

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



**DISEÑO DE UN ALGORITMO DE CORTE CON PENETRACIÓN CONTROLADA CON
BISTURÍ EMPLEANDO UN MANIPULADOR ROBÓTICO**

TESIS

**PARA OBTENER EL GRADO ACADÉMICO DE
MAESTRO EN CIENCIAS EN INGENIERÍA ELECTRÓNICA CON MENCIÓN EN
AUTOMÁTICA E INSTRUMENTACIÓN**

**ELABORADO POR
JUAN CARLOS SUÁREZ QUISPE**

**ASESOR
ARTURO ROJAS MORENO, Ph.D.**

LIMA - PERÚ

2022

DEDICATORIA

Dedico el presente trabajo a toda mi familia por apoyarme siempre, a mis padres por darme las cosas que ellos nunca tuvieron, a mis hermanos por compartirme su experiencia y en especial a la memoria de mi padre Efrén Suárez León. Dios lo guarde en su gloria.

AGRADECIMIENTOS

A PROCIENCIA por el apoyo financiero en el desarrollo de esta tesis mediante el proyecto titulado “Control Haptico Remoto de un Sistema Robótico para Aplicación en Cirugía”, con CONTRATO 144-2018-FONDECYT/BM.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO I.	
ANTECEDENTES Y DESCRIPCIÓN DEL PROBLEMA	3
1.1. Antecedentes	3
1.2. Descripción de la realidad problemática	12
1.3. Formulación del problema	13
1.4. Justificación e importancia de la investigación	13
1.5. Objetivos	14
1.5.1. Objetivo general	14
1.5.2. Objetivos específicos	14
1.6. Hipótesis	14
1.6.1. Hipótesis principal	14
1.6.2. Hipótesis secundarias	15
1.7. Variables e indicadores	15
1.8. Unidad de análisis	16
1.9. Tipo de investigación	16
1.10. Nivel de investigación	16
1.11. Periodo de análisis	16
1.12. Fuentes de información e instrumentos utilizados	16
1.13. Técnicas de recolección y procesamiento de datos	17
CAPÍTULO II.	
MARCO TEÓRICO Y CONCEPTUAL	18
2.1. Espacio de color RGB	18
2.2. Espacio de color HSV	18
2.3. Conceptos de lógica difusa	19
2.3.1. Conjunto crisp	19
2.3.2. Conjunto difuso	19
2.3.3. Implicación difusa	19
2.4. El coeficiente de correlación de Pearson	20

CAPÍTULO III.

DESARROLLO DE LA TESIS	21
3.1. Especificaciones de diseño para la realización de cortes controlados	21
3.1.1. Definición del sistema dinámico a controlar	21
3.1.2. Especificaciones de los cortes a realizar	21
3.1.3. Especificaciones de control	22
3.2. Desarrollo del efector final	22
3.2.1. Diseño del efector final	23
3.2.2. Diseño de la plataforma de muestras	25
3.2.3. El robot MK2 modificado	25
3.2.4. Interface electrónica de operación del robot MK2 modificado	28
3.3. Análisis de color	29
3.3.1. Imagen tomada	30
3.3.2. Píxel de color y región de color	30
3.3.3. Píxel negro	31
3.3.4. La paleta de colores	31
3.3.5. Obtención de la región interior de la paleta de colores	32
3.3.6. Detección de los píxeles de color en la paleta de colores	34
3.3.7. Análisis de la paleta de colores	35
3.3.8. Detección de píxeles de color por el criterio de umbrales de color	40
3.4. Algoritmo de agrupación de píxeles por cuatro barridos	42
3.5. Definición de las pseudo-marcas de color	45
3.6. Obtención de las pseudo-marcas de color candidatas	46
3.7. Análisis de confiabilidad de pseudo-regiones de color candidatas	48
3.7.1. Criterio de medias y medianas	48
3.7.2. Criterio de varianzas	49
3.7.3. Criterio de áreas	50
3.7.4. Obtención de las pseudo-marcas de color	51
3.8. Obtención de imagen reducida	54
3.8.1. La región del bisturí	54
3.8.2. La pseudo-región del bisturí	54
3.8.3. La imagen reducida	55
3.9. Obtención de la pseudo-región del bisturí	60
3.9.1. Algoritmo de obtención de la pseudo-región del bisturí	65
3.10. Estimación del grado de penetración del bisturí	67
3.11. Control de la profundidad de corte con bisturí	70

