otros) y son usados como entrada a la configuración de control correspondiente (ver Figura 4.3).
 - ***Control visual híbrido***, también conocida como *hybrid-based visual servoing* (HBVS) o *2 1/2 D visual servoing*, es la combinación de los dos enfoques anteriores donde el error a minimizar se especifica tanto en el plano de imagen como en el espacio cartesiano (ver Figura 4.4).

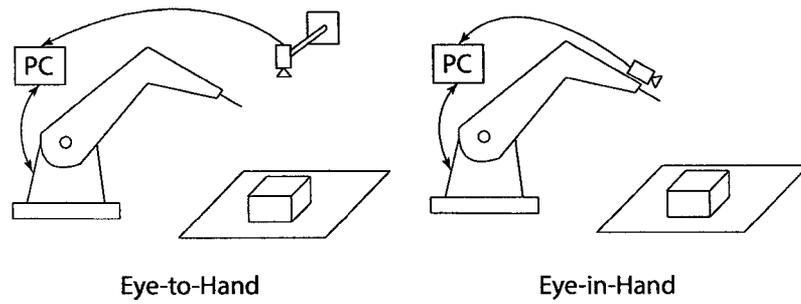


Figura 4.1: Ubicación del sistema de visión.

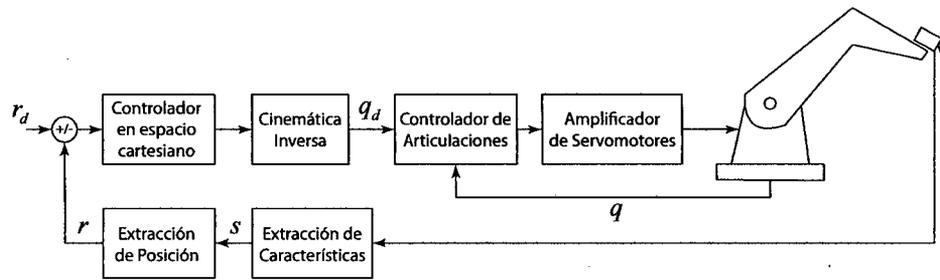


Figura 4.2: Control ver y mover estático/dinámico basado en posición.

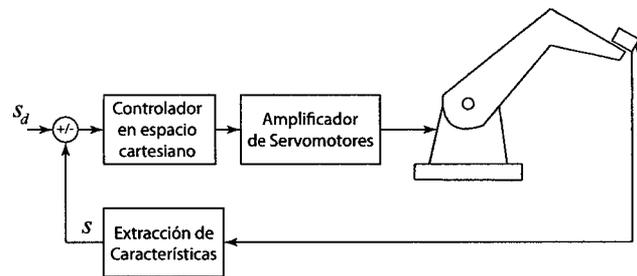


Figura 4.3: Control visual directo basado en imagen.

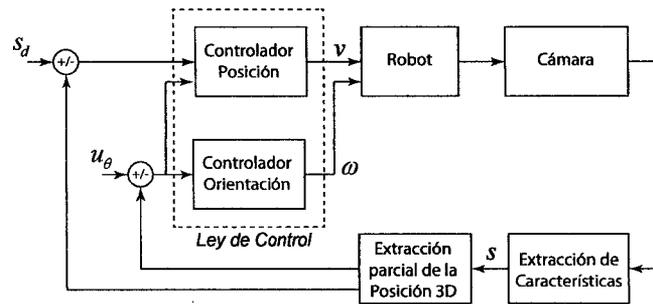


Figura 4.4: Control visual híbrido.

4.2. Control Visual Basado en Imagen

Como se mencionó, esta técnica hace uso de la información que extrae el sistema de *visión por computador* del mundo físico, para ser empleados como entrada en su configuración de control. La ley de control se caracteriza por utilizar una *matriz de interacción* que representa la relación geométrica de la tarea a realizar.

Esta Sección desarrolla el IBVS considerando el enfoque *cámara en el extremo del robot y ver y mover dinámico* para el PTU-CAM, ya que es una configuración muy versátil para el propósito de regulación y seguimiento de un objeto.

Por lo general, los sistemas que utilizan IBVS, con la configuración antes mencionada, son multifrecuencia (ver Figura 4.5), es decir, la frecuencia de muestreo del controlador del robot es mayor que del IBVS, y cuya comunicación es de forma asíncrona [32]. Sin embargo, para no hacer más complejo la simulación y teniendo en cuenta el objetivo principal, todos los sistemas se discretizan a un sola frecuencia (mayor).

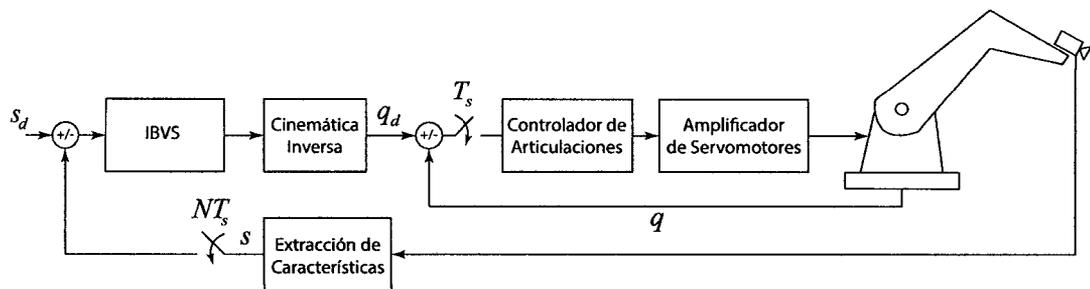


Figura 4.5: Sistema IBVS multifrecuencia.

4.2.1. Matriz de Interacción

La representación del movimiento de una cámara respecto de un punto fijo (ver Figura 4.6) está dado por

$${}^C\dot{r} = -\omega_C^{(C)} \times {}^C r - v_C^{(C)}, \quad (4.1)$$

donde ${}^C r = (X, Y, Z)^T$ es el vector posición del punto fijo respecto a la cámara $\{C\}$, y $v_C^{(C)} = (v_x, v_y, v_z)^T$ y $\omega_C^{(C)} = (\omega_x, \omega_y, \omega_z)^T$ son la velocidad lineal y angular de la cámara respectivamente, expresados en el sistema coordenado $\{C\}$. La demostración de (4.1) puede verse en el Apéndice A.4.2.

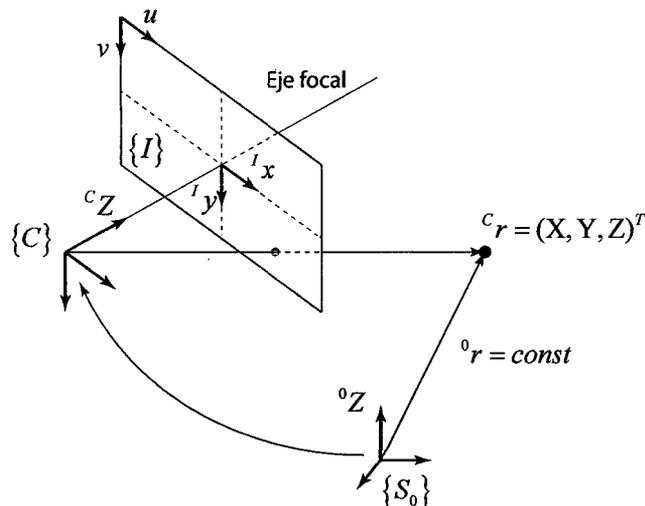


Figura 4.6: Movimiento de una cámara respecto de un punto fijo.

Desarrollando (4.1) se tiene

$$\begin{aligned} \dot{X} &= -v_x - \omega_y Z + \omega_z Y, \\ \dot{Y} &= -v_y - \omega_z X + \omega_x Z, \\ \dot{Z} &= -v_z - \omega_x Y + \omega_y X. \end{aligned} \quad (4.2)$$

Derivando la relación geométrica del punto fijo y su proyección en el plano de imagen, obtenida en (3.1), se tiene

$$\begin{aligned} \dot{x} &= (\dot{X} - x\dot{Z})/Z, \\ \dot{y} &= (\dot{Y} - y\dot{Z})/Z. \end{aligned} \quad (4.3)$$

De (4.2) y (4.3) se puede encontrar una relación entre la variación en el tiempo de la proyección del punto fijo en el plano de la imagen $(x, y)^T$, y la velocidad (lineal y angular) de la cámara respecto del punto fijo expresado en $\{C\}$, representado por $V_C^{(C)} = (v_C^{(C)}, \omega_C^{(C)})^T$, como

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -\frac{f}{Z} & 0 & \frac{x}{Z} & \frac{xy}{Z} & -\frac{f^2+x^2}{f} & y \\ 0 & -\frac{f}{Z} & \frac{y}{Z} & \frac{f^2+y^2}{f} & -\frac{xy}{Z} & -x \end{pmatrix} V_C^{(C)}, \quad (4.4)$$

donde la matriz de orden 2×6 representa la *matriz de interacción* relacionado a la proyección del punto fijo en el plano de la imagen, f es la distancia focal de la cámara, y $Z := {}^C Z$ representa la distancia del punto fijo respecto de la cámara.

Asimismo, de (4.4) y (3.2) se puede encontrar una relación entre la variación en el tiempo de las características visuales del punto fijo $s = (u, v)^T$ y la velocidad (lineal y angular) del cámara respecto del punto fijo y expresado en $\{C\}$, como

$$\dot{s} = L_s V_C^{(C)}, \quad (4.5)$$

donde L_s es la *matriz de interacción* relacionada a las características visuales del punto fijo y cuya forma es

$$L_s = \begin{pmatrix} -\frac{f_x}{Z} & 0 & \frac{u-c_u}{Z} & \frac{(u-c_u)(v-c_v)}{f_y} & -\frac{f_x^2+(u-c_u)^2}{f_x} & \frac{f_x(v-c_v)}{f_y} \\ 0 & -\frac{f_y}{Z} & \frac{v-c_v}{Z} & \frac{f_y^2+(v-c_v)^2}{f_y} & -\frac{(u-c_u)(v-c_v)}{f_x} & -\frac{f_y(u-c_u)}{f_x} \end{pmatrix}, \quad (4.6)$$

donde f_x, f_y, c_u y c_v son los *parámetros intrínsecos* de la cámara (ver Tabla 3.1).

4.2.2. Ley de Control

El principal objetivo en los *controles servovisuales*, es reducir al mínimo el error de forma

$$e(t) = s(t) - s_d, \quad (4.7)$$

donde $s(t)$ es el vector de características de un objeto observado en un instante de tiempo, y s_d es el vector de características deseadas. Para encontrar una relación entre $e(t) := e$ con $V_C^{(C)}$ se deriva (4.7) considerando (4.5), obteniéndose

$$\dot{e} = L_e V_C^{(C)}, \quad (4.8)$$

donde $L_e := L_s$ representa la *matriz de interacción* relacionado al error.

Considerando $V_C^{(C)}$ como entrada al controlador del robot y la tarea del IBVS como una regulación, asegurando una disminución exponencial desacoplado del error ($\dot{e} = -\lambda e$, $\lambda > 0$), entonces la **ley de control del IBVS** se define por

$$V_C^{(C)} = -\lambda L_e^+ e, \quad (4.9)$$

donde L_e^+ representa la pseudoinversa de L_e y puede encontrarse usando (2.10) o (2.11) dependiendo de su rango.

En la práctica, es casi imposible saber perfectamente los valores de L_e o L_e^+ , así que una aproximación o estimación de una de estas dos matrices debe ser realizado [30]. Por lo general, en el proceso de *calibración de cámara* se estiman varios parámetros necesarios para construir L_e , representado por \widehat{L}_e , para luego estimar su inversa usando técnicas numéricas clásicas.

Para evaluar la estabilidad de los sistemas *servovisuales* se utiliza el análisis de *Lyapunov*. En particular, se considera la función candidata de *Lyapunov* definido por la norma del error cuadrático

$$V(t) = \frac{1}{2} \|e(t)\|^2 = \frac{1}{2} e^T e, \quad (4.10)$$

donde $e(t) := e$ ahora es considerado como la obtenida de reemplazar (4.9) en (4.8), es decir,

$$\dot{e} = -\lambda L_e L_e^+ e, \quad (4.11)$$

donde $L_e L_e^+ \neq 0$ para un caso extremo como la pseudoinversa izquierda (2.11). Entonces, derivando (4.10) y usando (4.11) se tiene

$$\dot{V}(t) = e^T \dot{e} = -e^T \lambda (L_e L_e^+) e, \quad (4.12)$$

donde $\dot{V}(t) < 0$ cuando $L_e L_e^+$ es definido positivo, asegurando la *estabilidad asintótica global*. En [30] se demuestra que el IBVS tiene una *estabilidad asintótica local* inclusive para el caso extremo antes descrito.

4.2.3. Compensación al Movimiento del Objeto

La técnica conocida como *seguimiento visual* o *visual tracking* se encarga de mejorar la trayectoria descrita por las características visuales en la ley de control del IBVS para objetos con movimiento desconocidos.

De (4.9) se tiene

$$\dot{e} = L_e V_C^{(C)} + \frac{\partial e}{\partial t}, \quad (4.13)$$

donde $\partial e / \partial t$ representa la variación en el tiempo de e debido al movimiento del objeto, generalmente desconocido.

Reemplazando (4.9) en (4.13) se obtiene la nueva ley de control

$$V_C^{(C)} = -\lambda \widehat{L}_e^+ e - \widehat{L}_e^+ \frac{\partial e}{\partial t}, \quad (4.14)$$

donde $\widehat{\partial e / \partial t}$ es la estimación de $\partial e / \partial t$ y $\widehat{L}_e^+ \widehat{\partial e / \partial t}$ se conoce como el compensador

al movimiento del objeto.

La estimación $\widehat{\partial e / \partial t}$ puede encontrarse por diferentes métodos para compensar el movimiento del objeto, por ejemplo, a través de un término integral [30], como

$$\left(\frac{\widehat{\partial e}}{\partial t} \right)_{(k+1)} = \left(\frac{\widehat{\partial e}}{\partial t} \right)_{(k)} + \mu e_{(k)}, \quad (4.15)$$

donde μ es la ganancia integral.

En [11] hace una revisión de métodos que consiste en la comparación de la velocidad observada en la imagen y la velocidad que se puede predecir de la ley de control; además de utilizar filtros (p. ej., filtro de kalman) para eliminar ruidos involucrados en la adquisición de información. También, en relación a lo último, en [33] realiza un método más elaborado incluyendo compensación a movimientos abruptos en el modelo de estimación.

4.2.4. Transformación de Velocidades

En (4.5), la velocidad de la cámara está expresado en su sistema coordenado $\{C\}$ y para expresarlo en el sistema coordenado adjunto al extremo del robot $\{S_n\}$, considerando que la cámara está acoplado (soldado) a éste, se puede usar

$$V_n^{(n)} := {}^0V_n^{(n)} = \begin{pmatrix} {}^nR_C & S({}^np_C){}^nR_C \\ 0_{3 \times 3} & {}^nR_C \end{pmatrix} V_C^{(C)}, \quad (4.16)$$

donde nR_C representa la matriz de rotación de la cámara $\{C\}$ respecto $\{S_n\}$ y $S({}^np_C)$ representa la *matriz antisimétrica* relacionado al vector posición np_C . La demostración puede verse en el Apéndice A.6.

Si $V_n^{(n)}$ quiere expresarse en el sistema coordenado de referencia ubicado en la base del robot $\{S_0\}$, entonces se puede usar

$$V_n := {}^0V_n = \begin{pmatrix} {}^0R_n & 0_{3 \times 3} \\ 0_{3 \times 3} & {}^0R_n \end{pmatrix} V_n^{(n)}, \quad (4.17)$$

donde 0R_n representa la matriz de rotación del sistema coordenado adjunto al extremo del robot respecto a su base.

Por otro lado, se puede relacionar la variación en el tiempo de las características visuales de un objeto (\dot{s}) con la variación en el tiempo de las variables asociados a las articulaciones del robot (\dot{q}), como

$$\dot{s} = J_s \dot{q}, \quad (4.18)$$

donde J_s es conocido como *jacobiano de imagen* [30] y puede calcularse por

$$J_s = L_s M^{-1} J, \quad (4.19)$$

donde J es el *jacobiano* del robot y M representa la matriz de transformación que relaciona V_n y $V_C^{(C)}$, obtenida de multiplicar (4.16) y (4.17), es decir,

$$V_n = M V_C^{(C)}. \quad (4.20)$$

4.3. Calibración Cámara–Robot

Esta calibración se encarga de estimar la relación mecánica que existe entre la cámara y el robot (nT_C). El procedimiento general es asumir una plantilla de calibración fija para que el robot realice diferentes posturas, permitiendo que la cámara enfoque a éste para adquirir información de sus características visuales y luego establecer un sistema de ecuaciones para estimar nT_C . En éste se distinguen dos formulaciones muy conocidas (ver referencia [34]): la *formulación* $AX = ZB$ (ver Figuras 4.7) y la *formulación* $AX = XB$ (ver Figura 4.8).

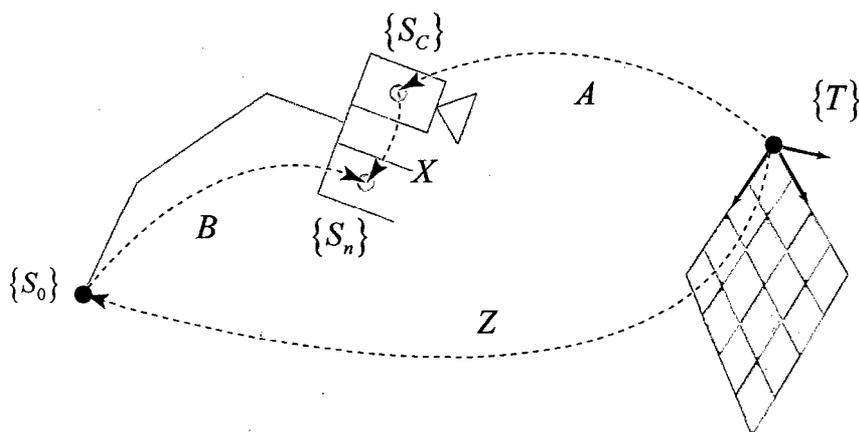


Figura 4.7: Formulación $AX = ZB$.

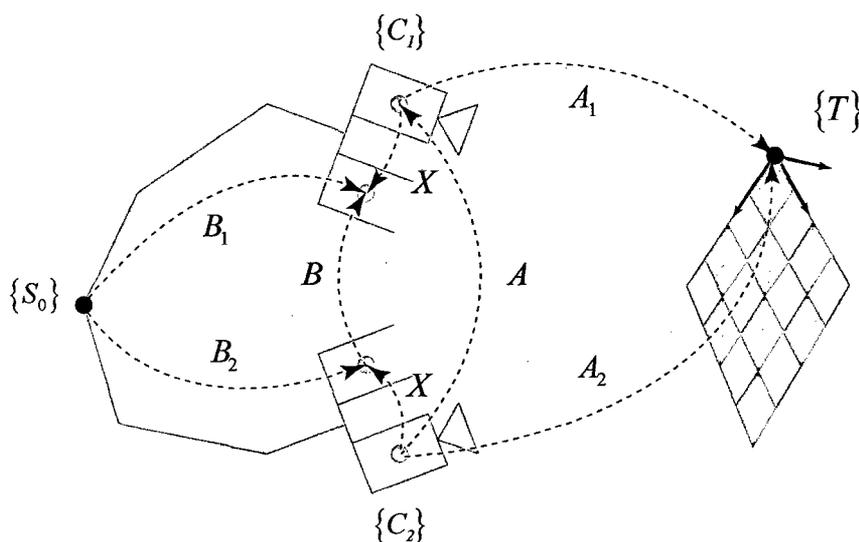


Figura 4.8: Formulación $AX = XB$.

- **Formulación $AX = ZB$** , ésta reproduce directamente las transformaciones rígidas en el sistema *cámara-robot*, donde muchas veces se asume Z como constante o conocido para estimar X , a partir de A y B . Generalmente la ecuación se descompone en sus partes de rotación y traslación, como

$$\begin{aligned} R_A R_X &= R_Z R_B, \\ R_A p_X + p_A &= R_Z p_B + p_Z. \end{aligned} \tag{4.21}$$

- **Formulación** $AX = XB$, ésta se crea a partir de dos movimientos del sistema *cámara-robot* para eliminar el termino Z y estimar X de forma directa, y con $A = A_2A_1^{-1}$ y $B = B_2^{-1}B_1$. Generalmente la ecuación se descompone en sus partes de rotación y traslación, como

$$\begin{aligned} R_A R_X &= R_X R_B, \\ R_A p_X + p_A &= R_X p_B + p_X. \end{aligned} \tag{4.22}$$

Los métodos para solucionar ambas formulaciones consiste en 3 formas: separable, simultanea e iterativo. La primera plantea la solución de la componente de orientación separado de la componente de posición, permitiendo una solución simple y rápida, sin embargo, los errores calculados a partir de la componente de orientación se traslada a la componente de posición. El segundo plantea la solución de las dos componentes simultáneamente, cuyos resultados son variables dependiendo de la escala de la componente de posición. Por último, la tercera plantea la solución de las componentes de forma iterativa, utilizando técnicas de optimización y cuyos resultados son más precisos que las anteriores, sin embargo, la solución suele ser compleja y depende en general de criterios de partidas [35].

En [36] describe ampliamente la solución para la formulación $AX = ZB$, donde utiliza la forma *separable* con la representación de *cuaterniones* para la matriz de rotación. En [37] describe la solución para la formulación $AX = XB$, utilizando técnicas de optimizan para la forma *simultanea*, en [38] utiliza *cuaterniones duales* para la misma forma, y en [39] realiza una solución simple de forma *separada* utilizando la representación de *vector de giro unitario* para la matriz de rotación. Por último, en [34] realiza un revisión de las dos formulaciones utilizando diversos métodos de optimización.

Se elige la formulación $AX = ZB$ usando el *producto de Kronecker*, en referencia a [40], donde se encuentra buenos resultados. En éste se define lo siguiente.

Definition 1 El operador apila una matriz $n \times m$ en un vector $nm \times 1$. El apilamiento de la matriz A de $n \times m$, denotado como $\text{vec}(A)$, es el vector formado por el apilamiento de las columnas de A en un vector $nm \times 1$

$$\text{vec}(A) = [a_{11}, a_{12}, \dots, a_{1n}, \dots, a_{mn}]^T, \quad (4.23)$$

Definition 2 El producto de Kronecker es un operador de matriz binaria que se asigna dos matrices dimensionadas arbitrariamente en una matriz más grande, con estructura de bloque especial. Dada la matriz A de $n \times m$ y la matriz B de $p \times q$, el producto Kronecker, denotado como $A \otimes B$, es la matriz $np \times mq$ con la estructura de bloques

$$A \otimes B = [a_{ij}B] = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{1m}B & \cdots & a_{mn}B \end{pmatrix}, \quad (4.24)$$

Usando el operador de apilación en las ecuaciones de (4.21) se tiene

$$\begin{pmatrix} R_A \otimes I_3 & -I_3 \otimes R_B^T \end{pmatrix} \begin{pmatrix} \text{vec}(R_X) \\ \text{vec}(R_Z) \end{pmatrix} = 0, \quad (4.25)$$

$$\begin{pmatrix} I_3 \otimes t_B^T & -R_A & I_3 \end{pmatrix} \begin{pmatrix} \text{vec}(R_X) \\ t_X \\ t_Z \end{pmatrix} = t_A, \quad (4.26)$$

donde I_3 es la matriz identidad de orden 3×3 . También, simplificando (4.25) y (4.26) con $\text{vec}(R_X) = x$ y $\text{vec}(R_Z) = z$ se tiene

$$\begin{pmatrix} R_A \otimes I_3 & -I_3 \otimes R_B^T & 0 & 0 \\ 0 & I_3 \otimes t_B^T & -R_A & I_3 \end{pmatrix} \begin{pmatrix} x \\ z \\ t_X \\ t_Z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ t_A \end{pmatrix}. \quad (4.27)$$

La solución de x , z , t_X y t_Z se puede determinar mediante la técnica de mínimos cuadrados. La parte R_X y R_Z , construido por x y z , puede no ser una matriz ortogonal debido al ruido, y la fórmula de rotación de *Rodrigues'* puede ser utilizada para hacer ortogonal a R_X y R_Z .

PTU-CAM

Como antes se explicó, la cámara DS325 tiene sus parámetros *intrínsecos* y de *distorsión* optimizados para sus dos cámaras (color y de profundidad). En esta tesis se utiliza el marco de profundidad (ver Tabla 4.1) para la calibración *cámara-robot*, debido a que se encuentra el mapa de color sincronizado. Si se utilizara el marco de color se debería realizar más operaciones. Por tanto, solo queda encontrar la matriz de acoplamiento entre la cámara DS325 (marco de profundidad) y el último eslabón del PTU, denotado como ${}^G T_C$ (G: gripper, C: cámara).

Tabla 4.1: Parámetros de cámara DS325 (marco de profundidad)

Tipo	Símbolo	Valor
Resolución	(w, h)	(320, 240)
Intrínseco	(f_x, f_y)	(224.501999, 230.494003)
Intrínseco	(c_u, c_v)	(160, 120)
Distorsión radial	(k_1, k_2, k_3)	(-0.170103, 0.144064, -0.047699)
Distorsión tangencial	(p_1, p_2)	(0, 0)

La Figura 4.9 muestra la plantilla utilizada para la calibración. Ésta contiene tres círculos de color rojo, los cuales representen a los vértices de los vectores ortogonales del plano X-Y de la plantilla. La posición 3D de estos círculos se

obtiene a partir de los algoritmos de procesamiento de imagen (ver Sección 3.4) y del mapa de profundidad de la cámara DS325. Para eliminar algún ruido en la adquisición digital de la plantilla se utiliza el filtro *Gaussian Blur*.

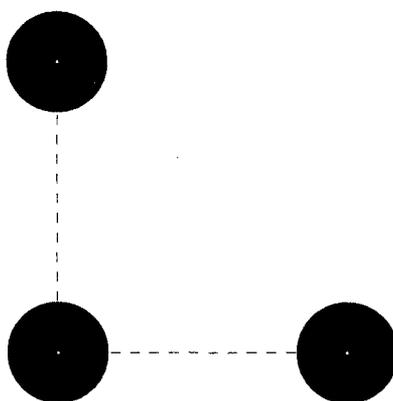


Figura 4.9: Plantilla de calibración ($AX = ZB$).

Como datos de entrada, para el método de calibración $AX = ZB$, se tiene a las matrices $A^{-1} = {}^C T_W$ y $B = {}^B T_G$, los cuales son las matrices de transformación homogénea de la plantilla (W: world) respecto a la cámara y del último eslabón respecto a la base (B: base) del robot respectivamente; y los datos de salida son $X^{-1} = {}^G T_C$ y $Z = {}^W T_B$, los cuales son las matrices de transformación homogénea de la cámara respecto al último eslabón y de la base del robot respecto a la plantilla.

La Figura 4.10 muestra la disposición del entorno de calibración. Cincuenta datos de entrada (posturas del módulo) son procesados utilizando el algoritmo de *Levenberg-Marquardt*, la transformación de *Rodrigues'* y los respectivos límites para cada variable incógnita; como resultado se obtuvo 22 iteraciones y una norma residual de 0.3308, estos se puede observar en la Tabla 4.2 y su simulación en la Figura 4.11.

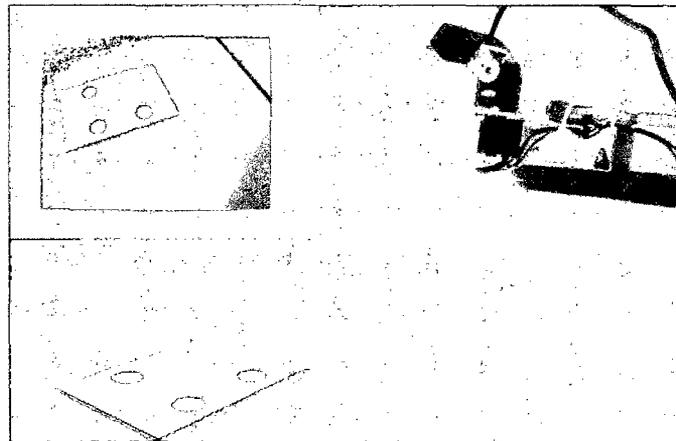


Figura 4.10: Disposición del entorno de calibración ($AX = ZB$).

Tabla 4.2: Parámetros de cámara-robot del PTU-CAM

Parámetro	Valor
2R_C	$\begin{pmatrix} 0.065589 & -0.997812 & 0.008337 \\ 0.995981 & 0.064953 & -0.061665 \\ 0.060988 & 0.012348 & 0.998062 \end{pmatrix}$
2p_C	$\begin{pmatrix} 0.019000 \\ 0.020585 \\ -0.192641 \end{pmatrix}$

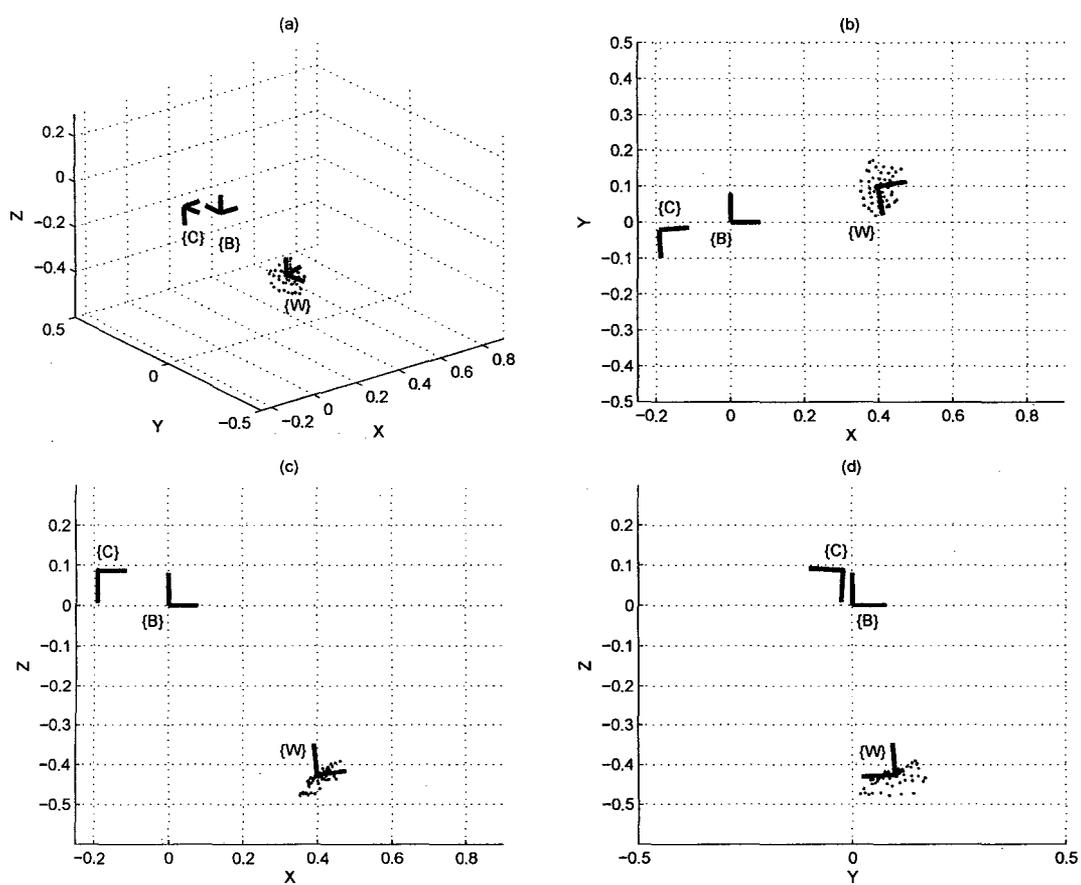


Figura 4.11: Simulación de la Calibración $AX = ZB$ en el PTU-CAM.

CAPÍTULO 5

RESULTADOS

Este Capítulo se compone de dos secciones. El primero muestra las gráficas de las simulaciones del IBVS y su comparación con el *control proporcional*, para un objeto *fijo* y otro con *movimiento pendular variable*. La segunda muestra las gráficas de la implementación, donde se explica el software respectivo para el IBVS, la comparación entre el modelo real y de simulación, y su respectiva comparación con el *control proporcional*.

5.1. Simulación

La simulación del IBVS se realiza en MatLab y Simulink, construyendo el modelo de cámara pinhole, la función objeto 3D (fijo o dinámico) y la estimación del mismo en 2D. Para observar el rendimiento del control se construyen dos gráficos 3D en función del tiempo que describen a un objeto *fijo* y otro con *movimiento pendular variable* (MPV). Este último simula el movimiento de un ají en su arbusto producido por el aire de la zona, y se realiza el mismo procedimiento comparando con el *control proporcional*.

Antes de realizar la simulación se determina el *tiempo de muestreo* para discretizar el modelo continuo del PTU-CAM. Éste debe estar dentro de un rango $T_{s_{min}} \leq T_s \leq T_{s_{max}}$.

La Figura 2.10 muestra el tiempo de establecimiento ($T_f = 0.3$ s) del modelo

de motor DC del AX-12 en respuesta a una entrada escalón (12 V), de aquí se establece el *tiempo de muestreo máximo* $T_{s_{max}} = 150$ ms. Éste es el doble de su frecuencia de establecimiento que evita el *Aliasing*¹.

El *tiempo de muestreo mínimo* depende de la velocidad de transmisión (Baud Rate) de los paquetes de instrucción entre el computador y los AX-12, además de los retardos (latencia) de los dispositivos y Software. Cada AX-12 maneja instrucciones de lectura, escritura y estado para la posición angular (ver Figura 2.2), cuya velocidad de transmisión se establece en el dispositivo (dato: 8 bit, parada: 1 bit), ya que el dispositivo USB2Dynamixel se maneja a través de un *COM Virtual*². Por tanto, el *tiempo de muestreo mínimo* se puede encontrar con

$$T_{s_{min}} \approx (N_a + 1) \left(\frac{\text{número de bits}}{\text{baud rate}} + \text{latencia} \right), \quad (5.1)$$

donde N_a es el número de AX-12 y el adicional es para resincronización. Finalmente, con 23 byte de datos de escritura, lectura y estado, 1 Mbps de velocidad de transmisión y un retardo de 0.5 ms por paquete, se obtiene $T_{s_{min}} = 5$ ms.

Estableciendo a 30 fps la velocidad de adquisición de datos y 320×240 de resolución para la cámara de color y de profundidad del DS325, se obtiene un tiempo promedio de procesamiento y control de $T_s = 33$ ms (ver Figura 5.1); éste es elegido como el *tiempo de muestreo* para la simulación, ya que parte del real y cumple el rango establecido.

La Figura 5.2 muestra la respuesta del modelo PTU-CAM discretizado (método de *Tustin*) a $T_s = 33$ ms y con $K_p = 80$ y $K_v = 0.8$. Estos valores proporcionales fueron sintonizados con base a la disertación [18] y afinados con la respuesta real.

Los errores medios entre la gráfica real y la discretizada (ver Figura 5.2) de

¹Efecto que provoca a las señales continuas se tornen indistinguibles cuando se muestrea por debajo del doble de su frecuencia.

²Configuración del computador en donde el puerto USB se ve como un puerto serie (COM/RS232) y cuya velocidad de transmisión es gestionado por este.

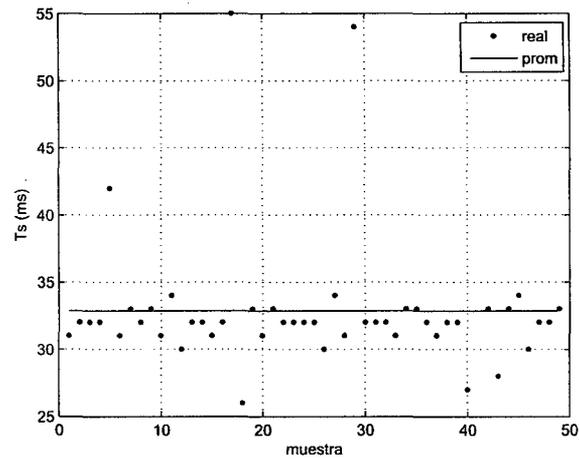


Figura 5.1: Tiempo promedio de procesamiento y control.

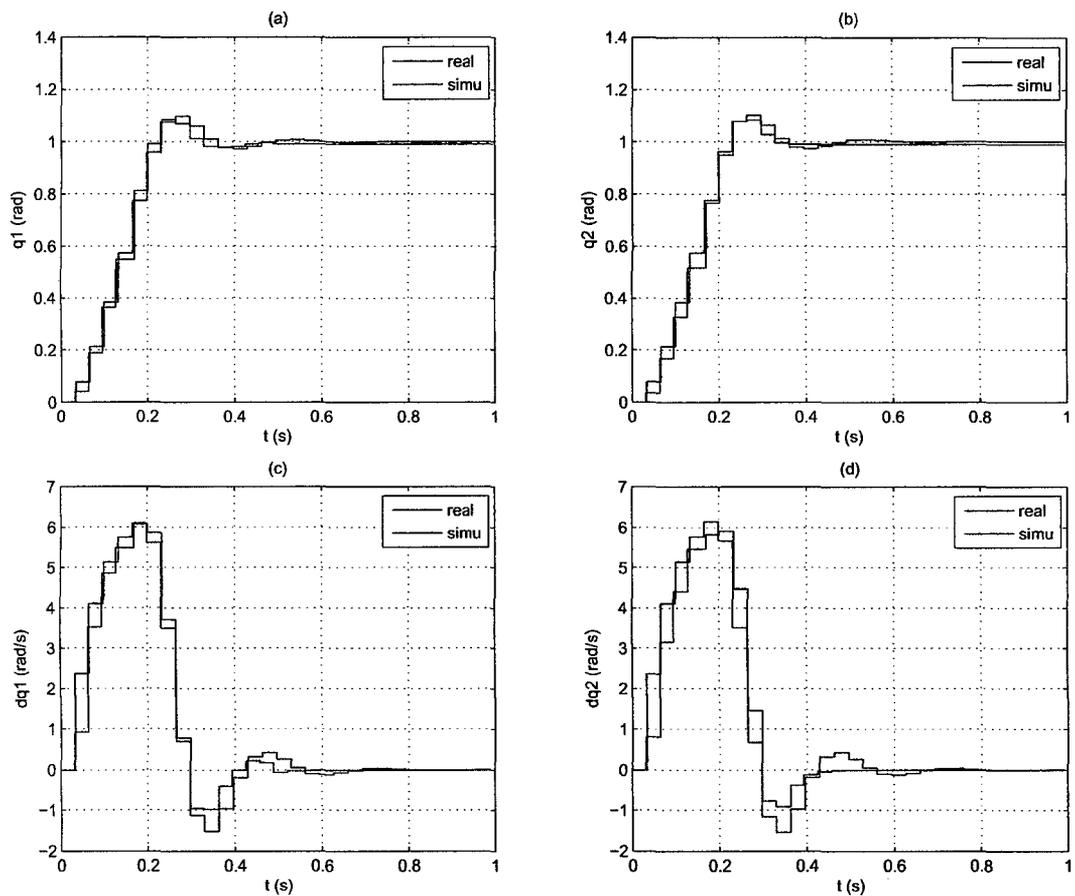


Figura 5.2: Respuesta real y de simulación al escalón del modelo PTU-CAM. (a) y (b) Posición angular de las articulaciones. (c) y (d) Velocidad angular de las articulaciones.

las posiciones angulares q_1 y q_2 son 1.0939° y 1.253° respectivamente. Además, la Figura 5.3 muestra algunas respuesta de variables internas del modelo, como la tensión y corriente en los actuadores de las articulaciones, y los torques que generan los eslabones.

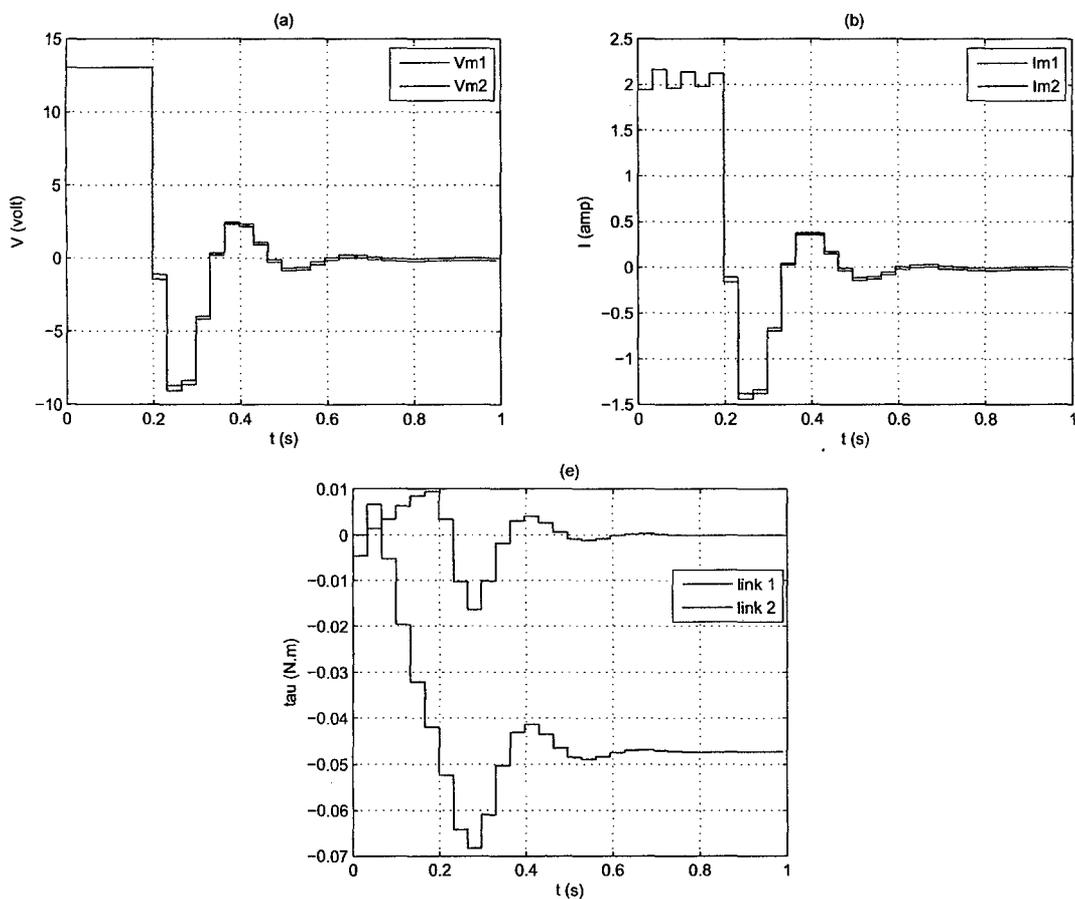


Figura 5.3: Respuesta de simulación al escalón del modelo PTU-CAM. (a) Tensión eléctrica en los actuadores de las articulaciones. (b) Corriente eléctrica en los actuadores de las articulaciones. (c) Torques generados por los eslabones.

Por último, se establece las características visuales deseadas $s_d = (160, 120)$ píxeles para la simulación e implementación, ya que coincide con el centro del Frame de la cámara y es fácil observarlo.

5.1.1. Respuesta del IBVS

Aquí se mostrará la respuesta del IBVS en simulación para un objeto *fijo* y otro con *movimiento pendular variable*, y demás variables del sistema.

La Figura 5.4 muestra el diagrama de bloques de todo el sistema (en *Simulink*), donde el bloque “IBVS” representa a ley de control encontrado en (4.9), el bloque “Transform. Vel.” representa a la transformaciones de velocidades encontrado en (4.20), el bloque “Jacobiano PTU-CAM” representa el Jacobiano encontrado en (2.9), el bloque “Integrator” representa al integrador discreto (método trapezoidal definido en el estado), el bloque “PTU-CAM” representa al modelo discretizado mostrado en la Figura 2.12, el bloque “CAM” representa al modelo de cámara sin distorsión encontrada en (3.3), el bloque “Constant” representa a las características visuales deseadas (s_d), y el bloque “Obj(t)” representa la función en el tiempo de la trayectoria que realiza el objeto.

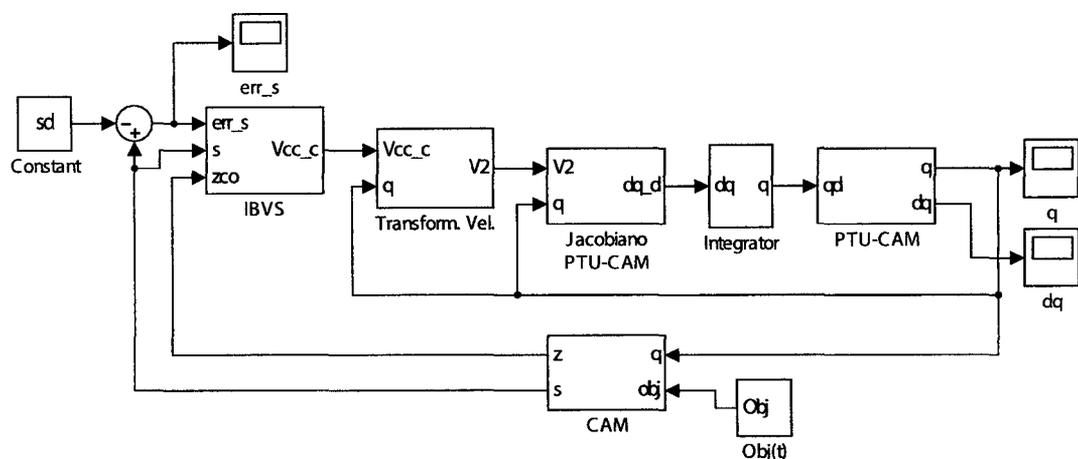


Figura 5.4: Diagrama de bloques del sistema con el IBVS.

La Figura 5.5 muestra varias gráficas de la simulación del sistema para un objeto fijo, ubicado en la posición (0.5,0.4,0.35) m y utilizando el IBVS con diferentes ganancias. La Figura 5.5a muestra la estimación del movimiento del objeto visto desde cámara hasta llegar al centro del Frame. La Figura 5.5b muestra la gráfica que describe la profundidad del objeto respecto de la cámara. Las

gráficas de las señales de control y las velocidades angulares pueden observarse en las Figuras 5.5c-f. Por último, para obtener una respuesta suavizada se elige la ganancia media $K = 10$ en las simulaciones del IBVS.