3.11.1. Aplicación de un controlador proporcional integral derivativo	71
3.11.2. Aplicación de un controlador difuso	74
3.11.3. Aplicación de un controlador proporcional integral difuso	80
3.11.4. Elección de la técnica de control de profundidad de corte	89
3.12. Algoritmo de corte con penetración controlada	90

CAPÍTULO IV.

ANÁLISIS DE LOS RESULTADOS DE LA INVESTIGACIÓN Y CONTRASTACIÓN DE LAS HIPÓTESIS 94

4.1. Configuración experimental	94
4.2. Medición de la profundidad de corte	97
4.3. Resultados de la investigación	99
4.4. Contrastación de la hipótesis principal	101
4.5. Contrastación de las hipótesis secundarias	101

CONCLUSIONES 104

RECOMENDACIONES 106

BIBLIOGRAFÍA 107

ANEXO A.

SUB-RUTINAS DE LÓGICA DIFUSA 112

A.1. La sub-rutina <i>rule</i>	112
A.2. La sub-rutina <i>rule_K</i>	114

ANEXO B.

LA INTERFACE ELECTRÓNICA DE OPERACIÓN DEL ROBOT MK2 MODIFICADO 118

ANEXO C.

MEDICIÓN DE LA PROFUNDIDAD DE CORTE 120

ÍNDICE DE ILUSTRACIONES

1.1. (a) Configuración experimental para la medida de la fuerza y la profundidad de corte. (b) Sección de corte de tejido blando obtenida bajo el método de control propuesto.	3
1.2. Fotografía tomada por la cámara instantánea.	5
1.3. Perfiles de corte para tiempos incrementados de exposición (de izquierda a derecha: 2.5s, 3.5s, y 4.5s) y densidad de poder, frecuencia de escaneo y longitud constantes.	6
1.4. Mecanismo del efector final.	7
1.5. (a) Estructura del sistema robótico de corte. (b) Extracción de la curva de corte.	8
1.6. Configuración experimental.	9
1.7. (a) Configuración experimental para medir las fuerzas de corte en hígado de cerdo. (b) Módulo de Young efectivo local para segmentos de carga de corte típico.	10
1.8. (a) Configuración experimental. (b) Fotografía para la estimación de la profundidad de corte usando procesamiento de imágenes, para una profundidad de $L_{b1} - L_{b2}$	11
1.9. Plataforma experimental.	12
3.1. Vistas renderizadas isométrica (a), y posterior (b) del acople general.	23
3.2. Vistas renderizadas isométrica (a), y posterior (b) de la base del efector final.	24
3.3. Vistas renderizadas isométrica (a), y posterior (b) del soporte del bisturí con dos agarraderas.	24
3.4. Vistas renderizadas frontal (a), superior (b), e isométrica (c) del efector final.	24
3.5. (a) Vista renderizada de la plataforma de muestras. (b) Plataforma de muestras.	25
3.6. Robot EEZbotARM MK2.	26
3.7. (a) Vista renderizada del adaptador para el acople del efector final. (b) Vista renderizada de la estructura mecánica del robot MK2 modificado, con el efector final acoplado.	26
3.8. Robot MK2 modificado, con el efector final acoplado.	27
3.9. Algoritmo; generar $ang1$ y $ang2$	28
3.10. Imagen tomada.	30
3.11. Representación gráfica de una paleta de colores.	32