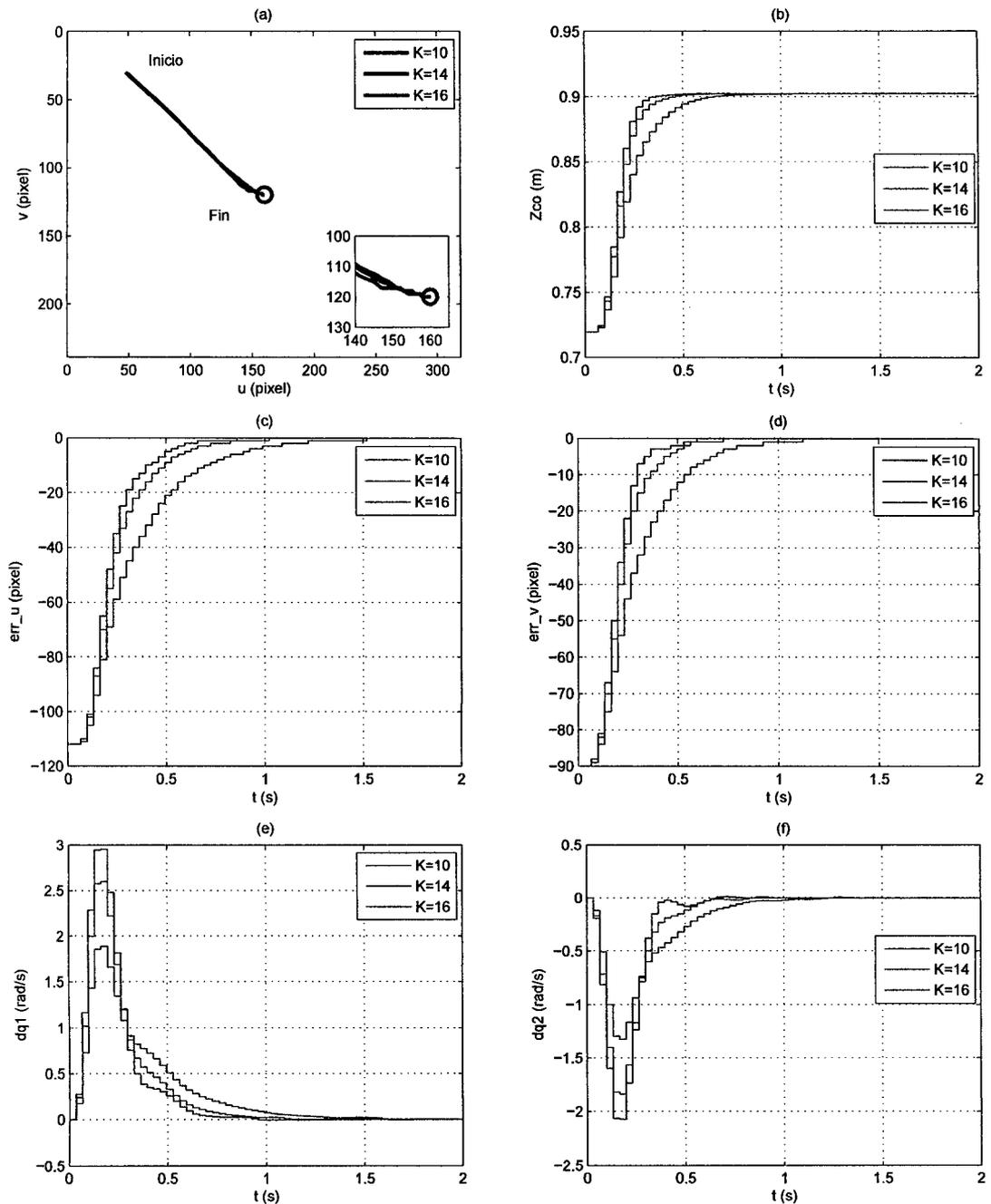


Figura 5.5: Simulación del sistema utilizando el IBVS para un objeto fijo y diferentes ganancias. (a) Frame de la cámara. (b) Profundidad del objeto respecto a la cámara. (c) y (d) Errores en las características visuales (señales de control). (e) y (f) Velocidades angulares de las articulaciones.

Simulación del IBVS (objeto fijo)

La Figura 5.6a muestra la simulación del sistema en el espacio de tareas, donde se encuentra el módulo, el objeto fijo (rojo) ubicado en $(0.5, 0.4, 0.35)$ m, y la estimación de la trayectoria (azul) de las características visuales (centro del Frame de la cámara) sobre el plano $X = 0.5$ (ver Figura 5.6c). La Figura 5.6b muestra el movimiento del objeto (rojo) visto desde la cámara hasta llegar al centro del Frame o características visuales deseadas (azul).

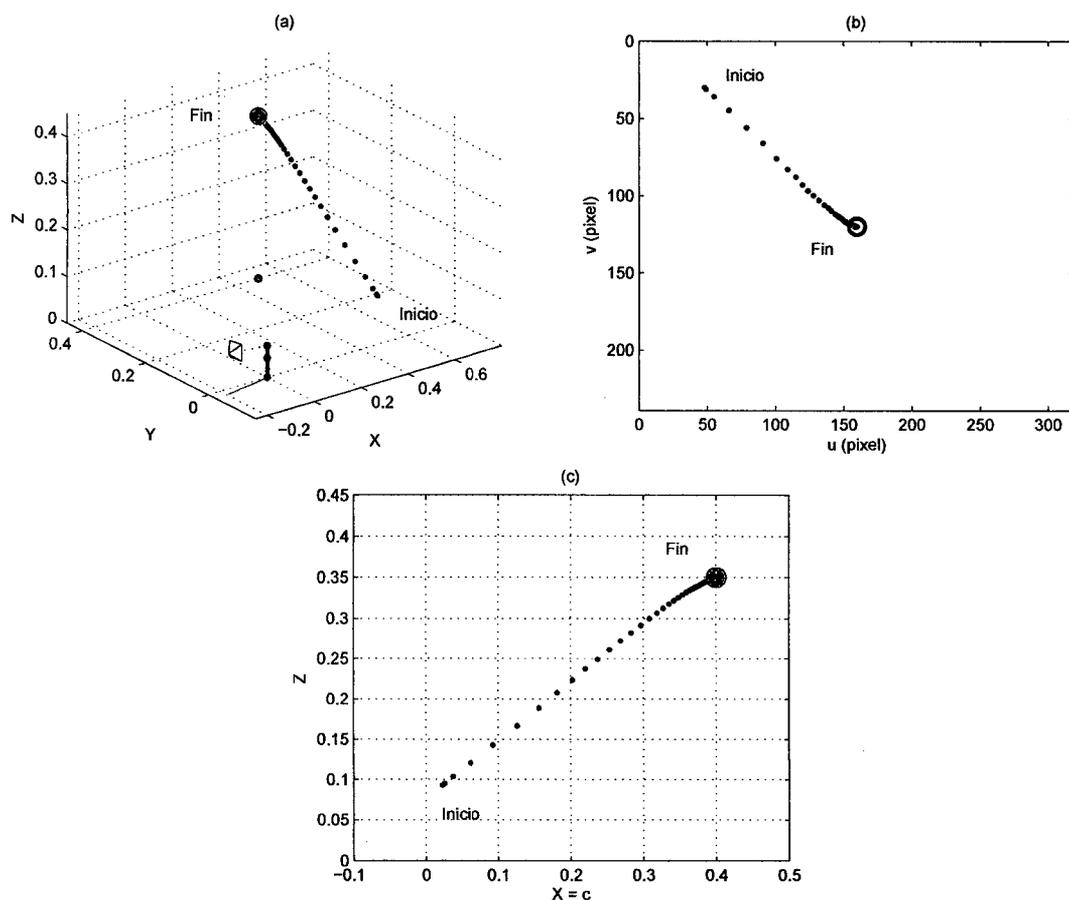


Figura 5.6: Simulación del sistema utilizando el IBVS ($K = 10$) para un objeto fijo. (a) Vista del espacio de tareas con el objeto (rojo) y la estimación de las características visuales (azul), sobre el plano $X = 0.5$. (b) Frame de la cámara. (c) Vista del plano $X = 0.5$.

La Figura 5.7a muestra las señales de control del IBVS para un objeto fijo, donde se observa el trabajo de regulación que tiene un comportamiento exponen-

cial decreciente, debido a su ley de control. La Figura 5.7b muestra la gráfica de la profundidad del objeto respecto de la cámara. La velocidad (lineal y angular) de la cámara respecto del objeto y expresado en el sistema coordenado de la cámara (señales de salida del IBVS) puede observarse en las Figuras 5.7c y d; donde se aprecia la nulidad de la velocidad angular de la cámara en su eje de profundidad, ya que ésta no rota en ese eje. Las velocidad (lineal y angular) del extremo del último eslabón del módulo respecto de la base del mismo puede observarse en las Figura 5.7e y f; donde se aprecia la poca velocidad angular en el eje X , debido a lo antes explicado.

Las Figuras 5.8a y b, muestran las comparaciones entre las señales deseadas (azul) y las respuestas actuales (rojo) de las velocidades angulares de las articulaciones, obtenida del *Jacobiano* del módulo. También, las Figuras 5.8c y d, muestran las correspondientes posiciones angulares de las articulaciones. En estas gráficas se observan un buen seguimiento de las respuestas actuales a las señales deseadas.

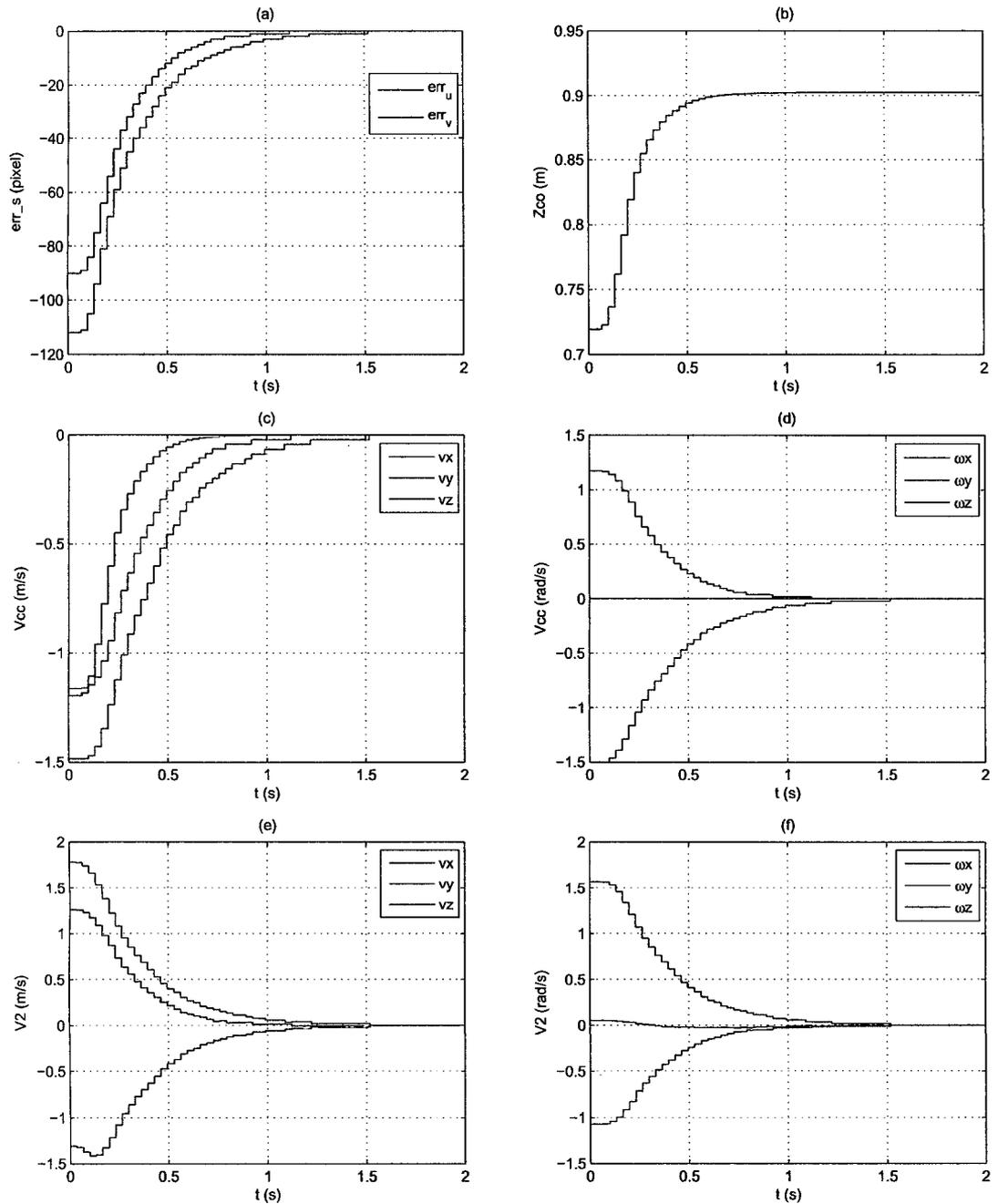


Figura 5.7: Respuesta simulada del IBVS ($K = 10$) para un objeto fijo. (a) Error de las características visuales. (b) Profundidad del objeto respecto a la cámara. (c) y (d) Velocidad de la cámara respecto del objeto y expresado en la cámara. (e) y (f) Velocidad absoluta del último eslabón (gripper) respecto de la base del módulo.

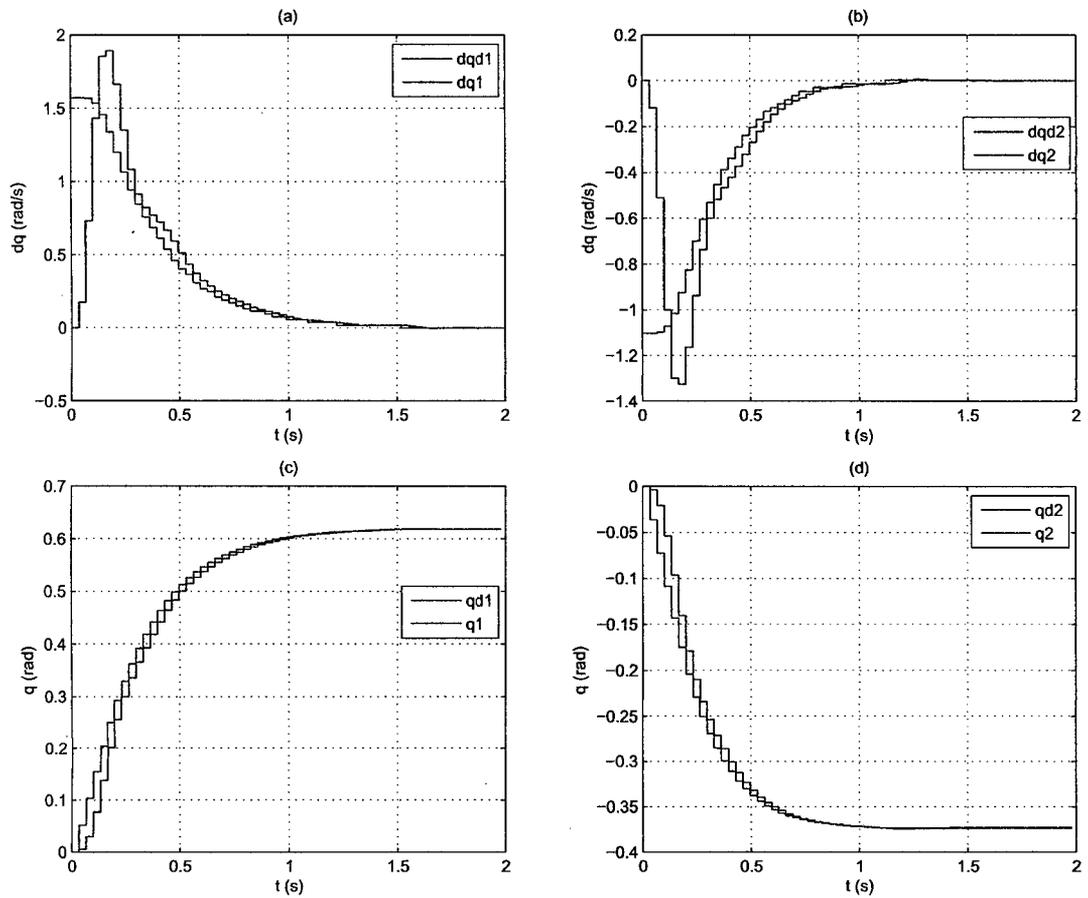


Figura 5.8: Respuesta simulada del IBVS ($K = 10$) para un objeto fijo. (a) y (b) Velocidad angular deseada (azul) y actual (rojo) de las articulaciones. (c) y (d) Posición angular deseada (azul) y actual (rojo) de las articulaciones.

Simulación del IBVS (objeto con MPV)

Una forma sencilla y formal de aproximar el movimiento complejo de un ají producido por el viento de la zona de cosecha y vibración por maquinaria pesada, en periodos pequeños, es utilizando una trayectoria pendular variable, debido a que el movimiento pendular es más notoria en sus ejes X y Y , y un movimiento casi lineal en su eje Z (ver Figura 5.9). La finalidad de implementar esta trayectoria es observar el comportamiento del IBVS a movimientos variables por parte del objeto; y en futuras aplicaciones establecer un rango de confianza, donde el ají oscile en pequeños grados para detener el movimiento del módulo y proceder con la cosecha automatizada.

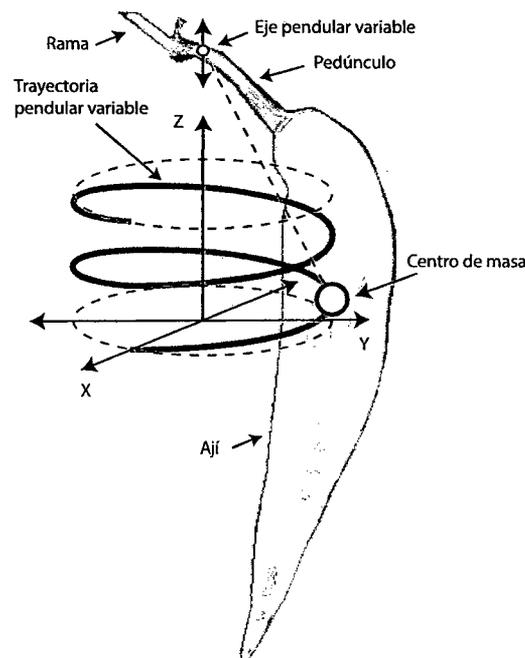


Figura 5.9: Trayectoria del centro de masa de un objeto (ají) con movimiento pendular variable.

La Figura 5.10a muestra la simulación del sistema en el espacio de tareas, donde se encuentra el módulo, el objeto con *movimiento pendular variable* (rojo) de radio 0.025 m y velocidad constante (proporcional al tiempo $t = 1$ s que tarda en dar 2 giros); y la estimación de la trayectoria (azul) de las características

visuales (centro del Frame de la cámara) sobre el plano $X = 0.5$ (ver Figura 5.10c). La Figura 5.10b muestra el movimiento del objeto (rojo) visto desde la cámara hasta llegar al centro del Frame o características deseadas (azul).

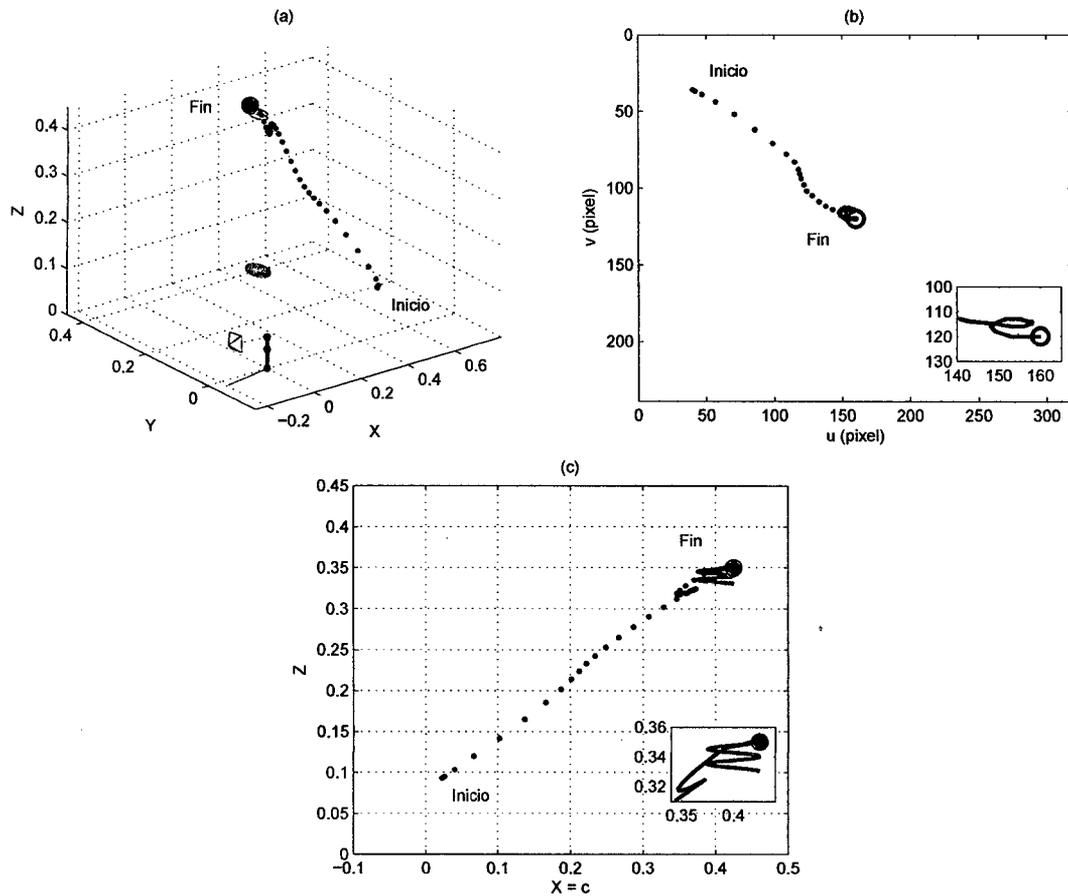


Figura 5.10: Simulación del sistema utilizando el IBVS ($K = 10$) para un objeto con movimiento pendular variable. (a) Vista del espacio de tareas con el objeto (rojo) y la estimación de las características visuales (azul), sobre el plano $X = 0.5$. (b) Frame de la cámara. (c) Vista del plano $X = 0.5$.

La Figura 5.11a muestra las señales de control del IBVS para un objeto con *movimiento pendular variable*, donde se observa el trabajo de regulación. La Figura 5.11b muestra la gráfica de la profundidad del objeto respecto de la cámara. La velocidad (lineal y angular) de la cámara respecto del objeto y expresado en el sistema coordenado de la cámara (señales de salida del IBVS) puede observarse en las Figura 5.11c y d; donde se aprecia la nulidad de la velocidad angular de la

cámara en su eje de profundidad, ya que ésta no rota en ese eje. Las velocidades (lineal y angular) del extremo del último eslabón del módulo respecto de la base del mismo puede observarse en las Figuras 5.11e y f; donde se aprecia la poca velocidad angular en el eje X , debido a lo antes explicado.

La Figura 5.12a y b, muestran las comparaciones entre las señales deseadas (azul) y las respuestas actuales (rojo) de las velocidades angulares de las articulaciones, obtenida del *Jacobiano* del módulo. También, las Figuras 5.12c y d, muestran las correspondientes posiciones angulares de las articulaciones. En estas gráficas se observan un buen seguimiento de las respuestas actuales a las señales deseadas.

En estas gráficas muestran que el tiempo de establecimiento por parte del IBVS es aproximadamente 0.5 s mayor al término del *movimiento pendular variable* del objeto, que demuestra un buen trabajo en la regulación de la tarea y no genera sobreimpulso.