3.12. Algoritmo; obtener la paleta de colores.	33
3.13. (a) Región interior de la paleta de colores. (b) Píxeles de color rojos, verdes y azules, detectados en la paleta de colores.	34
3.14. Representación de los valores numéricos de los píxeles de color de la paleta de colores en los canales R, G, B, H, S, y V. En los 6 sub-gráficos, el eje horizontal representa la cantidad de píxeles de color detectados en la paleta de colores, mientras que el eje vertical representa el valor numérico de cada píxel de color encontrado en la misma, en los canales R, G, B, H, S, y V.	35
3.15. Región cuadrada con puntos.	37
3.16. Algoritmo; analizar la paleta de colores.	39
3.17. Algoritmo; detectar píxeles de color.	41
3.18. (a) Imagen tomada. (b) Representación gráfica de los píxeles de color detectados utilizando el criterio de umbrales de color.	41
3.19. (a) Aplicación del primer barrido. (b) Aplicación del segundo barrido.	43
3.20. (a) Aplicación del tercer barrido. (b) Aplicación del cuarto barrido.	43
3.21. Algoritmo; agrupar píxeles por cuatro barridos.	45
3.22. Algoritmo; obtener la pseudo-marca candidata.	47
3.23. (a) Imagen IMG_{RGB} . (b) Aplicación de erosión y dilatación a IMG_{RGB} . (c) Imagen en color, contenedora de las 3 pseudo-marcas de color candidatas.	48
3.24. Algoritmo; evaluar la confiabilidad.	52
3.25. (a) Imagen tomada. (b) Pseudo-marcas de color.	53
3.26. (a) Imagen tomada. (b) Región del bisturí.	54
3.27. (a) Imagen tomada. (b) Representación binaria de la pseudo-región del bisturí en color blanco con fondo negro.	55
3.28. Algoritmo; obtener la imagen reducida.	59
3.29. (a) Imagen tomada IMG . (b) IMG rotada α grados (c) Imagen Reducida.	60
3.30. Canales extraídos de una imagen reducida; R (a), G (b), B (c), H (d), S (e) y V (f).	60
3.31. Bordes detectados en los canales extraídos de una imagen reducida; R (a), G (b), B(c), H(d), S(e), y V(f).	61
3.32. (a) Imagen de fusión de bordes. (b) Imagen de fusión de bordes con aplicación de un filtro media antes de detección de bordes. (c) Imagen con contornos cerrados rellenados, IMG_{SCL1}	61
3.33. (a) Imagen IMG_{SCL1} . (b) Aplicación de erosión. (c) Descarte de grupos pequeños. (d) Aplicación de dilatación.	62

3.34. Ejemplos de casos exitosos de obtención de la pseudo-región del bisturí. De izquierda a derecha; imágenes reducidas representadas en escala de grises, imágenes <i>RT</i> , imágenes <i>IMG_{SCL1}</i> , imágenes <i>IMG_{SCL4}</i>	63
3.35. Ejemplos de casos fallidos de obtención de la pseudo-región del bisturí. De izquierda a derecha; imágenes reducidas representadas en escala de grises, imágenes <i>RT</i> , imágenes <i>IMG_{SCL1}</i> , imágenes <i>IMG_{SCL4}</i>	64
3.36. Algoritmo; obtener la pseudo-región en la imagen reducida.	65
3.37. Algoritmo; obtener la pseudo-región del bisturí.	67
3.38. Representación del bisturí penetrando en la muestra.	68
3.39. Algoritmo; obtener el punto más bajo de la pseudo-región del bisturí.	69
3.40. Esquema de la aplicación del controlador PID.	71
3.41. Profundidad de corte controlada con el controlador PID.	72
3.42. Algoritmo; controlador PID.	73
3.43. Respuesta del sistema controlado PID ante una entrada de tipo escalón de 5mm de amplitud.	73
3.44. Esquema de la aplicación del controlador difuso.	74
3.45. Conjuntos difusos para <i>e</i> , <i>ce</i> y <i>u</i>	75
3.46. Profundidad de corte controlada con el controlador difuso.	77
3.47. Algoritmo; controlador difuso.	79
3.48. Respuesta del sistema controlado difuso ante una entrada de tipo escalón de 5mm de amplitud.	80
3.49. Esquema de la aplicación del control PI-difuso.	81
3.50. Conjuntos difusos para <i>e</i>	82
3.51. Conjuntos difusos para <i>ua</i>	83
3.52. Conjuntos difusos para <i>op</i> y <i>oi</i>	83
3.53. Profundidad de corte controlada con el controlador PI-difuso.	85
3.54. Algoritmo; controlador PI-difuso.	86
3.55. Algoritmo; <i>get_fKp</i>	87
3.56. Algoritmo; <i>get_fKi</i>	88
3.57. Respuesta del sistema controlado PI-difuso ante una entrada de tipo escalón de 5mm de amplitud.	89
3.58. Algoritmo; algoritmo de corte con penetración controlada.	93
4.1. Muestra de pecho de pollo sujeta a la plataforma de muestras.	94
4.2. Papel sanitario colocado sobre una muestra de pecho de pollo.	95
4.3. Plataforma de muestras y cámara de video ligeramente inclinadas.	95
4.4. Disposición de materiales.	96

4.5. Sección de muestra de pecho de pollo con dos líneas de abertura.	97
4.6. Sección de muestra de pecho de pollo con dos líneas de abertura resaltadas. .	97
4.7. Cara seccionada vista con un microscopio digital.	98
4.8. Medición de la profundidad de corte.	98
4.9. Profundidades de corte medidas en relación a las referencias de control de profundidad de corte.	101
4.10. Efecto final acoplado al sensor FT300, sosteniendo un bisturí. Esta fotografía fue tomada en el Laboratorio de Mecatrónica de la Universidad de Ingeniería y Tecnología UTEC.	102
A.1. Algoritmo; $rule$	112
A.2. Algoritmo; $trian_v$	113
A.3. Algoritmo; $trian_d$	114
A.4. Algoritmo; $rule_K$	115
A.5. Algoritmo; $trian_vK$	116
A.6. Algoritmo; $trape_d$	117
B.1. Esquema electrónico de la interface electrónica de operación del robot MK2 modificado.	118
B.2. Diseño de la tarjeta electrónica impresa de la interface electrónica de operación del robot MK2 modificado.	119
B.3. Interface electrónica de operación del robot MK2 modificado.	119
C.1. Prueba #1; cara seccionada (a), medición de la profundidad de corte (b).	120
C.2. Prueba #2; cara seccionada (a), medición de la profundidad de corte (b).	120
C.3. Prueba #3; cara seccionada (a), medición de la profundidad de corte (b).	121
C.4. Prueba #4; cara seccionada (a), medición de la profundidad de corte (b).	121
C.5. Prueba #5; cara seccionada (a), medición de la profundidad de corte (b).	121
C.6. Prueba #6; cara seccionada (a), medición de la profundidad de corte (b).	122
C.7. Prueba #7; cara seccionada (a), medición de la profundidad de corte (b).	122
C.8. Prueba #8; cara seccionada (a), medición de la profundidad de corte (b).	122
C.9. Prueba #9; cara seccionada (a), medición de la profundidad de corte (b).	123
C.10. Prueba #10; cara seccionada (a), medición de la profundidad de corte (b).	123
C.11. Prueba #11; cara seccionada (a), medición de la profundidad de corte (b).	123
C.12. Prueba #12; cara seccionada (a), medición de la profundidad de corte (b).	124
C.13. Prueba #13; cara seccionada (a), medición de la profundidad de corte (b).	124
C.14. Prueba #14; cara seccionada (a), medición de la profundidad de corte (b).	124