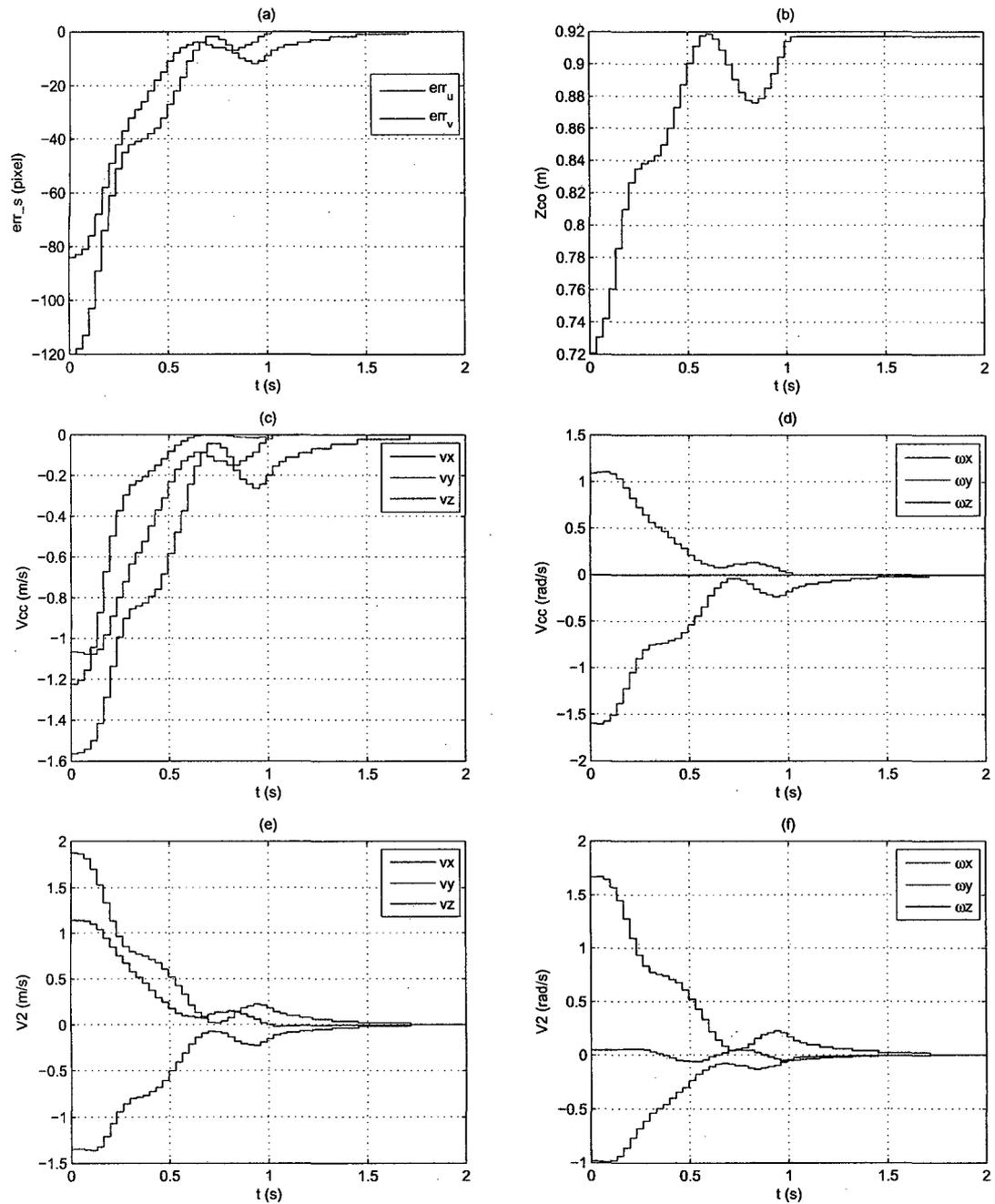


Figura 5.11: Simulación del sistema utilizando el IBVS ($K = 10$) para un objeto con movimiento pendular variable. (a) Error de las características visuales. (b) Profundidad del objeto respecto a la cámara. (c) y (d) Velocidad de la cámara respecto del objeto y expresado en la cámara. (e) y (f) Velocidad absoluta del último eslabón (gripper) respecto de la base del módulo.

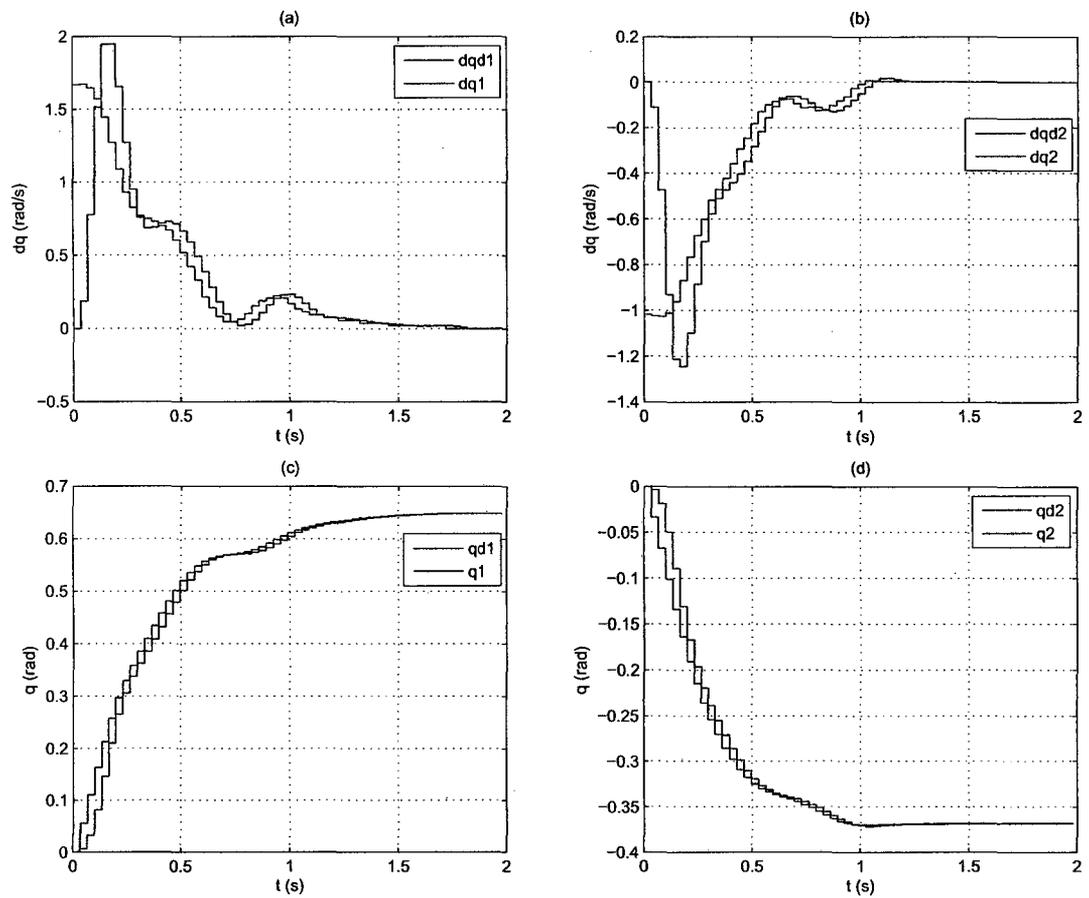


Figura 5.12: Simulación del sistema utilizando el IBVS ($K = 10$) para un objeto con movimiento pendular variable. (a) y (b) Velocidad angular deseada (azul) y actual (rojo) de las articulaciones. (c) y (d) Posición angular deseada (azul) y actual (rojo) de las articulaciones.

5.1.2. Comparación entre el IBVS y el control proporcional

Como antes se mencionó, la forma más sencilla de implementar un control visual en un PTU-CAM (2 GDL) es utilizando un *control proporcional*. Éste relaciona directamente las velocidades angulares deseadas de las articulaciones con las señales de control o errores de las características visuales (e_u, e_v).

La Figura 5.13 muestra el diagrama de bloques del *control proporcional* (en *Simulink*), donde “Ctrl P” representa a éste, el bloque “Integrator” representa al integrador discreto (método trapezoidal definido en el estado), el bloque “PTU-CAM” representa al modelo discretizado mostrado en la Figura 2.12, el bloque “CAM” representa al modelo de cámara sin distorsión encontrada en (3.3), el bloque “Constant” representa a las características visuales deseadas (s_d), y el bloque “Obj(t)” representa la función en el tiempo de la trayectoria que realiza el objeto.

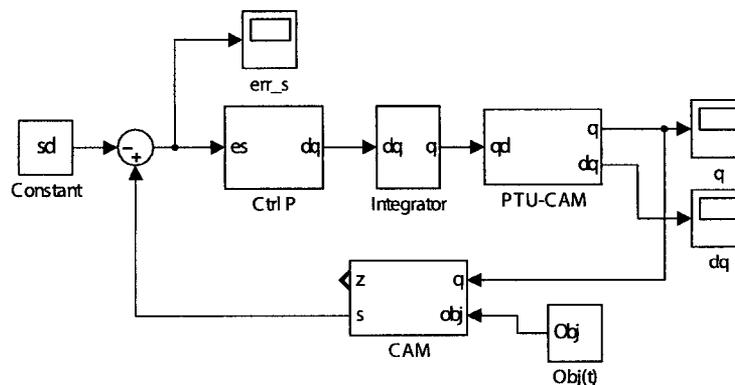


Figura 5.13: Diagrama de bloques del sistema con el control proporcional.

La Figura 5.14 muestra varias gráficas de la simulación del sistema en respuesta a la tarea de regulación para un objeto fijo, ubicado en la posición (0.5, 0.4, 0.35) m y utilizando el *control proporcional* con diferentes ganancias, que proporcionan los mismos tiempos de establecimiento que sus similares del IBVS.

La Figura 5.14a muestra la estimación del movimiento del objeto visto desde

cámara hasta llegar al centro del Frame. La Figura 5.14b muestra la gráfica que describe la profundidad del objeto respecto de la cámara. Las gráficas de las señales de control y las velocidades angulares pueden observarse en las Figuras 5.14c-f.

La Figura 5.15 muestra las gráficas de comparación de las características visuales (señales de control) entre ambos controles, utilizando sus correspondientes ganancias que definen el mismo tiempo de establecimiento.

En estas gráficas se observan que a mayor es la ganancia del *control proporcional* más tiende al sobreimpulso, a diferencia del IBVS que se mantiene en una respuesta suavizada para el mismo tiempo de establecimiento. Por tanto, para obtener una comparación justa con el IBVS ($K = 10$) se elige la ganancia $K = 0.0183$ en las simulaciones del *control proporcional*.

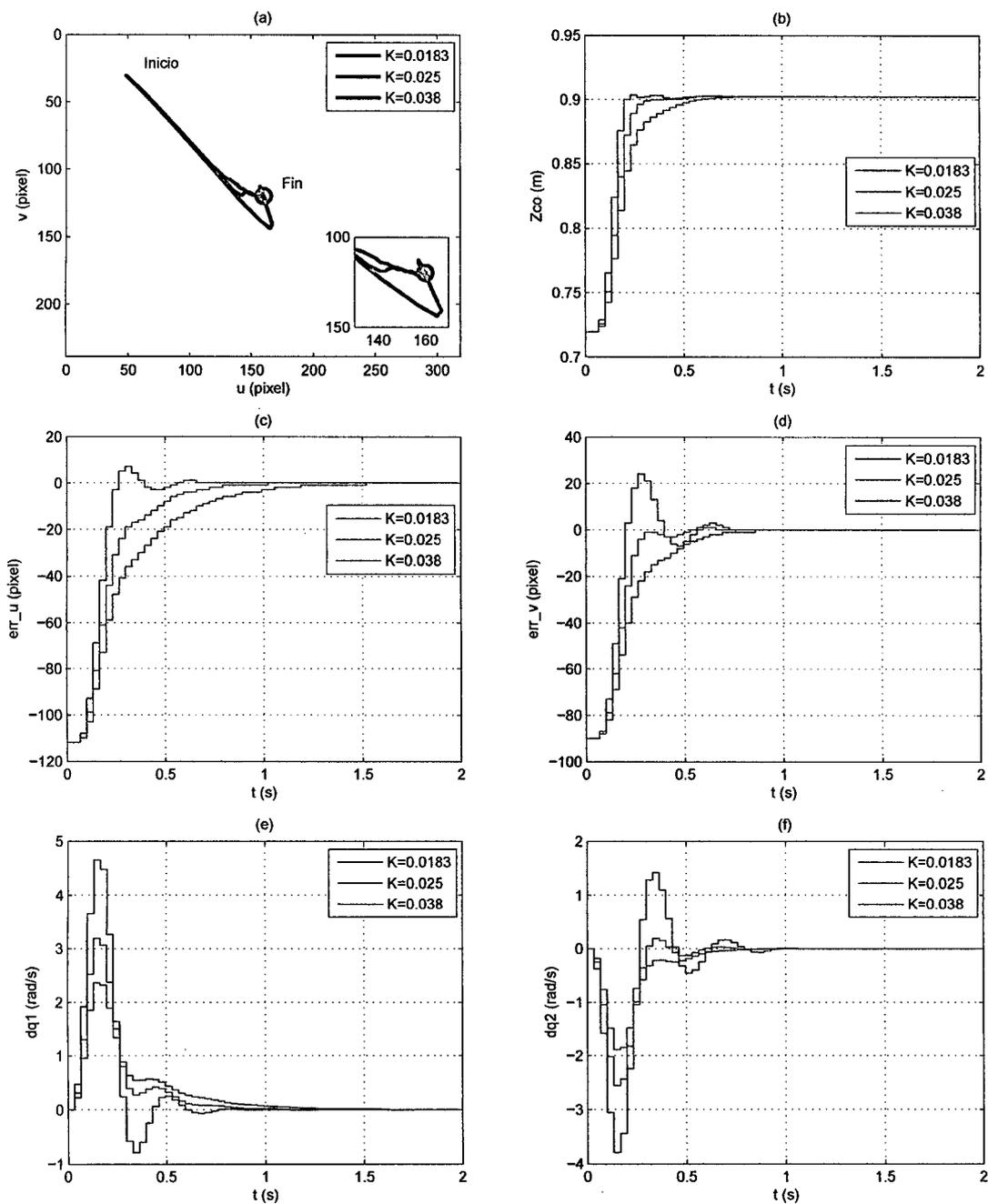


Figura 5.14: Simulación del sistema utilizando el control proporcional para un objeto fijo y diferentes ganancias, que proporcionan los mismos tiempos de establecimiento que sus similares del IBVS. (a) Frame de la cámara. (b) Profundidad del objeto respecto a la cámara. (c) y (d) Error de las características visuales (señales de control). (e) y (f) Velocidad angular de las articulaciones.

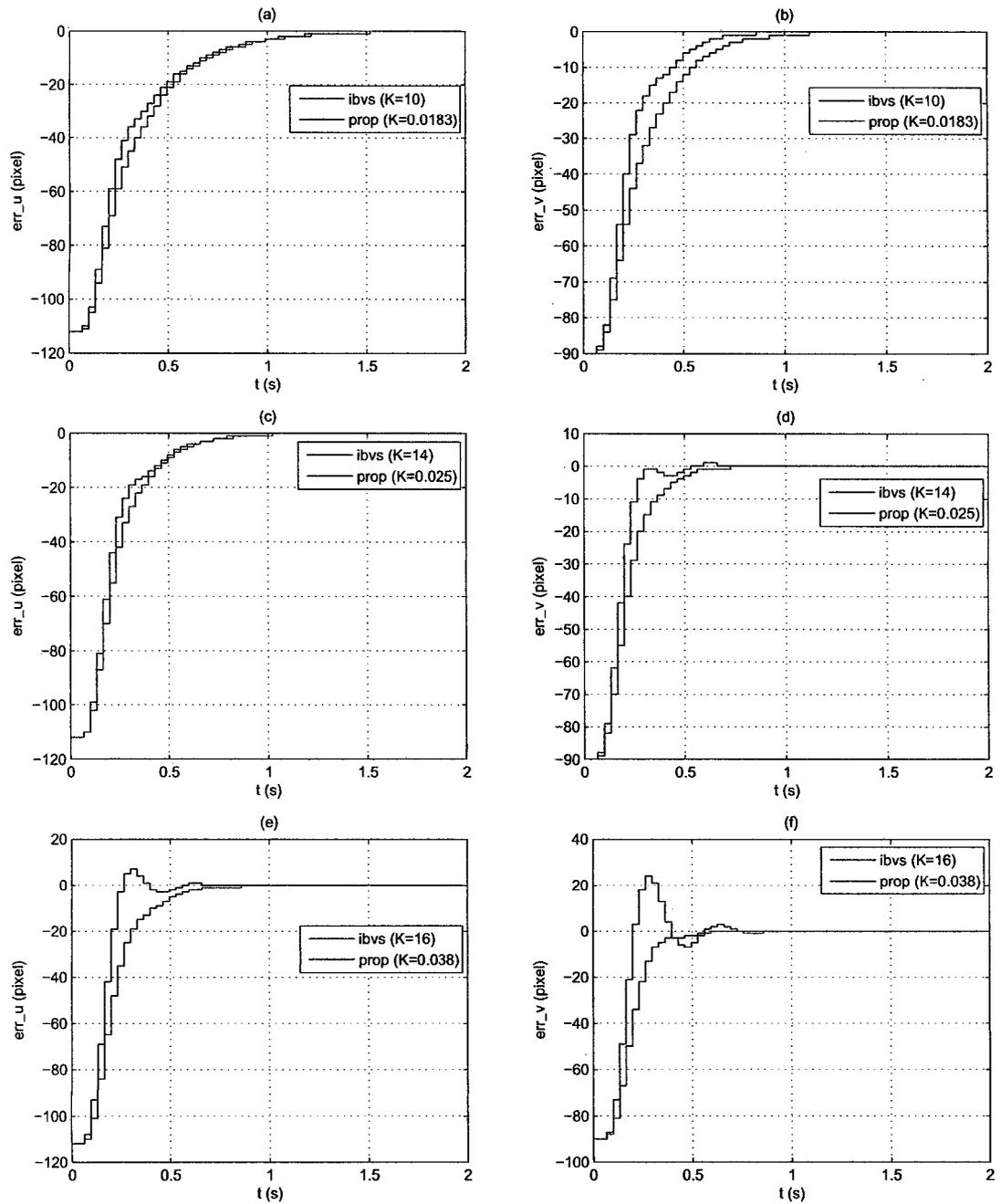


Figura 5.15: (a)–(f) Simulación de la comparación de los error de las características visuales (señales de control) entre el control proporcional y el IBVS para un objeto fijo y diferentes ganancias que proporcionan los mismos tiempos de establecimiento.

Simulación del IBVS y control proporcional (objeto fijo)

La Figura 5.16a muestra la simulación del sistema en el espacio de tareas, donde se encuentra el módulo, el objeto fijo (rojo) ubicado en $(0.5, 0.4, 0.35)$ m, y la estimación de las trayectorias de las características visuales (centro del Frame de la cámara) en el IBVS (azul) y el *control proporcional* (verde), sobre el plano $X = 0.5$ (ver Figura 5.16c). La Figura 5.16b muestra el movimiento del objeto visto desde la cámara, para el IBVS (rojo) y el control proporcional (cian), hasta llegar al centro del Frame o características visuales deseadas (azul).

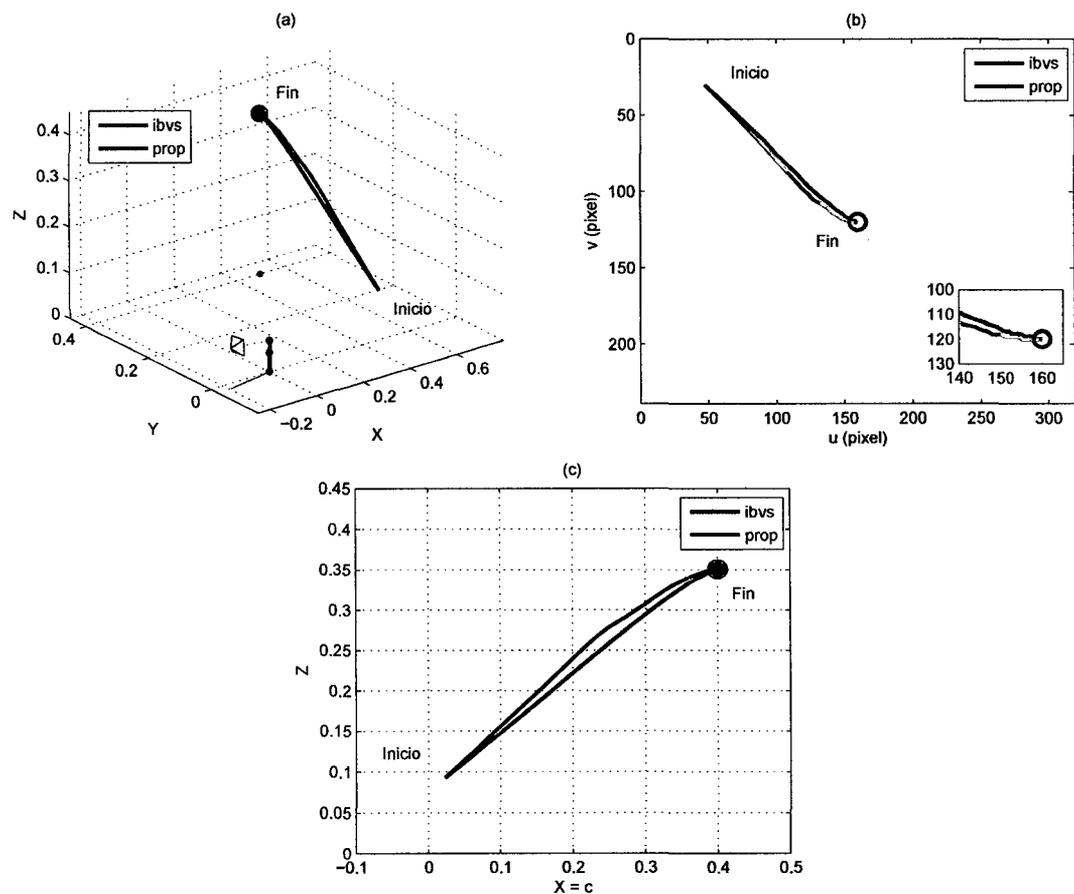


Figura 5.16: Simulación del sistema utilizando el IBVS ($K = 10$) y el control proporcional ($K = 0.0183$) para un objeto fijo. (a) Vista del espacio de tareas con el objeto (rojo) y la estimación de las trayectorias de las características visuales en el IBVS (azul) y el *control proporcional* (verde), sobre el plano $X = 0.5$. (b) Frame de la cámara. (c) Vista del plano $X = 0.5$.

Las Figuras 5.17a y b, muestran las señales de control del IBVS (azul) y

del *control proporcional* (rojo) para un objeto fijo, donde se observa el trabajo de regulación; y sus respectivas velocidades angulares, deseadas y actuales, de las articulaciones para ambos controles pueden verse en las Figuras 5.17c-f.

La Figura 5.18 muestra las posiciones angulares, deseadas y actuales, de las articulaciones para el IBVS (azul) y el *control proporcional* (rojo).

En estas gráficas simuladas de comparación muestran que el *control proporcional* tiene una respuesta similar al IBVS para un objeto fijo, pero con la tendencia a llegar al establecimiento lo más rápido y de manera brusca (movimiento), ya que no considera la cinemática del módulo. Sin embargo, éste control puede servir para pruebas rápidas sin complicaciones en el cálculo del modelos matemáticos. Por último, la Tabla 5.1 muestra los errores RMS de las características visuales y del movimiento en las articulaciones para ambos control.

Tabla 5.1: Error RMS de la simuladas del IBVS y del control proporcional para un objeto fijo.