C.15. Prueba #15; cara seccionada (a), medición de la profundidad de corte (b). . . .	125
C.16. Prueba #16; cara seccionada (a), medición de la profundidad de corte (b). . . .	125
C.17. Prueba #17; cara seccionada (a), medición de la profundidad de corte (b). . . .	125
C.18. Prueba #18; cara seccionada (a), medición de la profundidad de corte (b). . . .	126
C.19. Prueba #19; cara seccionada (a), medición de la profundidad de corte (b). . . .	126
C.20. Prueba #20; cara seccionada (a), medición de la profundidad de corte (b). . . .	126
C.21. Prueba #21; cara seccionada (a), medición de la profundidad de corte (b). . . .	127
C.22. Prueba #22; cara seccionada (a), medición de la profundidad de corte (b). . . .	127
C.23. Prueba #23; cara seccionada (a), medición de la profundidad de corte (b). . . .	127
C.24. Prueba #24; cara seccionada (a), medición de la profundidad de corte (b). . . .	128
C.25. Prueba #25; cara seccionada (a), medición de la profundidad de corte (b). . . .	128
C.26. Prueba #26; cara seccionada (a), medición de la profundidad de corte (b). . . .	128
C.27. Prueba #27; cara seccionada (a), medición de la profundidad de corte (b). . . .	129
C.28. Prueba #28; cara seccionada (a), medición de la profundidad de corte (b). . . .	129
C.29. Prueba #29; cara seccionada (a), medición de la profundidad de corte (b). . . .	129
C.30. Prueba #30; cara seccionada (a), medición de la profundidad de corte (b). . . .	130

ÍNDICE DE TABLAS

3.1. Reglas difusas de $FRM1$	76
3.2. Reglas difusas de $FRMp$	84
3.3. Reglas difusas de $FRMi$	84
4.1. Profundidades de corte obtenidas con referencias de 5mm.	100
4.2. Profundidades de corte obtenidas con referencias de 7.5mm.	100
4.3. Profundidades de corte obtenidas con referencias de 10mm.	100

RESUMEN

El presente trabajo de tesis se orientó al diseño de un algoritmo de corte con penetración controlada con bisturí, empleando un manipulador robótico. Este algoritmo utiliza como variable controlada, la profundidad de corte, empleada para medir el grado de penetración del bisturí, y como señal de control usa la referencia de rapidez angular enviada al servomotor que ejerce el movimiento vertical del efector final del manipulador robótico. El algoritmo emplea además múltiples constantes que dependen del criterio del usuario tales como factores de tolerancia, factores de escalamiento, parámetros que definen los conjuntos difusos, entre otros. Este algoritmo fue diseñado para realizar cortes verticales con bisturí a superficies planas de tejido blando y se enfoca, exclusivamente, en el control del grado de penetración de los cortes, por lo que no considera el análisis o el control del perfil de la incisión dejada por el bisturí en la muestra, ya que no es objeto de esta investigación. Este trabajo no abarcó un estudio de los rangos máximos de profundidades de corte controladas realizables, empleando este algoritmo; sin embargo, las herramientas utilizadas para las pruebas (bisturí, robot, muestra, entre otras) permitieron realizar incisiones con referencias de control de profundidad de corte de hasta 10mm. Este algoritmo fue desarrollado para ejercer el control de la profundidad de corte en función a información extraída de imágenes tomadas por una cámara de video ubicada en una posición fija con línea de vista perpendicular al plano que forma la hoja de un bisturí. Por su parte, durante las pruebas del algoritmo, el bisturí fue sostenido por el efector final, diseñado y fabricado para el presente trabajo de tesis, que tuvo como finalidad proveer la capacidad de sostener un bisturí al manipulador robótico.

El algoritmo se compone de varias sub-rutinas y su funcionamiento puede ser resumido en dos etapas. En la primera se estima el grado de penetración del bisturí y en la segunda se calcula la señal de control. La estimación del grado de penetración se consigue detectando la región del bisturí dentro de una sub-imagen extraída de una imagen tomada por la cámara de video. Esta sub-imagen es obtenida localizando tres referencias visuales colocadas sobre el mango del bisturí, denominadas marcas de color. Por su parte, el cálculo de la señal de control se realiza utilizando un controlador PI-difuso (proporcional integral difuso) en el que las ganancias de control, proporcional e integral, son calculadas empleando una máquina de reglas difusas para cada una. Dicho controlador fue elegido tras evaluar el funcionamiento de los controladores PID, difuso y PI-difuso, y encontrar que únicamente

los controladores PID y PI-difuso cumplieron con las especificaciones de control al permitir obtener tiempos de asentamiento menores a 90 segundos y sobreelongaciones máximas porcentuales menores al 25 %. La evaluación tuvo como resultado la elección del controlador PI-difuso, debido a su mayor versatilidad de implementación.

El algoritmo de corte con penetración controlada, realizado en un programa de computadora, fue utilizado para la ejecutar cortes con penetración controlada con bisturí, empleando un manipulador robótico. Las pruebas de corte fueron efectuadas haciendo incisiones con bisturí a muestras de pecho de pollo. Esta clase de muestra (pecho de pollo) fue elegida debido a su presencia en la bibliografía citada, como tejido blando, y su fácil acceso en el mercado local.

Se realizaron 10 incisiones por cada una de las 3 referencias (5mm, 7.5mm y 10mm), por lo que se realizó un total de 30 pruebas. Los resultados de las pruebas mostraron que se obtuvieron errores relativos porcentuales de las profundidades de corte controladas de entre -33.00 % y 30.80 %; valores que son similares a los citados en la bibliografía. Por su parte, se obtuvo un coeficiente de correlación de Pearson de 0.89, que al ser cercano a 1, evidenció una fuerte correlación entre las profundidades de corte controladas obtenidas experimentalmente y sus respectivas referencias de control de profundidad de corte. Por estas razones, se concluyó que los resultados obtenidos de la aplicación del algoritmo de corte con penetración controlada fueron satisfactorios y, en consecuencia, se cumplió con el objetivo principal de la tesis.

ABSTRACT

The aim of this thesis is to design a cutting algorithm with controlled penetration with a scalpel, using a manipulator robot. This algorithm uses the cutting depth as a controlled variable, used to measure the scalpel penetration degree, and as a control signal it uses the angular speed reference sent to the servomotor that generates the vertical movement of the end effector of the manipulator robot. The algorithm also uses multiple constants that depend on the user's criteria, such as tolerance factors, scaling factors, parameters that define the fuzzy sets, and others. This algorithm was designed to perform vertical cuts with the scalpel on flat soft tissue surfaces and focuses exclusively on controlling the penetration degree of the cuts, so it does not consider the analysis or control of the incision profile made by the scalpel during the incision, since it is not the aim of this research. This work did not include a study of the maximum ranges of achievable controlled cutting depths, using this algorithm; however, the tools used for the tests (scalpel, robot, sample, and others) made it possible to make incisions with cutting depth control references of up to 10mm. This algorithm was developed to control the cutting depth based on information extracted from images taken by a video camera located in a fixed position with a line of sight perpendicular to the scalpel blade. During the tests of the algorithm, the scalpel was hold by the end effector, designed and manufactured for this thesis work, which had the purpose of providing the ability to hold a scalpel to the manipulator robot.

The algorithm is conformed of several sub-routines and its operation can be summarized in two stages. In the first, the scalpel penetration degree is estimated and in the second, the control signal is calculated. The estimation of the penetration degree is achieved by detecting the scalpel region within a sub-image extracted from an image taken by the video camera. This sub-image is obtained by locating three visual references placed on the scalpel handle, called color marks. The calculation of the control signal is performed using a PI-fuzzy controller (fuzzy integral proportional controller) in which the control gains, proportional and integral, are calculated using a fuzzy rule machine for each one. That controller was chosen after evaluating the operation of the PID, fuzzy and PI-fuzzy controllers, and finding that only the PID and PI-fuzzy controllers achieve the control specifications by obtaining settling times of less than 90 seconds and maximum percentage overshoots less than 25%. The evaluation resulted in the choice of the fuzzy-PI controller, due to its greater implementation

versatility.

The controlled penetration cutting algorithm, performed in a computer program, was used to make controlled penetration cuts with a scalpel, using a manipulator robot. The cutting tests were carried out by making incisions with a scalpel on chicken breast samples. This kind of sample (chicken breast) was chosen due to its presence in the bibliography, as soft tissue, and its easy access in the local market.