Control	Objeto fijo	
	e_s (píxeles)	\dot{q} (rpm)
IBVS	33.2351	4.6432
Proporcional	31.0443	5.3241

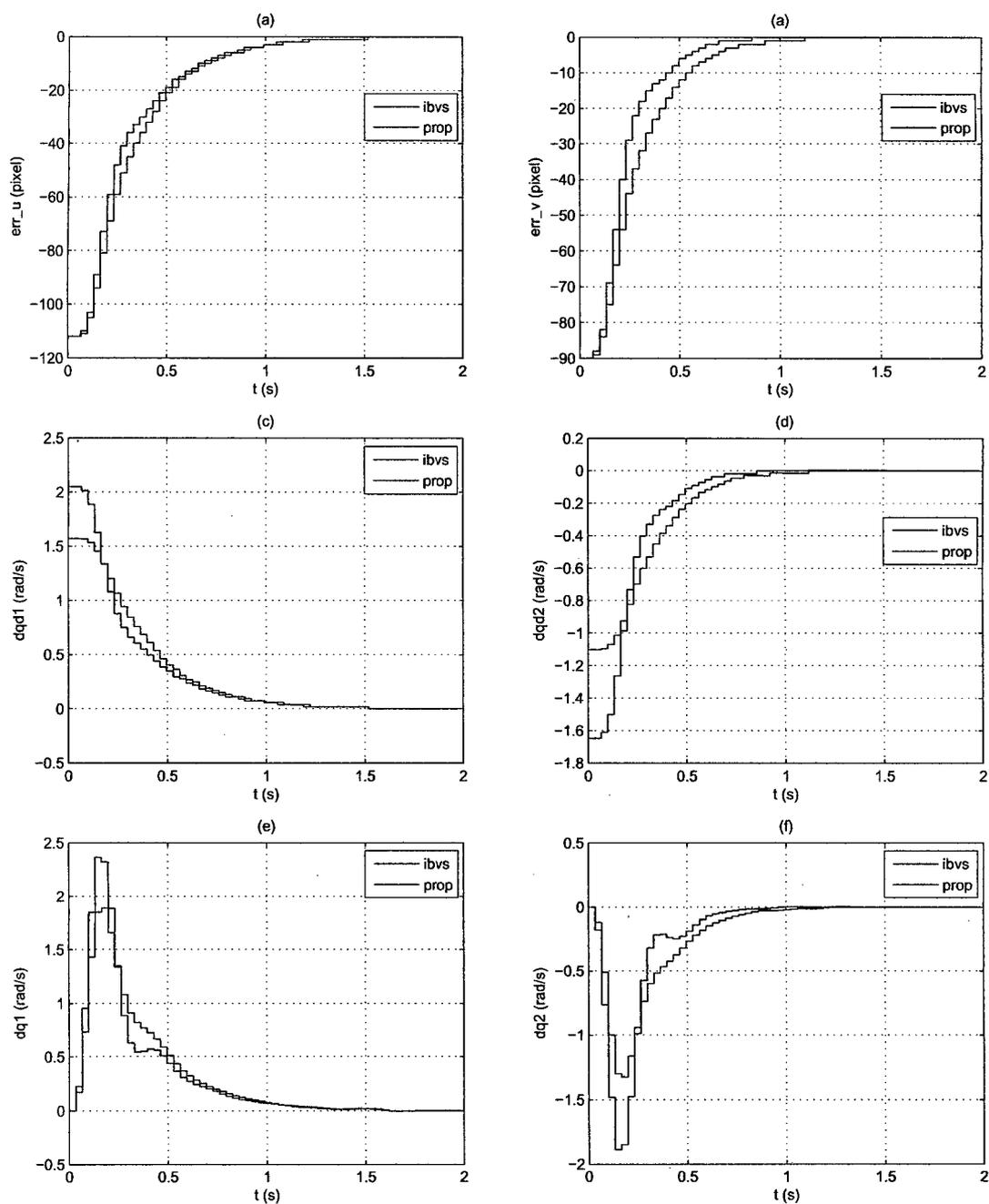


Figura 5.17: Respuesta simulada del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto fijo. (a) y (b) Error de las características visuales. (c) y (d) Velocidad angular deseada de las articulaciones. (e) y (f) Velocidad angular actual de las articulaciones.

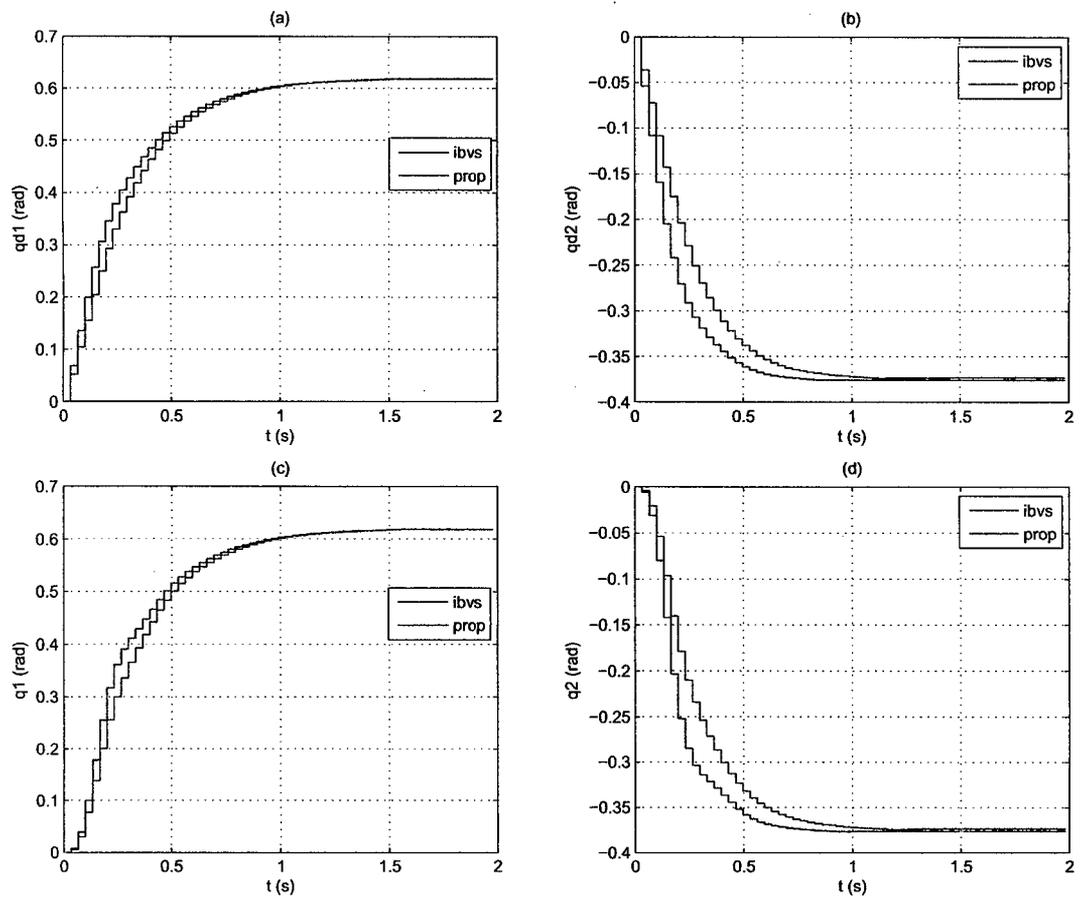


Figura 5.18: Respuesta simulada del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto fijo. (a) y (b) Posición angular deseada de las articulaciones. (c) y (d) Posición angular actual de las articulaciones.

Simulación del IBVS y control proporcional (objeto con MPV)

La Figura 5.19a muestra la simulación del sistema en el espacio de tareas, donde se encuentra el módulo, el objeto con *movimiento pendular variable* (rojo) de radio 0.025 m y velocidad constante (proporcional al tiempo $t = 1$ s que tarda en dar 2 giros), y la estimación de las trayectorias de las características visuales (centro del Frame de la cámara) en el IBVS (azul) y el *control proporcional* (verde), sobre el plano $X = 0.5$ (ver Figura 5.19c). La Figura 5.19b muestra el movimiento del objeto visto desde la cámara, para el IBVS (rojo) y el control proporcional (cian), hasta llegar al centro del Frame o características visuales deseadas (azul).

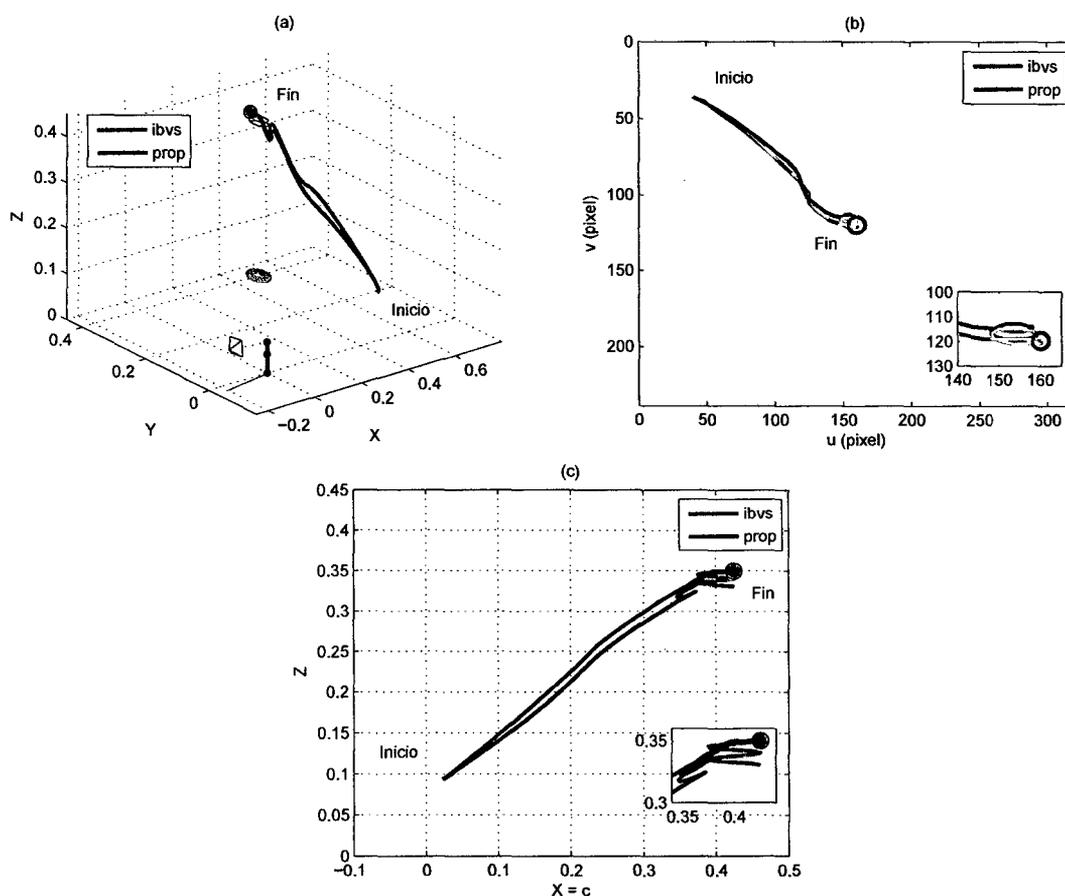


Figura 5.19: Respuesta simulada del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto con movimiento pendular variable. (a) Vista del espacio de tareas con el objeto (rojo) y la estimación de las trayectorias de las características visuales en el IBVS (azul) y el *control proporcional* (verde), sobre el plano $X = 0.5$. (b) Frame de la cámara. (c) Vista del plano $X = 0.5$.

Las Figuras 5.20a y b, muestran las señales de control del IBVS (azul) y del *control proporcional* (rojo) para un objeto con *movimiento pendular variable*, donde se observa el trabajo de regulación. Las velocidades angulares, deseadas y actuales, de las articulaciones para ambos controles pueden verse en las Figuras 5.20c-f.

La Figura 5.21 muestra las posiciones angulares, deseadas y actuales, de las articulaciones para el IBVS (azul) y el *control proporcional* (rojo), en la tarea de regulación de la posición en características visuales de un objeto con *movimiento pendular variable*.

En estas gráficas simuladas de comparación muestran que el *control proporcional* tiene un respuesta cercana al IBVS para un objeto con movimiento pendular variable, ya que se hace notoria la tendencia al sobreimpulso y como consecuencia un movimiento mayor en las articulaciones. Además, el tiempo de establecimiento para ambos controles es aproximadamente 0.5 s mayor al termino del movimiento del objeto en la tarea de regulación, desmostando que el IBVS tiene un mejor rendimiento en movimiento que el *control proporcional*. Por último, la Tabla 5.2 muestra los errores RMS de las características visuales y del movimiento en las articulaciones para ambos control.

Tabla 5.2: Error RMS de la simuladas del IBVS y del control proporcional para un objeto con movimiento pendular variable.

Control	Objeto con MPV	
	e_s (píxeles)	\dot{q} (rpm)
IBVS	32.4880	4.6234
Proporcional	30.5657	5.2165

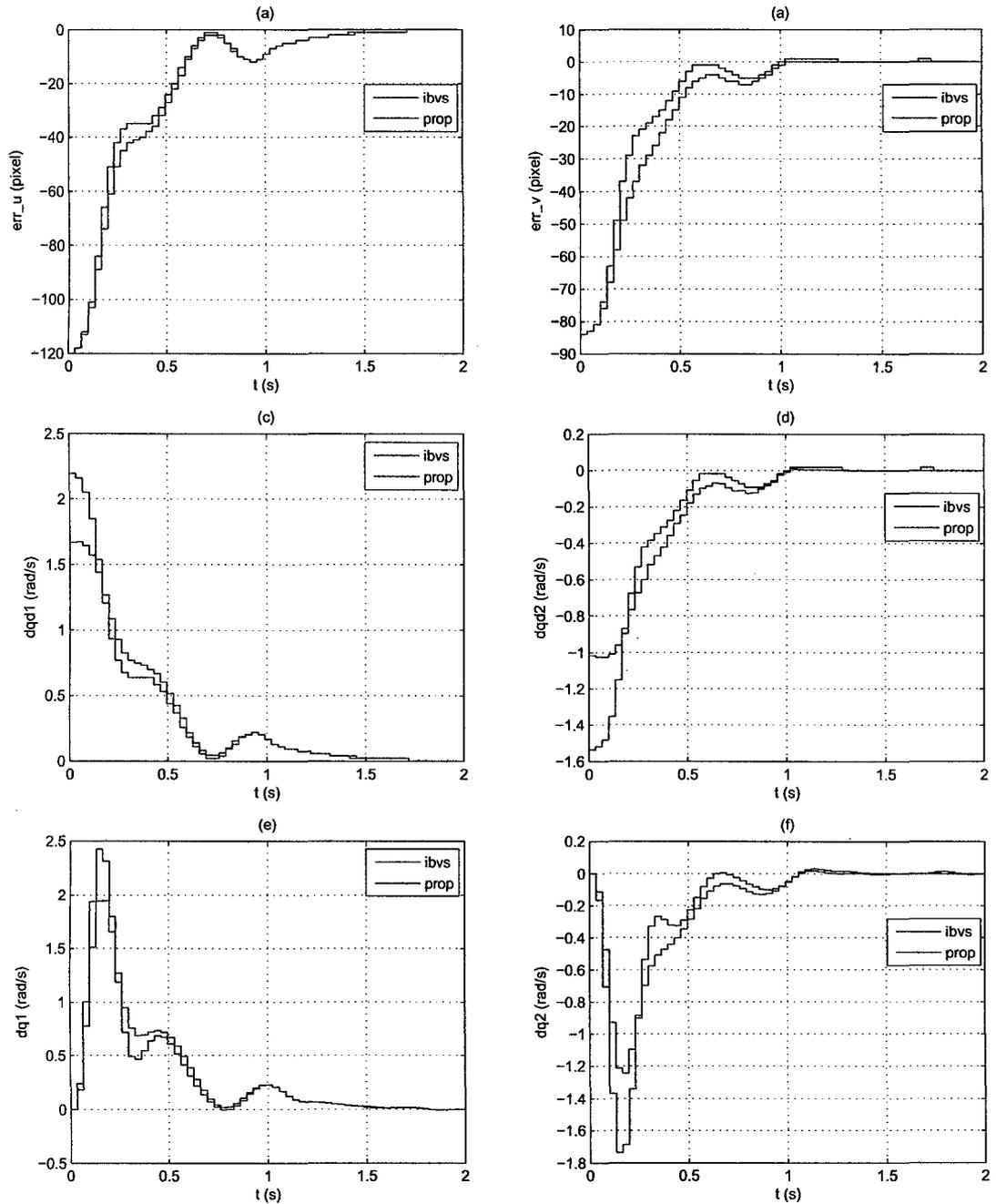


Figura 5.20: Respuesta simulada del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto con movimiento pendular variable. (a) y (b) Error de las características visuales. (c) y (d) Velocidad angular deseada de las articulaciones. (e) y (f) Velocidad angular actual de las articulaciones.

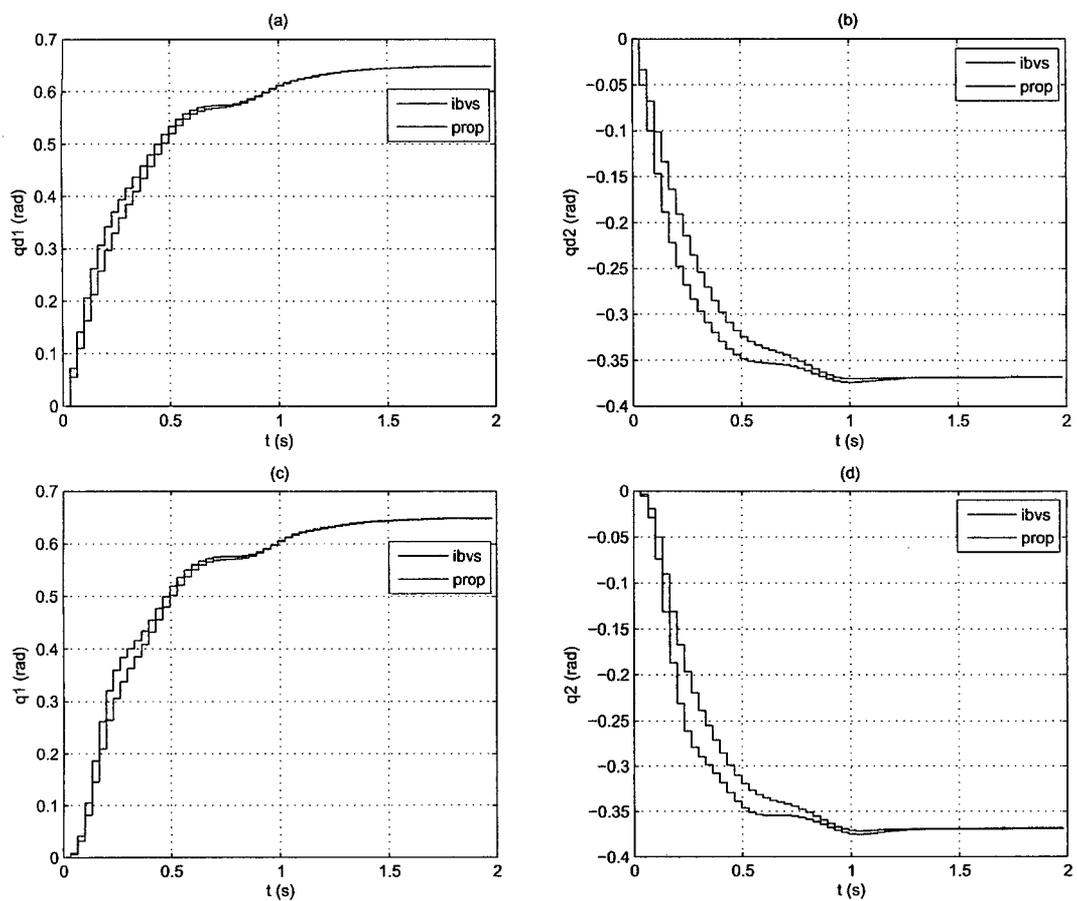


Figura 5.21: Respuesta simulada del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto con movimiento pendular variable. (a) y (b) Posición angular deseada de las articulaciones. (c) y (d) Posición angular actual de las articulaciones.

5.2. Implementación

El programa de control IBVS está implementado utilizando librerías Qt C/C++, OpenCV, OpenGL, GLSL, USB2Dynamixel y DepthSenseSDK. Éste consta de dos *hilos de ejecución*³ y una clase de control principalmente. El primero es el hilo principal (main), en cual se define los entornos gráficos y se inicia el hilo secundario. En este último, se ejecuta la frecuencia y eventos de adquisición de los mapas de color, profundidad y vértices de la cámara DS325. La clase de control recibe información del hilo secundario y ejecuta las instrucciones de control manipulando los servomotores AX-12 (articulaciones). La Figura 5.22 muestra el diagrama de bloques del programa.

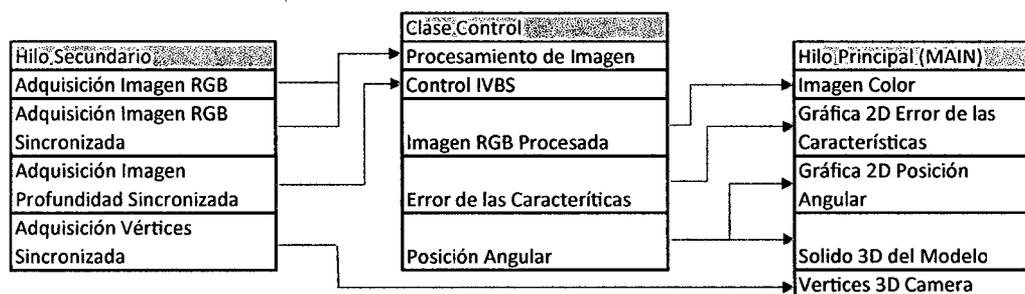


Figura 5.22: Diagrama de bloques del programa del IBVS.

Los entornos gráficos del programa son cuatro. El primero es la Imagen RGB de la cámara de color del DS325 con su respectivo reconocimiento de Objeto. El segundo son las posiciones angulares actuales y el error de las características deseadas. El tercero es el mapa de color sincronizado en 3D de la cámara de profundidad. El cuarto son los movimiento del PTU-CAM en un gráfico 3D del solido. Estos dos últimos entornos 3D utilizan las librerías OpenGL y GLSL reescritas para Qt, que permiten almacenar los gráficos en búfer del GPU y ejecutarlos en el mismo. La Figura 5.23 muestra estos entornos interactuando con piel humana, y a su izquierda muestra 2 vistas extras del mapa de color sincronizado.

³Thread o subproceso, permite ejecutar una tarea al mismo (concurrentemente) tiempo con otra.

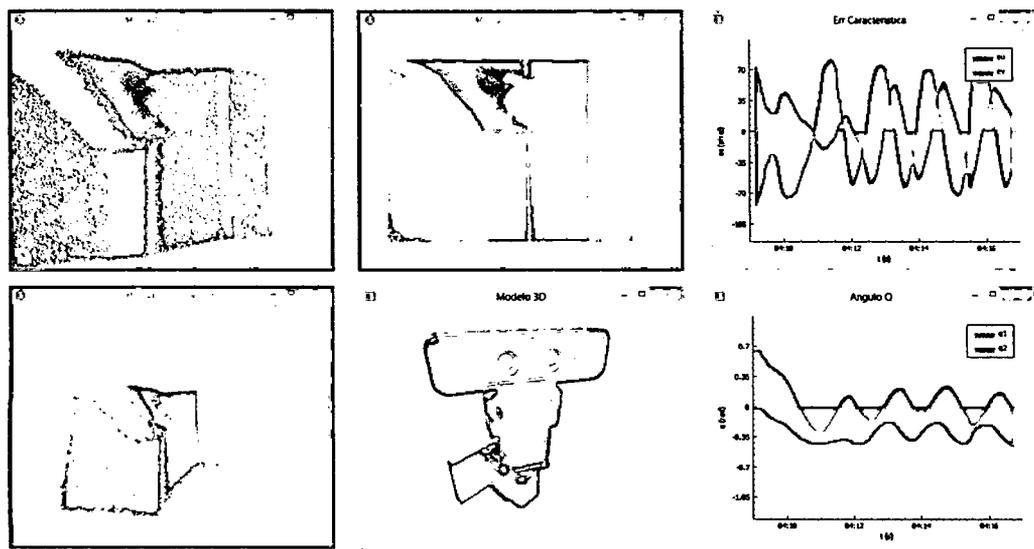


Figura 5.23: Entornos gráficos del control IBVS.