10 incisions were made for each of the 3 references (5mm, 7.5mm and 10mm), for which a total of 30 tests were performed. The results of the tests showed that percentage relative errors of the controlled cutting depths between -33.00 % and 30.80 % were obtained; values that are similar to those in the literature. For its part, a Pearson correlation coefficient of 0.89 was obtained, which, being close to 1, showed a strong correlation between the controlled cutting depths obtained from the tests and their respective cutting depth control references. For these reasons, it was concluded that the results obtained from the application of the controlled penetration cutting algorithm were successful and, consequently, the main objective of the thesis was accomplished.

INTRODUCCIÓN

En la actualidad existen diferentes sistemas robóticos orientados a la realización de cortes quirúrgicos controlados, tanto comerciales como de investigación. Algunos de ellos fundamentan su desarrollo en la idea básica de que normalmente la realización de este tipo de cortes depende de la habilidad particular, adquirida por medio de la experiencia, que tenga un cirujano en específico. Esto implica que la falta de aquella habilidad representa un problema para otros médicos, así como también dificultades con la precisión y repetitividad de las incisiones. Los cortes con penetración controlada, en diferentes casos, abordan el problema de la estimación de la profundidad de corte, por medio del análisis de la relación entre esta (la profundidad de corte) y la fuerza de corte. En estos casos, se tiene el problema de que aquella fuerza depende de la respuesta mecánica del tejido, la cual no necesariamente es constante incluso en diferentes ubicaciones de una misma muestra. Asimismo, en otros casos, aquella estimación se aborda desde el punto de vista del análisis de imágenes en el que el cálculo matemático no depende de la respuesta mecánica. La realización de cortes controlados implica también el uso de alguna técnica de control que rija la lógica por medio de la cual se realice la incisión, hecho que ha sido abordado por diferentes autores.

El presente trabajo de tesis se desarrolló como parte de una de las actividades del proyecto identificado con código 144-2018-FONDECYT/BM, llevado a cabo por la Universidad de Ingeniería y Tecnología (UTEC) y financiado por Concytec – Fondecyt. Este trabajo se realizó en base a un manipulador robótico de dos grados de libertad, de los que únicamente fue necesario controlar uno, y tuvo como objetivo diseñar un algoritmo de corte con penetración controlada con bisturí, empleando un manipulador robótico, para la realización de cortes en muestras de tejido blando de superficie plana.

El algoritmo de corte con penetración controlada con bisturí, empleando un manipulador robótico, en este trabajo de tesis, será denominado algoritmo de corte con penetración controlada. Este algoritmo fue desarrollado en base al análisis de imágenes tomadas por una cámara de video ubicada en una posición fija con línea de vista perpendicular con respecto al plano que forma la hoja de un bisturí sostenido por el efector final, diseñado y fabricado para el presente trabajo de tesis, el mismo que fue acoplado a un manipulador robótico.

El capítulo I comprende los antecedentes de trabajos que abordan la realización de cortes con penetración controlada, la descripción de la realidad problemática, la formulación del problema, la justificación e importancia de la investigación, los objetivos, las hipótesis, las variables e indicadores, la unidad de análisis, el tipo de investigación, el nivel de investigación, el periodo de análisis, las fuentes de información e instrumentos utilizados, y, finalmente, las técnicas de recolección y procesamiento de datos.

El capítulo II muestra un marco teórico referente a la descripción de los espacios de color RGB y HSV, conceptos de lógica difusa enfocados a la implicación difusa y, finalmente, al coeficiente de correlación de Pearson.

El capítulo III expone el desarrollo del trabajo de la tesis. En este capítulo se comienza definiendo las especificaciones de diseño para la realización de cortes controlados, se continúa con el diseño y fabricación de un efector final que provea la capacidad de sostener un bisturí a un manipulador robótico acorde con el estándar ISO 9409-1: 2004. Posteriormente, se definen conceptos tales como imagen tomada, que se refiere a una fotografía de la escena del robot sosteniendo el bisturí, entre otros. A continuación, se desarrolla el algoritmo de corte con penetración controlada. Este algoritmo, que a su vez se compone de una serie de sub-rutinas, se orienta en una primera etapa a estimar el grado de penetración del bisturí dentro de una imagen tomada y en una segunda, a calcular la señal de control que será enviada al robot.