En la Figura 5.24 muestra la disposición del entorno de regulación para ambos controles, y en las siguientes gráficas muestran las respuestas de la implementación del sistema real para un objeto fijo, utilizando el IBVS y el *control proporcional*, donde ambos controles se comparan con sus respectivas simulaciones y entre ellos.

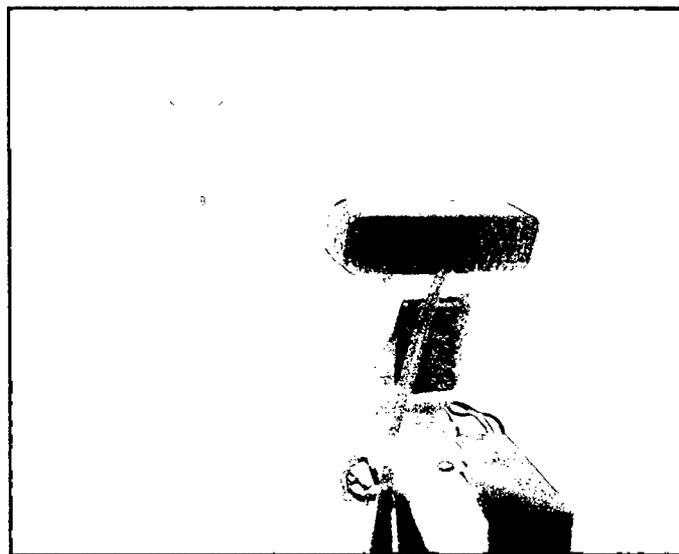


Figura 5.24: Disposición del PTU-CAM y el objeto fijo.

Respuesta real del IBVS y simulación (objeto fijo)

La Figura 5.25a muestra la respuesta real del sistema en el espacio de tareas, donde se encuentra el módulo, el objeto fijo (rojo), y la estimación de las trayectorias de las características visuales (centro del Frame de la cámara) en el IBVS real (azul) y el simulado (verde), sobre el plano $X = 0.5$ (ver Figura 5.25c). La Figura 5.25b muestra el movimiento del objeto visto desde la cámara, para el IBVS real (rojo) y el simulado (cian), hasta llegar al centro del Frame o características visuales deseadas (azul); y la Figura 5.25d muestra las respectivas gráficas de profundidad del objeto respecto de la cámara.

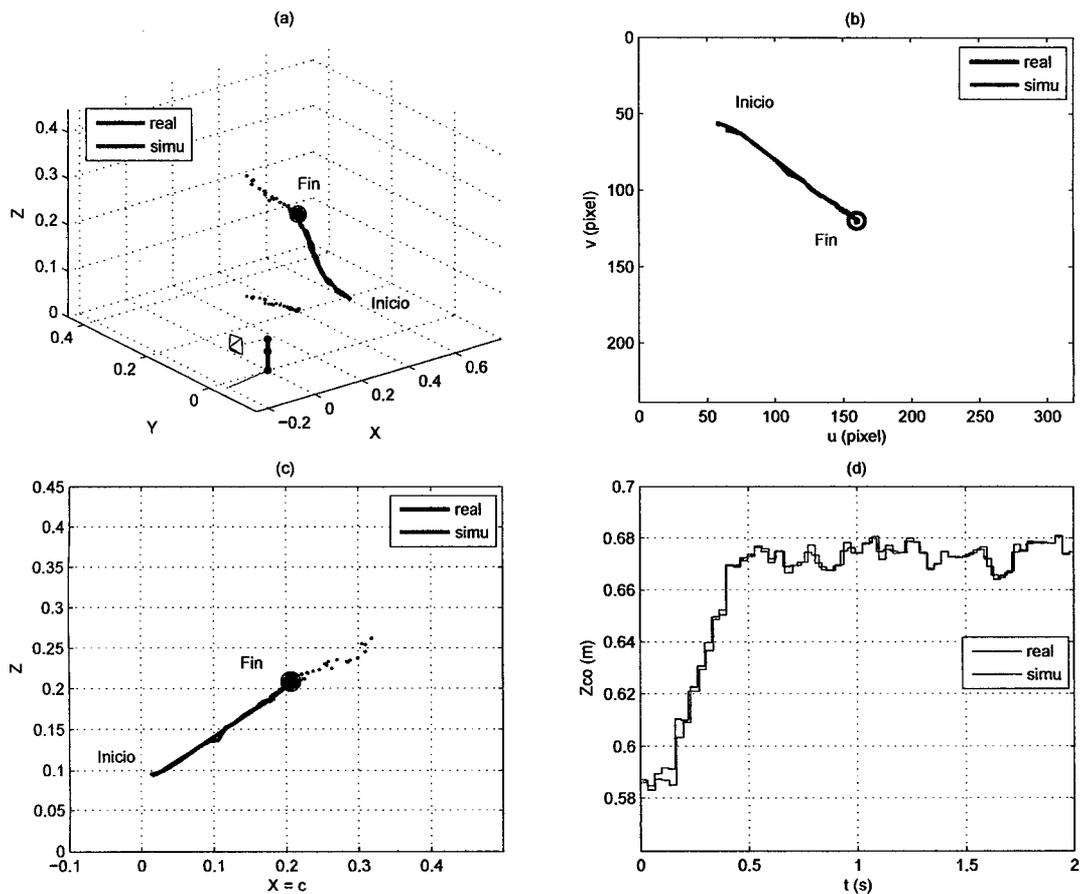


Figura 5.25: Respuesta real y simulada del sistema utilizando el IBVS ($K = 10$) para un objeto fijo. (a) Vista del espacio de tareas con objeto (rojo) y la estimación de las características visuales, sobre el plano $X = 0.5$. (b) Frame de la cámara. (c) Vista del plano $X = 0.5$. (d) Profundidad del objeto respecto a la cámara.

Las Figuras 5.26a y b, muestran las señales de control del IBVS real (azul) y simulado (rojo) para un objeto fijo, donde se observa el trabajo de regulación. Las Figuras 5.26c-f muestran las velocidades (lineal y angular) de la cámara respecto del objeto y expresado en el sistema coordenado de la cámara (señales de salida del IBVS), donde se aprecia la nulidad de la velocidad angular de la cámara en su eje de profundidad, ya que ésta no rota en ese eje.

Las Figuras 5.27a-d, muestran las velocidades (producto del IBVS real y simulado) del extremo del último eslabón del módulo respecto de la base del mismo, donde se aprecia la poca velocidad angular en el eje X , debido a lo antes explicado. Las Figuras 5.27e y f, muestran las respectivas velocidades angulares deseadas de las articulaciones, obtenida del *Jacobiano* del módulo.

Las Figuras 5.28a y b, muestran las velocidades angulares actuales (producto del IBVS real y simulado) de las articulaciones. Las Figuras 5.28c-f, muestran las respectivas posiciones angulares deseadas y actuales.

En estas gráficas muestran que la implementación del IBVS tiene un comportamiento muy aproximado que su simulación, incluso con el ruido natural (puntos pequeños rojos) que genera la adquisición del mapa de profundidad por parte de la cámara (ver Figura 5.25a), cumpliendo con el mismo tiempo de estableciendo aproximado descrito anteriormente y demostrando la veracidad del modelo matemático encontrado.

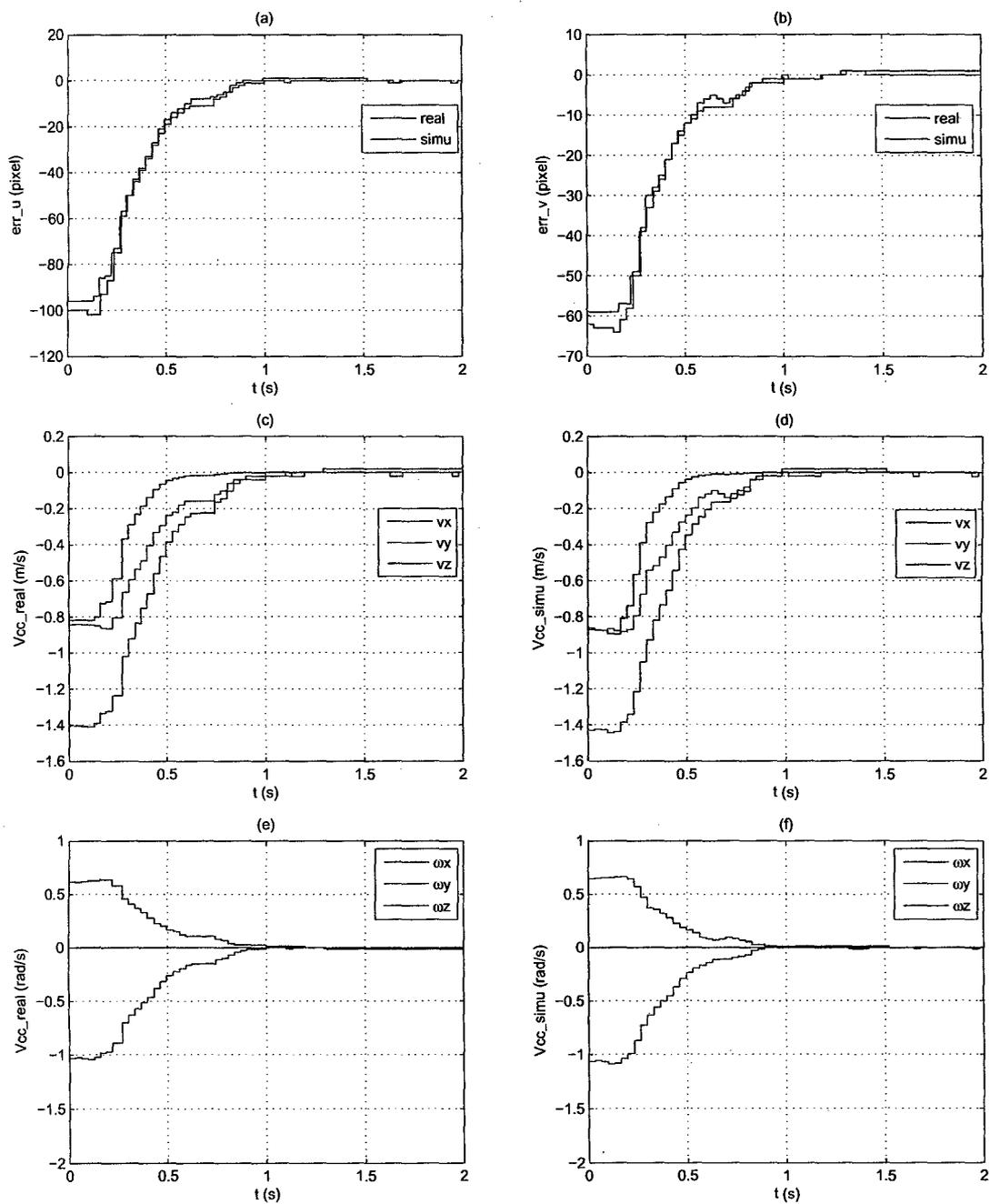


Figura 5.26: Respuesta real y simulada del sistema utilizando el IBVS ($K = 10$) para un objeto fijo. (a) y (b) Error de las características visuales. (c)-(f) Velocidad de la cámara respecto del objeto y expresado en la cámara.

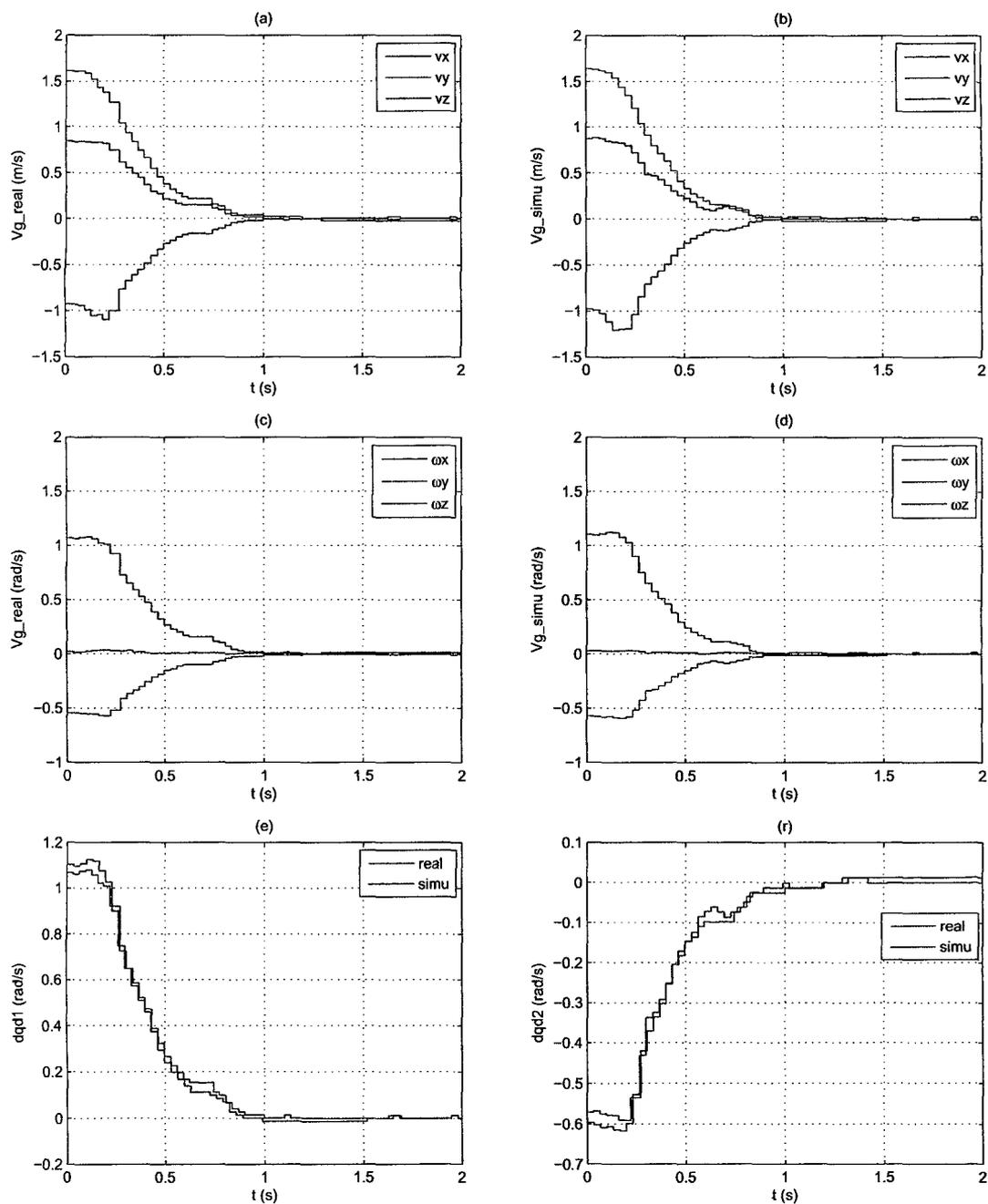


Figura 5.27: Respuesta real y simulada del sistema utilizando el IBVS ($K = 10$) para un objeto fijo. (a)-(d) Velocidad absoluta del último eslabón (gripper) respecto de la base del módulo. (e) y (f) Velocidad deseada de las articulaciones.

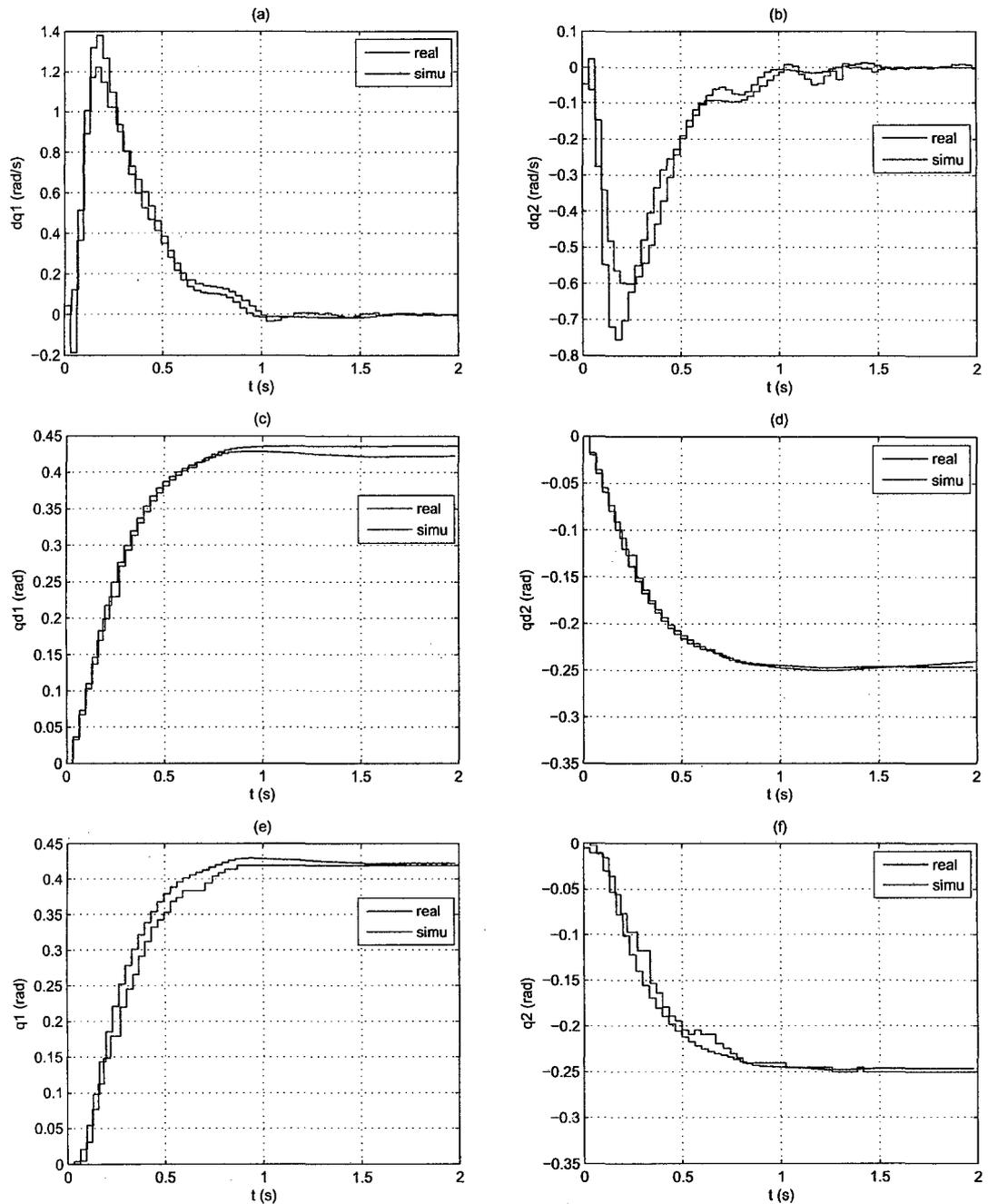


Figura 5.28: Respuesta real y simulada del IBVS ($K = 10$) para un objeto fijo. (a) y (b) Velocidad angular actual de las articulaciones. (c) y (d) Posición deseada de las articulaciones. (e) y (f) Posición actual de las articulaciones.

Respuesta real del control proporcional y simulación (objeto fijo)

La Figura 5.29a muestra la respuesta del sistema en el espacio de tareas, donde se encuentra el módulo, el objeto fijo (rojo), y la estimación de las trayectorias de las características visuales (centro del Frame de la cámara) en el *control proporcional* real (azul) y el simulado (verde), sobre el plano $X = 0.5$ (ver Figura 5.29c). La Figura 5.29b muestra el movimiento del objeto visto desde la cámara, para el *control proporcional* real (rojo) y el simulado (cian), hasta llegar al centro del Frame o características visuales deseadas (azul); y la Figura 5.29d muestra las respectivas gráficas de profundidad del objeto respecto de la cámara.

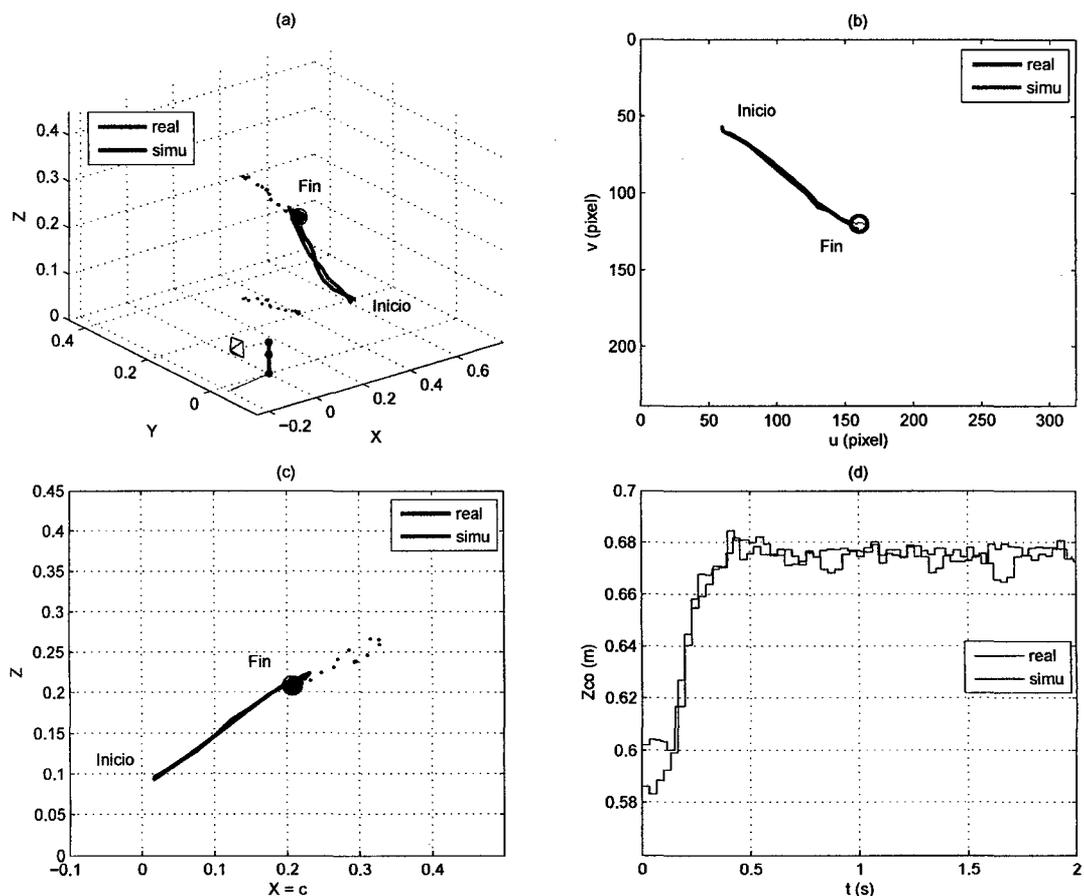


Figura 5.29: Respuesta real y simulada del control proporcional ($K = 0.0183$) para un objeto fijo. (a) Vista del espacio de tareas con objeto (rojo) y la estimación de las características visuales, sobre el plano $X = 0.5$. (b) Frame de la cámara. (c) Vista del plano $X = 0.5$. (d) Profundidad del objeto respecto a la cámara.

Las Figuras 5.30a y b, muestran las señales de control del *control proporcional* real (azul) y simulado (rojo) para un objeto fijo, donde se observa el trabajo de regulación. Las Figuras 5.30c-f, muestran las respectivas velocidades angulares deseadas y actuales de las articulaciones.

Las Figuras 5.31a y b, muestran las posiciones angulares deseadas (producto del *control proporcional* real y simulado) de las articulaciones. las Figuras 5.31c y d, muestran las respectivas posiciones angulares actuales.

En estas gráficas muestran que la implementación del *control proporcional* tiene un comportamiento cercano a su simulación, debido a diversos factores reales como el ruido natural (puntos pequeños rojos) que genera la adquisición del mapa de profundidad por parte de la cámara (ver Figura 5.29a), pero principalmente al no considerar la *cinemática* del módulo provocando un respuesta brusca (movimiento) al llegar al establecimiento.

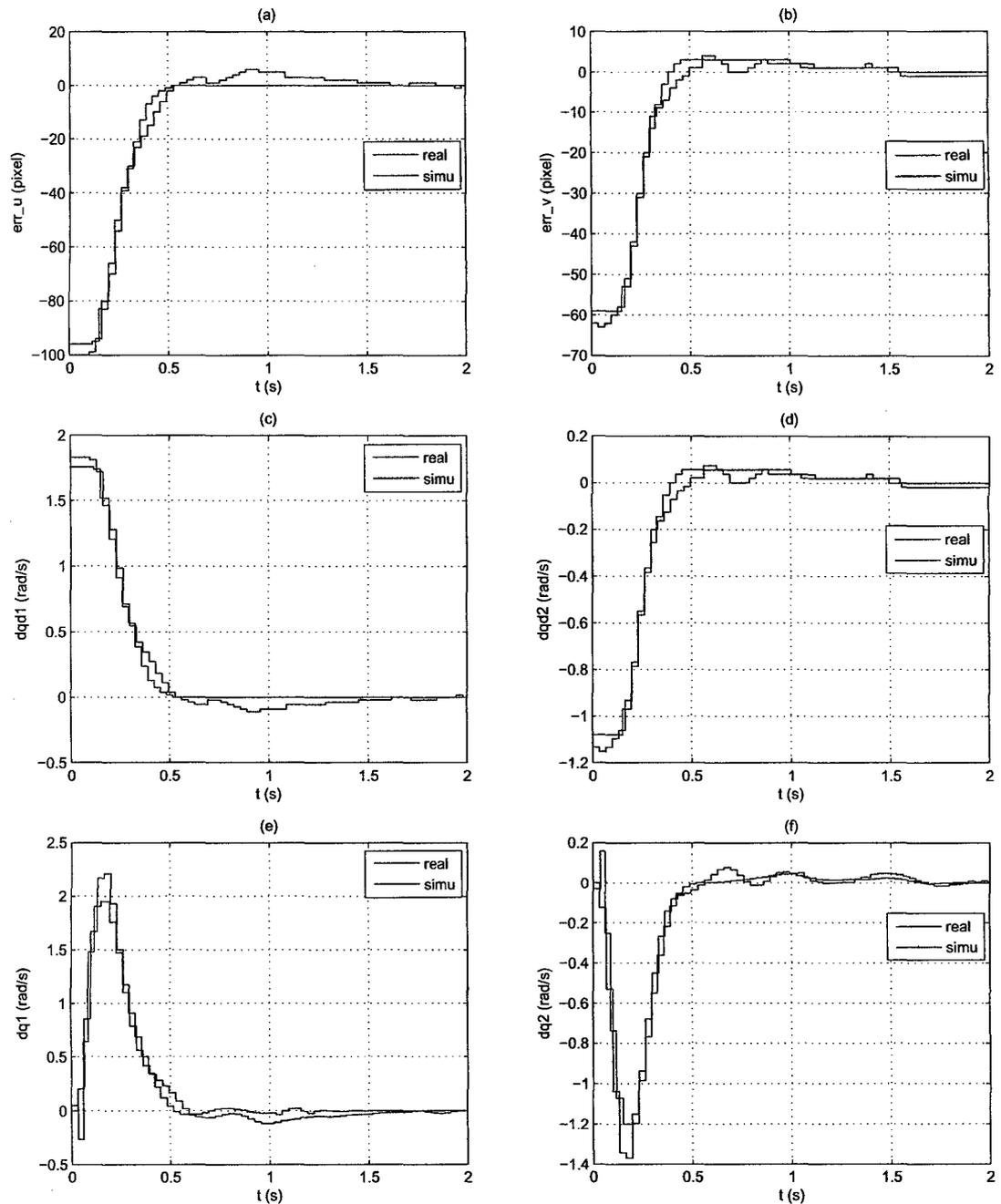


Figura 5.30: Respuesta real y simulada del control proporcional ($K = 0.0183$) para un objeto fijo. (a) y (b) Error de las características visuales. (c), (d) Velocidad deseada de las articulaciones. (e) y (f) Velocidad angular actual de las articulaciones.

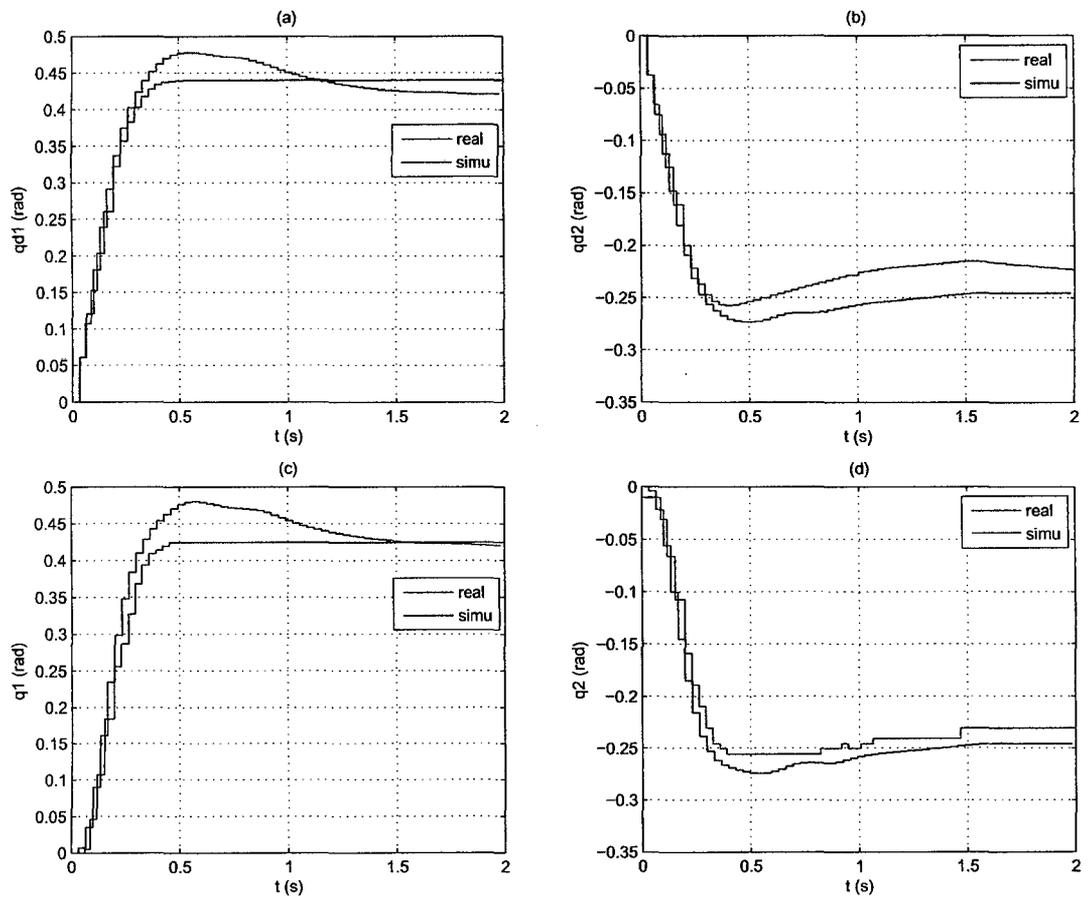


Figura 5.31: Respuesta real y simulada del control proporcional ($K = 0.0183$) para un objeto fijo. (a) y (b) Posición deseada de las articulaciones. (c) y (d) Posición actual de las articulaciones.

Respuesta real del IBVS y control proporcional (objeto fijo)

La Figura 5.32a muestra la respuesta real del sistema, donde se encuentra el módulo, el objeto fijo (rojo), y la estimación de las trayectorias de las características visuales (centro del Frame de la cámara) en el IBVS (azul) y el *control proporcional* (verde), sobre el plano $X = 0.5$ (ver Figura 5.32c). La Figura 5.32b muestra el movimiento del objeto visto desde la cámara, para el IBVS (rojo) y el control proporcional (cian), hasta llegar al centro del Frame o características visuales deseadas (azul); y la Figura 5.32d muestra las respectivas gráficas de profundidad del objeto respecto de la cámara.

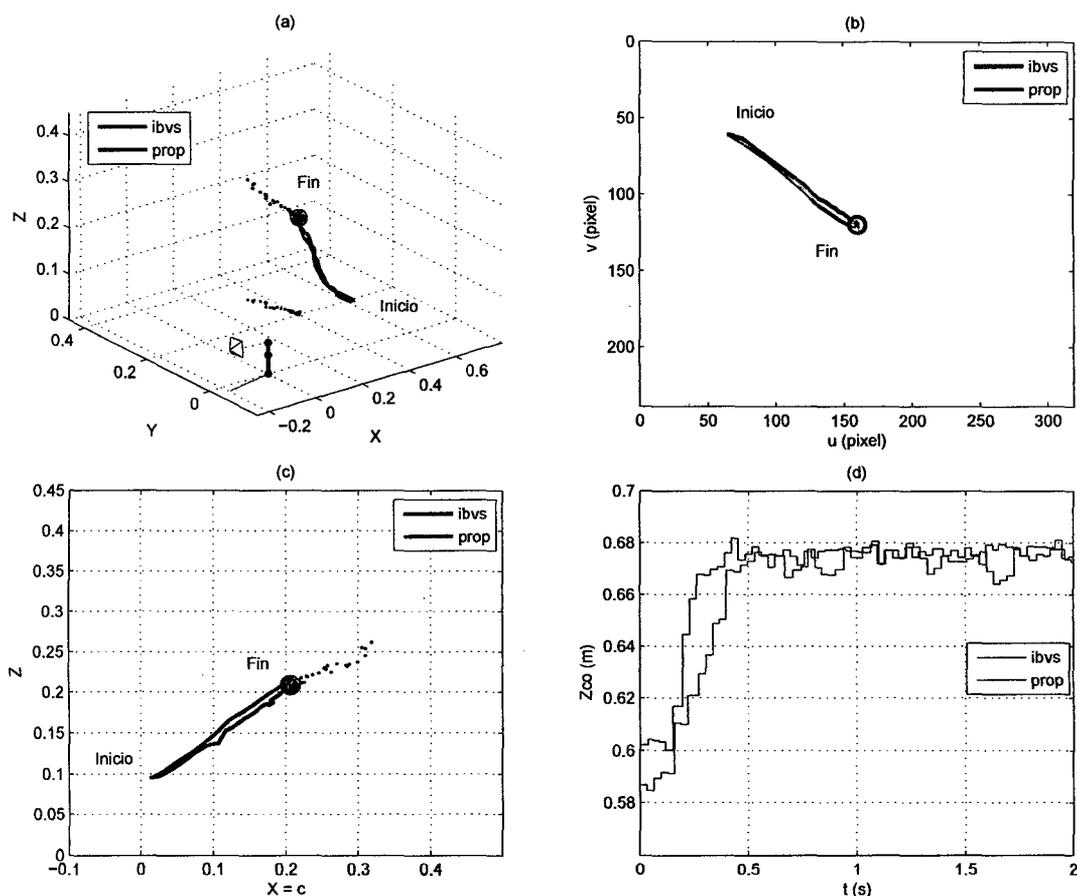


Figura 5.32: Respuesta real del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto fijo. (a) Vista del espacio de tareas con objeto (rojo) y la estimación de las características visuales, sobre el plano $X = 0.5$. (b) Frame de la cámara. (c) Vista del plano $X = 0.5$. (d) Profundidad del objeto respecto a la cámara.

Las Figuras 5.33a y b, muestran las señales de control real del IBVS (azul) y del *control proporcional* (rojo) para un objeto fijo, donde se observa el trabajo de regulación. Las Figuras 5.33c-f, muestran sus respectivas velocidades angulares, deseadas y actuales, de las articulaciones.

La Figura 5.34 muestra las posiciones angulares, deseadas y actuales, de las articulaciones para el IBVS (azul) y el *control proporcional* (rojo).

En estas gráficas reales de comparación muestran que el *control proporcional* tiene un respuesta parecida al IBVS para un objeto fijo, puesto que se hace notoria la tendencia al sobreimpulso y como consecuencia un movimiento mucho mayor en las articulaciones para llegar al mismo tiempo de establecimiento. Por tanto, el IBVS tiene mejor rendimiento que el *control proporcional* en el movimiento y produce una respuesta más suavizada. Por último, la Tabla 5.3 muestra los errores RMS de las características visuales y del movimiento en las articulaciones para ambos control.

Tabla 5.3: Error RMS de la implementación del IBVS y del control proporcional para un objeto fijo.

Control	Objeto fijo	
	e_s (píxeles)	\dot{q} (rpm)
IBVS	23.5870	2.4351
Proporcional	20.8006	3.3865

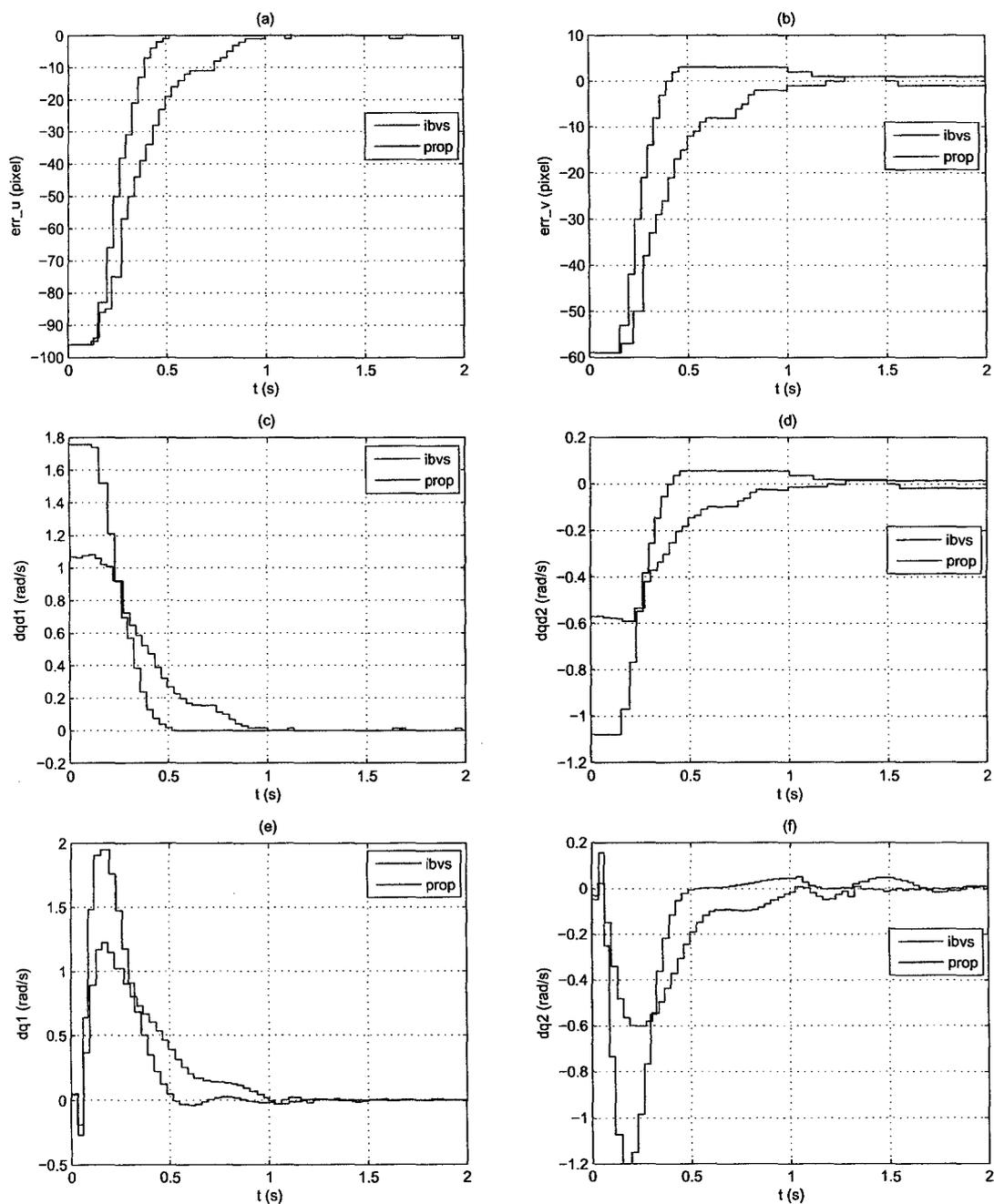


Figura 5.33: Respuesta real del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto fijo. (a) y (b) Error de las características visuales. (c) y (d) Velocidad deseada de las articulaciones. (e) y (f) Velocidad angular actual de las articulaciones.

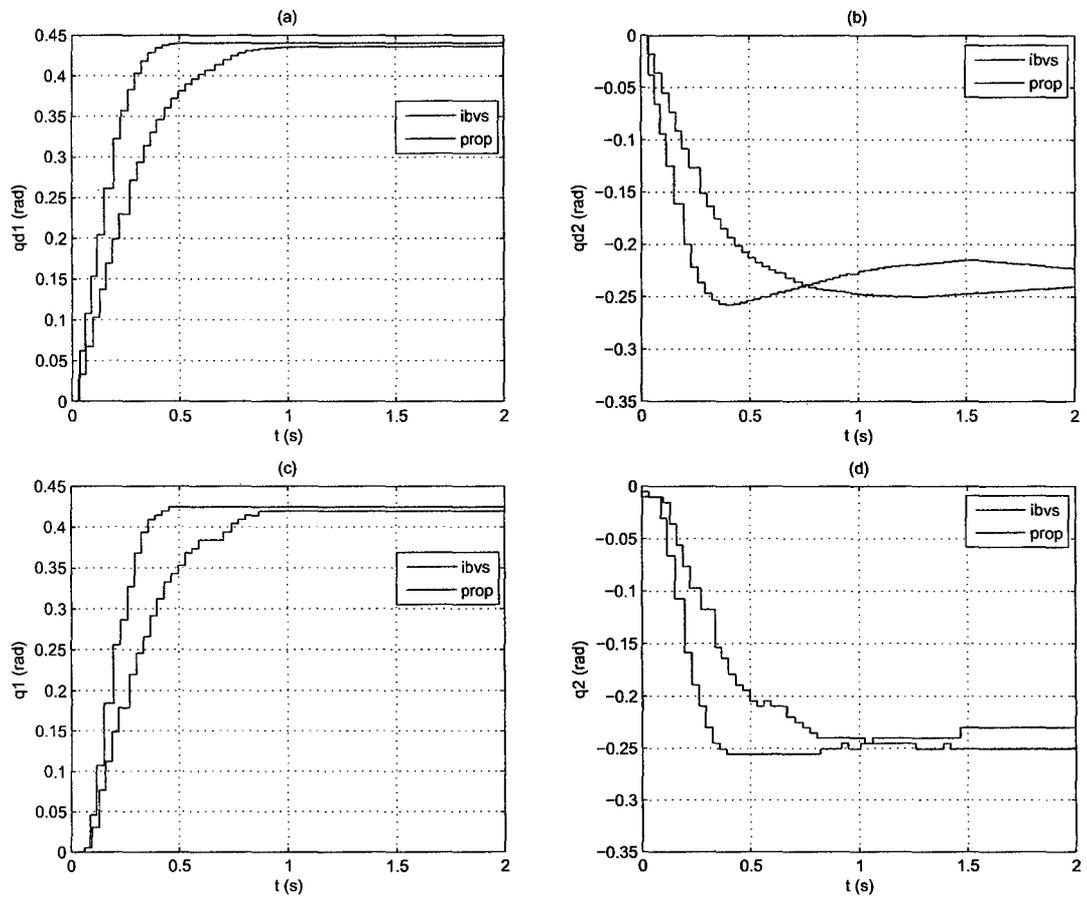


Figura 5.34: Respuesta real del IBVS ($K = 10$) y del control proporcional ($K = 0.0183$) para un objeto fijo. (a) y (b) Posición deseada de las articulaciones. (c) y (d) Posición actual de las articulaciones.

CONCLUSIONES Y TRABAJOS FUTUROS

La presente tesis implementó el *control visual basado en imagen* en un PTU–CAM TOF, para reducir el sobreimpulso en la tarea de regulación de la posición de la cámara respecto de un objeto de color verde deformable, y realizar un movimiento suavizado en su trayecto. También, se encontró un modelo matemático del módulo, para observar el comportamiento del control en simulación, y un algoritmo para encontrar la posición de un objeto en el Frame de la cámara.

A continuación se describirán las conclusiones de la presente tesis y algunos de los posibles trabajos futuros que pueden continuar desarrollándose como resultado de la implementación.

Conclusiones

- La implementación del *control visual basado en imagen* (IBVS) permitió suavizar el movimiento en la tarea de regulación de la posición de la cámara del módulo respecto del objeto deformable, y de las articulaciones en el PTU–CAM, evitando movimientos bruscos sin sobreimpulso, incluso para el ruido que genera la adquisición del mapa de profundidad por parte de la cámara TOF.
- Utilizando el *control visual basado en imagen* (IBVS) se redujo en 28 % la velocidad angular RMS (ver Tabla 5.3) y el sobreimpulso de las señales de control en comparación al *control proporcional*. Por tanto, se asegura

un ahorro importante en la vida útil de los actuadores al utilizar el IBVS, ya que se necesita menos movimiento para llegar a la misma respuesta de forma suavizada y en el mismo tiempo de establecimiento que el *control proporcional*.

- En simulación, se representó al movimiento del ají producido por vientos y vibraciones mediante una trayectoria pendular variable, donde el IBVS mejoró el desempeño en el movimiento de las articulaciones del PTU-CAM reduciendo en 11 % la velocidad angular RMS (ver Tabla 5.2) en comparación al *control proporcional*.
- El error en características visuales entre la respuesta real y simulada es de 1.9184 píxeles en el IBVS (ver las Figuras 5.26a y b) y de 2.5204 píxeles en el *control proporcional* (ver las Figuras 5.30a y b). Esto indica que el IBVS tiene un comportamiento más definido en simulación, incluso para errores en posición angular característicos en los servomotores.
- El modelo matemático del PTU-CAM tuvo un error RMS de 1.0939° y 1.253° (ver las Figuras 5.2a y b) en comparación a la respuesta real al escalón unitario (rad) para cada articulación. Este error RMS está relacionado con variación en el tiempo de muestreo real, ya que depende de la gestión del sistema operativo para la comunicación, procesamiento y control de datos (ver Figura 5.1).
- El desarrollo del algoritmo para la identificación de la región de interés de la imagen, centroide del objeto deformable o características visuales principales, permitió reconocer el 63 % del objeto para una baja iluminación artificial (40 %). Esto puede apreciarse en la Figura 3.13 y la Tabla 3.3.
- La formulación $AX = ZB$ usando el producto de *Kronecker*, tiene un error RMS de 0.3308 en el proceso de calibración *cámara-robot*, con 22 iteracio-

nes, 50 datos de imágenes y configuraciones geométrica del módulo. Con éste se obtuvo un desplazamiento aproximado de 19 cm. del centro focal del modelo de la cámara respecto de su centro de masa (ver Tabla 4.2).

Trabajos Futuros

- Al agregar más *grados de libertad* al módulo de reconocimiento y ubicación de objetivos, permite una mayor área de trabajo y versatilidad en el movimiento. Sin embargo, esto genera problemas de estabilidad y vibración que perturban el desempeño en la adquisición de la información visual por la cámara, siendo mínimo en 2 GDL pero muy apreciable en más *grados de libertad*. Por tanto, un modelo más completo del módulo y controles no lineales deben ser explorados como compensación al movimiento, esto constituye el principal estudio para un futuro trabajo.

BIBLIOGRAFÍA

- [1] Peter I. Corke. *Visual Control of Robots: High-Performance Visual Servoing*. Research Studies Press Ltd., November 1996.
- [2] Zill e Hussnain, Sounkalo Dembélé, and Nicolas Andreff. Visual servoing in medical robotics. In *IEEE International Workshop on Biomedical Robotics and Biomechatronics*, Rome, Italy, June 2012.
- [3] J.-F. Lots, D. M. Lane, E. Trucco, and F. Chaumette. A 2d visual servoing for underwater vehicle station keeping. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2767–2772, Seoul, Korea, May 2001.
- [4] Rahul Rao, R. Vijay Kumar, and Camillo J. Taylor. Visual servoing of a ugv from a uav using differential flatness. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 743–748, 2003.
- [5] Rong-Fong Fung. *Visual Servoing*. InTech, April 2010.
- [6] Francois Chaumette and A. Santos. Tracking a moving object by visual servoing. In *12th World Congress, International Federation of Automatic Control*, volume 9, pages 409–414, Sydney, Australia, July 1993.
- [7] M. A. El-Bardini, E. A. Elsheikh, and M. A. Fkirin. Real time object tracking using image based visual servo technique. *International Journal of Computer Science and Emerging Technologies*, 2(2):252–257, April 2011.

- [8] Paolo Robuffo Giordano, Alessandro De Luca, and Giuseppe Oriolo. 3d structure identification from image moments. In *IEEE International Conference on Robotics and Automation*, pages 93–100, Pasadena, CA, May 2008.
- [9] Masaaki Shibata, Hideki Eto, and Masahide Ito. Image-based visual tracking to fast moving target for active binocular robot. In *36th Annual Conference on IEEE Industrial Electronics Society*, pages 2727–2732, Glendale, AZ, November 2010.
- [10] Minoru Asada, Takamaro Tanaka, and Koh Hosoda. Visual tracking of unknown moving object by adaptive binocular visual servoing. In *IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, number 99TH8480, pages 249–254, Taipei, August 1999.
- [11] Grégory Flandin, François Chaumette, and Eric Marchand. Eye-in-hand/eye-to-hand cooperation for visual servoing. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2741–2746, San Francisco, CA, April 2000.
- [12] Xavier Clady, François Collange, Frédéric Jurie, and Philippe Martinet. Object tracking with a pan-tilt-zoom camera: application to car driving assistance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1653–1658, December 2001.
- [13] Paul Y. Oh and Rares I. Stanciu. Enhancing camera operator performance with computer vision based control. *International Journal of Signal Processing*, 1(2):113–119, 2004.
- [14] Sang joo Kim, Jin woo Park, and Jang-Myung Lee. Implementation of tracking and capturing a moving object using a mobile robot. *International Journal of Control, Automation, and Systems*, 3(3):444–452, September 2005.

- [15] Jaehong Park, Wonsang Hwang, Wook Bahn, Chang hun Lee, Tae il Kim, Muhammad Muneeb Shaikh, Kwang soo Kim, and Dong il Cho. Pan-tilt camera control for vision tracking system based on the robot motion and vision information. In *18th International Federation of Automatic Control World Congress*, pages 3165–3170, Milano, Italy, August 2011.
- [16] Cao Songxiao, Wang Xuanyin, and Xiang Ke. Visual object servo tracking based on the particle filter method using a pan-tilt-zoom camera. *International Journal of Advanced Robotic Systems*, 9(134), 2012.
- [17] Paramveer S. Dhillon. Robust real-time face tracking using an active camera. *Computational Intelligence in Security for Information Systems Advances in Intelligent and Soft Computing*, 63:179–186, 2009.
- [18] Tiago Miguel Corredoura Vaz Rato. *Actuated Character: Humanoid robots for videogames interaction*. PhD thesis, Universidade Técnica de Lisboa, Lisboa, Portugal, June 2012.
- [19] www.robotis.com.
- [20] Anibal Ollero Baturone. *Robotica, Manipuladores Y Robots Moviles*. Alfaomega & Marcombo, 1st edition, April 2007.
- [21] A. Barrientos, L. Peñín, B. Balaguer, and R. Aracil. *Fundamentos de Robótica*. McGraw–Hill, 2nd edition, julio 2007.
- [22] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, 5th edition, November 2005.
- [23] Pedro Daniel Dinis Teodoro. *Humanoid Robot: Development of a simulation environment of an entertainment humanoid robot*. PhD thesis, Universidade Técnica de Lisboa, Lisboa, Portugal, September 2007.

- [24] Rafael Kelly, Victor Santibáñez Davila, and Julio Antonio Loría Perez. *Control of Robot Manipulators in Joint Space*. Springer, 2005 edition, June 2005.
- [25] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice–Hall, 2nd edition, November 2011.
- [26] Chris Harris and Mike Stephens. A combined corner and edge detector. *Plessey Research Roke Manor*, pages 147–151, 1988.
- [27] Gerhard X. Ritter and Joseph N. Wilson. *Handbook of Computer Vision Algorithms in Image Algebra*. CRC Press Inc, 2nd edition, September 2000.
- [28] Rastislav Lukac and Kostantinos N. Plataniotis. *Color Image Processing: Methods and Applications*. CRC Press, 1st edition, October 2006.
- [29] Andreas Koschan and Mongi Abidi. *Digital Color Image Processing*. Wiley–Interscience, 1st edition, April 2008.
- [30] Bruno Siciliano and Oussama Khatib. *Springer Handbook of Robotics*. Springer, May 2008.
- [31] Maximiliano Bueno López and Marco A. Arteaga Pérez. Control servovisual de robots manipuladores: un enfoque hacia la industria. *Epsilon*, pages 9–31, Julio–Diciembre 2012.
- [32] Juan Ernesto Solanes Galbis. *Visual Servoing Multifrecuencia*. PhD thesis, Departamento de Ingeniería de Sistemas y Automática, Valencia, España, Diciembre 2010.
- [33] Farabi Bensalah and Francois Chaumette. Compensation of abrupt motion changes in target tracking by visual servoing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 181–187, Pittsburgh, PA, August 1995.

- [34] Klaus H. Strobl and Gerd Hirzinger. Optimal hand-eye calibration. In *IEEE/RSJ International Conference on Intelligent Robots and System*, pages 4647–4653, Beijing, China, October 2006.
- [35] Mili Shah, Roger D. Eastman, and Tsai Hong. An overview of robot-sensor calibration methods for evaluation of perception systems. College Park, MD, March 2012.
- [36] Fadi Dornaika and Radu Horaud. Simultaneous robot-world and hand-eye calibration. *IEEE Transactions on Robotics and Automation*, 14(4):617–622, August 1998.
- [37] Zijian Zhao. Hand-eye calibration using convex optimization. In *IEEE International Conference on Robotics and Automation Shanghai International Conference Center*, pages 2947–2952, Shanghai, China, May 2011.
- [38] Konstantinos Daniilidis. Hand-eye calibration using dual quaternions. *The International Journal of Robotics Research*, 18(3):286–298, March 1999.
- [39] Roger Y. Tsai and Reimar K. Lenz. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(6):345–358, June 1989.
- [40] Lin Wang Aiguo Li and Defeng Wu. Simultaneous robot-world and hand-eye calibration using dual-quaternions and kronecker product. *International Journal of the Physical Sciences*, 5(10):1530–1536, September 2010.

APÉNDICE A

Movimiento de Cuerpos Rígidos

A.1. Matriz de Rotación

La *matriz de rotación* es la representación de la orientación de un sistema coordinado $\{S_i\}$ adjunto a cuerpo rígido respecto de otro $\{S_j\}$ en el espacio euclidiano, y tiene la forma

$${}^jR_i = (\hat{x} \ \hat{y} \ \hat{z}) = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}, \quad (\text{A.1})$$

donde \hat{x}, \hat{y} y \hat{z} , de orden 3×1 , son vectores unitarios ortonormales entre si que representan a los ejes de la orientación de $\{S_i\}$ respecto de $\{S_j\}$.

Tabla A.1: Propiedades de la matriz de rotación

-
- $R \in SO(3)$
 - $R^{-1} \in SO(3)$
 - $R^{-1} = R^T$
 - Las columnas o las filas son ortogonales entre sí.
 - Cada columna o cada fila es un vector unitario.
 - $\det(T) = 1$
-

Tabla A.2: Rotaciones elementales de la matriz de rotación

Rotación de $\{S_i\}$ alrededor de \hat{z}_j un ángulo θ :

$$\text{Rot}(\hat{z}_j, \theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Rotación de $\{S_i\}$ alrededor de \hat{y}_j un ángulo θ :

$$\text{Rot}(\hat{y}_j, \theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$$

Rotación de $\{S_i\}$ alrededor de \hat{x}_j un ángulo θ :

$$\text{Rot}(\hat{x}_j, \theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix}$$

Tabla A.3: Representaciones de orientación a matrices de rotación equivalentes

Z-Y-X Ángulos de Euler (α, β, γ) :

$${}^j R_i = \begin{pmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{pmatrix}$$

X-Y-Z Ángulos fijos (ψ, θ, ϕ) :

$${}^j R_i = \begin{pmatrix} c_\phi c_\theta & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ s_\phi c_\theta & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{pmatrix}$$

Ángulo-eje $\theta \hat{\omega}$:

$${}^j R_i = \begin{pmatrix} \omega_x^2 v_\theta + c_\theta & \omega_x \omega_y v_\theta - \omega_z s_\theta & \omega_x \omega_z v_\theta + \omega_y s_\theta \\ \omega_x \omega_y v_\theta + \omega_z s_\theta & \omega_y^2 v_\theta + c_\theta & \omega_y \omega_z v_\theta - \omega_x s_\theta \\ \omega_x \omega_z v_\theta - \omega_y s_\theta & \omega_y \omega_z v_\theta + \omega_x s_\theta & \omega_z^2 v_\theta + c_\theta \end{pmatrix}$$

(Fórmula de Rodrigues': ${}^j R_i = I + \sin \theta S(\hat{\omega}) + (1 - \cos \theta)(\hat{\omega} \hat{\omega}^T - I)$)

Cuaternios $(\epsilon_0, \epsilon_1, \epsilon_2, \epsilon_3)^T$:

$${}^j R_i = \begin{pmatrix} 1 - 2(\epsilon_2^2 + \epsilon_3^2) & 2(\epsilon_1 \epsilon_2 - \epsilon_0 \epsilon_3) & 2(\epsilon_1 \epsilon_3 + \epsilon_0 \epsilon_2) \\ 2(\epsilon_1 \epsilon_2 + \epsilon_0 \epsilon_3) & 1 - 2(\epsilon_1^2 + \epsilon_3^2) & 2(\epsilon_2 \epsilon_3 - \epsilon_0 \epsilon_1) \\ 2(\epsilon_1 \epsilon_3 - \epsilon_0 \epsilon_2) & 2(\epsilon_2 \epsilon_3 + \epsilon_0 \epsilon_1) & 1 - 2(\epsilon_1^2 + \epsilon_2^2) \end{pmatrix}$$

Abreviaciones: $c_\theta := \cos \theta$, $s_\theta := \sin \theta$, $v_\theta := 1 - \cos \theta$ y similares

Tabla A.4: Matriz de rotación a representaciones de orientación equivalentes

Z-Y-X Ángulos de Euler (α, β, γ) :

$$\beta = \text{Atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2})$$

$$\alpha = \text{Atan2}\left(\frac{r_{21}}{\cos \beta}, \frac{r_{11}}{\cos \beta}\right)$$

$$\gamma = \text{Atan2}\left(\frac{r_{32}}{\cos \beta}, \frac{r_{33}}{\cos \beta}\right)$$

X-Y-Z Ángulos fijos (ψ, θ, ϕ) :

$$\theta = \text{Atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2})$$

$$\psi = \text{Atan2}\left(\frac{r_{21}}{\cos \theta}, \frac{r_{11}}{\cos \theta}\right)$$

$$\phi = \text{Atan2}\left(\frac{r_{32}}{\cos \theta}, \frac{r_{33}}{\cos \theta}\right)$$

Ángulo-eje (Rodrigues') $\theta \hat{\omega}$:

$$\theta = \cos^{-1}\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right)$$

$$\hat{\omega} = \frac{1}{2 \sin \theta} \begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix}$$

Cuaternios $(\epsilon_0, \epsilon_1, \epsilon_2, \epsilon_3)^T$:

$$\epsilon_0 = \frac{1}{2} \sqrt{1 + r_{11} + r_{22} + r_{33}}$$

$$\epsilon_1 = \frac{r_{32} - r_{23}}{4\epsilon_0}$$

$$\epsilon_2 = \frac{r_{13} - r_{31}}{4\epsilon_0}$$

$$\epsilon_3 = \frac{r_{21} - r_{12}}{4\epsilon_0}$$

A.2. Matriz Antisimétrica

En general, una matriz S de orden $n \times n$ se dice que es *antisimétrica* sí y solo sí

$$S^T + S = 0, \quad (\text{A.2})$$

es decir, sus elementos tiene la relación de $s_{ij} + s_{ji} = 0$, a excepción de los elementos de su diagonal que son nulos $s_{ii} = 0$. Por tanto, si S es de orden 3×3 y tiene un vector $a = (a_x, a_y, a_z)^T$ de valores independientes como entrada, entonces

$$S(a) = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & -a_x & 0 \end{pmatrix}. \quad (\text{A.3})$$

Tabla A.5: Propiedades de la matriz antisimétrica

-
- $S(\alpha a + \beta b) = \alpha S(a) + \beta S(b)$, a y $b \in \mathbb{R}^{3 \times 1}$
 - $S(a)b = a \times b$
 - $R(a \times b) = Ra \times Rb$, $R \in SO(3)$
 - $RS(a)R^T = S(Ra)$
 - $S^T(a) = S(-a)$
-

A.3. Derivada de la Matriz de Rotación

Suponiendo que la *matriz de rotación* está función de un ángulo θ alrededor de vector unitario \hat{k} , es decir, $R = \text{Rot}(\hat{k}, \theta) \in SO(3)$ y puesto que R es ortogonal para todo θ , entonces

$$RR^T = I. \quad (\text{A.4})$$

Derivando (A.4) respecto de θ , se tiene

$$\frac{dR}{d\theta}R^T + R\frac{dR^T}{d\theta} = 0. \quad (\text{A.5})$$

Es fácil demostrar que (A.5) tiene la forma de (A.2), si se define

$$S = \frac{dR}{d\theta}R^T. \quad (\text{A.6})$$

Por tanto, para $R = \text{Rot}(\hat{k}, \theta) \in SO(3)$, se tiene

$$\frac{dR}{d\theta} = S(\hat{k})R. \quad (\text{A.7})$$

Suponiendo que $R(t) = \text{Rot}(\hat{k}, \theta(t)) \in SO(3)$, entonces

$$\dot{R} = \frac{dR}{dt} = \frac{dR}{d\theta} \frac{d\theta}{dt} = \dot{\theta} S(\hat{k})R = S(\omega)R, \quad (\text{A.8})$$

donde $\omega = (\omega_x, \omega_y, \omega_z)^T$ es el vector de la velocidad angular.

A.4. Velocidad Lineal

A.4.1. Movimiento de un Punto Visto desde la Referencia

El vector posición de un punto ${}^i r$, perteneciente a un cuerpo rígido que tiene adjunto un sistema coordenado $\{S_i\}$ respecto de otro $\{S_j\}$, está dado por

$${}^j r = {}^j R_i {}^i r + {}^j p_i. \quad (\text{A.9})$$

Suponiendo que el punto tiene un movimiento rígido, $R(t) := {}^j R_i(t)$ y $p(t) := {}^j p_i(t)$, respecto de $\{S_j\}$ y se quiere encontrar su velocidad, entonces se debe derivar (A.9), obteniendo

$${}^j \dot{r} = \dot{R} {}^i r + \dot{p}, \quad (\text{A.10})$$

donde se debe notar que ${}^i \dot{r} = 0$, debido a que el punto y $\{S_i\}$ pertenecen al mismo cuerpo rígido.

Desarrollando (A.10) con ayuda de (A.8) se tiene

$${}^j \dot{r} = S(\omega) R {}^i r + \dot{p} = S(\omega) {}^i r^{(j)} + v = \omega \times {}^i r^{(j)} + v, \quad (\text{A.11})$$

donde $v := {}^j v_i$ y $\omega := {}^j \omega_i$ son los vectores de velocidad lineal y angular respectivamente de $\{S_i\}$ respecto de $\{S_j\}$.

A.4.2. Punto Fijo Visto desde la Referencia en Movimiento

Considerando a $\{S_i\}$ como fijo (sistema coordenado de referencia), donde ${}^i r$ es invariante en el tiempo, y $\{S_j\}$ con movimiento rígido, $R(t) := {}^i R_j(t)$ y $p(t) := {}^i p_j(t)$, entonces, al igual que (A.9) se tiene

$${}^j r = {}^i R_j^T {}^i r - {}^i R_j^T p_j, \quad (\text{A.12})$$

donde ${}^j r$ es el vector posición del punto fijo respecto de $\{S_j\}$ en movimiento. Para encontrar la velocidad de éste punto se debe derivar (A.12), obteniendo

$${}^j \dot{r} = \dot{R}^T ({}^i r - p) - R^T \dot{p}. \quad (\text{A.13})$$

Desarrollando (A.13) con ayuda de (A.8) y la Tabla A.5, se tiene

$$\begin{aligned} {}^j \dot{r} &= (S(\omega)R)^T ({}^i r - p) - R^T \dot{p}, \\ &= R^T S^T(\omega) ({}^i r - p) - R^T \dot{p}, \\ &= R^T (S(-\omega) ({}^i r - p)) - R^T \dot{p}, \\ &= -R^T \omega \times R^T {}^j r^{(i)} - v^{(j)}, \end{aligned} \quad (\text{A.14})$$

donde $v^{(j)} := {}^i v_j^{(j)} = R^T \dot{p}$ es el vector de velocidad lineal de $\{S_j\}$ respecto de $\{S_i\}$ y expresado en $\{S_j\}$, $\omega := {}^i \omega_j$ es el vector de velocidad angular de $\{S_j\}$ respecto de $\{S_i\}$ y ${}^j r^{(i)} = ({}^i r - p)$.

Finalmente, la velocidad de un punto fijo visto desde $\{S_j\}$ en movimiento es

$${}^j \dot{r} = -\omega^{(j)} \times {}^j r - v^{(j)}, \quad (\text{A.15})$$

donde $\omega^{(j)} := {}^i \omega_j^{(j)} = R^T \omega$ es el vector de velocidad angular de $\{S_j\}$ respecto de $\{S_i\}$ y expresado en $\{S_j\}$, y ${}^j r = R^T {}^j r^{(i)}$.

A.5. Velocidad Angular

Si se tiene un sistema coordenados $\{S_i\}$ que está adjunto a un cuerpo rígido con movimiento respecto de $\{S_0\}$ (${}^0 R_i(t)$ y ${}^0 p_i(t)$), y otro sistema coordenado $\{S_j\}$ que está adjunto al mismo cuerpo rígido (${}^0 R_j(t)$ y ${}^0 p_j(t)$) y relacionado con $\{S_i\}$ por ${}^i R_j$ y ${}^i p_j$, entonces se puede demostrar que la velocidad angular para

cualquier punto perteneciente a un cuerpo rígido es la misma, como

$$\begin{aligned}
 {}^0R_i(t) &= {}^0R_j(t)^jR_i, \\
 {}^0\dot{R}_i &= {}^0\dot{R}_j^jR_i, \\
 S(\omega_i){}^0R_i &= S(\omega_j){}^0R_j^jR_i, \\
 S(\omega_i) &= S(\omega_j),
 \end{aligned} \tag{A.16}$$

donde $\omega_i := {}^0\omega_i$ y $\omega_j := {}^0\omega_j$ son los vectores de velocidad angular de $\{S_i\}$ y $\{S_j\}$ respectivamente, y son iguales si ambos pertenecen a un mismo cuerpo rígido.

A.6. Transformación de Velocidad

Si se tiene un sistema coordenado $\{S_i\}$ adjunto a un cuerpo rígido con movimiento rígido respecto de un sistema coordenado de referencia $\{S_0\}$ (${}^0R_i(t)$ y ${}^0p_i(t)$), y se conoce la velocidad (v_j, ω_j) de un sistema coordenado $\{S_j\}$ respecto de $\{S_0\}$, adjunto al mismo cuerpo y relacionado con $\{S_i\}$ por iR_j y ip_j , es decir,

$${}^0p_j(t) = {}^0p_i(t) + {}^0R_i(t){}^ip_j, \tag{A.17}$$

y se quiere encontrar la velocidad (v_i, ω_i) de $\{S_i\}$, entonces, derivando (A.17), teniendo en cuenta (A.8), la Tabla A.5 y que $\{S_i\}$ y $\{S_j\}$ pertenecen al mismo cuerpo rígido, se obtiene

$$\begin{aligned}
 v_i &= v_j - \omega_j \times {}^ip_j^{(0)}, \\
 \omega_i &= \omega_j,
 \end{aligned} \tag{A.18}$$

donde $v_i := {}^0v_i$, $v_j := {}^0v_j$ y $\omega_i := {}^0\omega_i$, $\omega_j := {}^0\omega_j$ son los vectores de velocidad lineal y angular de $\{S_i\}$ y $\{S_j\}$ respectivamente, y ${}^ip_j^{(0)} = {}^0R_i{}^ip_j$ es el vector de posición de $\{S_j\}$ respecto de $\{S_i\}$ pero expresado en $\{S_0\}$.

Ahora, si se quiere expresar la velocidad (v_i, ω_i) respecto de su $\{S_i\}$, entonces

de (A.18) se tiene

$$\begin{aligned}
 {}^0R_i^T v_i &= {}^0R_i^T v_j - {}^0R_i^T (\omega_j \times {}^i p_j^{(0)}), \\
 v_i^{(i)} &= v_j^{(i)} + ({}^0R_i^T {}^i p_j^{(0)}) \times ({}^0R_i^T \omega_j), \\
 v_i^{(i)} &= v_j^{(i)} + ({}^0R_i^T {}^i p_j^{(0)}) \times ({}^0R_i^T {}^0R_j {}^0R_j^T \omega_j), \\
 v_i^{(i)} &= {}^i R_j v_j^{(j)} + S({}^i p_j) ({}^i R_j \omega_j^{(j)}),
 \end{aligned} \tag{A.19}$$

y

$$\begin{aligned}
 {}^0R_i^T \omega_i &= {}^0R_i^T \omega_j, \\
 \omega_i^{(i)} &= {}^i R_j \omega_j^{(j)}.
 \end{aligned} \tag{A.20}$$

Finalmente, escribiendo (A.19) y (A.20) matricialmente se tiene

$$\begin{pmatrix} v_i^{(i)} \\ \omega_i^{(i)} \end{pmatrix} = \begin{pmatrix} {}^i R_j & S({}^i p_j) {}^i R_j \\ 0_{3 \times 3} & {}^i R_j \end{pmatrix} \begin{pmatrix} v_j^{(j)} \\ \omega_j^{(j)} \end{pmatrix} \tag{A.21}$$