El capítulo IV se ocupa del análisis de los resultados de la investigación y la contrastación de las hipótesis. Aquí, se indican las condiciones y consideraciones tomadas durante los experimentos de corte con bisturí. Asimismo, se exponen los resultados que fueron obtenidos de dichos experimentos, en los que se realizaron cortes con bisturí a muestras de tejido de pecho de pollo, utilizando el algoritmo de corte con penetración controlada.

CAPÍTULO I

ANTECEDENTES Y DESCRIPCIÓN DEL PROBLEMA

1.1. Antecedentes

En esta sección se muestra una revisión de artículos científicos orientados a la realización de cortes controlados, poniendo especial interés en aquellos que abordan la realización de cortes quirúrgicos con penetración controlada, sin dejar de lado otros artículos que proporcionen información útil para el desarrollo de la tesis con respecto técnicas de estimación del grado de penetración del bisturí. Los artículos revisados fueron publicados en revistas científicas o conferencias de alcance internacional.

En [31], se propone un método de control de corte de tejido blando con bisturí, basado en la estimación y control de la profundidad del corte realizado con este. La profundidad de corte fue estimada en función a la fuerza de corte, haciendo uso de una relación matemática obtenida por ajuste de datos experimentales que relacionan ambas variables. Las pruebas del método fueron realizada utilizando una plataforma móvil de 3 ejes lineales. Esta incorporó un sensor de fuerza en su efector final, al que se acopló un bisturí, tal como se puede ver en la figura 1.1 (a).

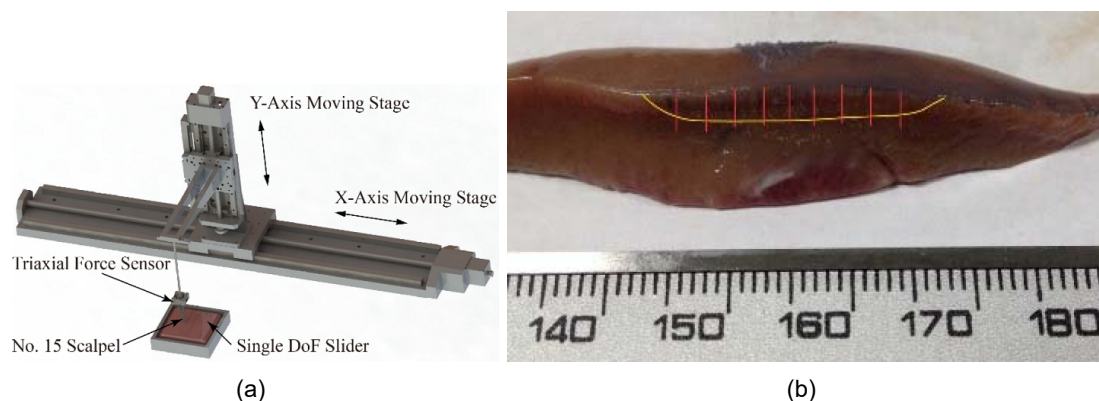


Figura 1.1: (a) Configuración experimental para la medida de la fuerza y la profundidad de corte. (b) Sección de corte de tejido blando obtenida bajo el método de control propuesto. (Fuente: Referencia [31])

La fuerza de corte fue medida en base a la información obtenida del sensor de fuerza. El tejido elegido para las pruebas fue riñón porcino, debido a su frecuente uso en experimentos de corte [7]. El método de control de corte se resume en tres pasos: primero, calcular

la fuerza de corte F_G requerida para obtener la profundidad deseada; luego, mover el bisturí hasta una posición cercana al tejido; y finalmente, descender el bisturí a una rapidez de 0.1mm/s, hasta que la fuerza registrada durante el experimento F_{RT} alcance un valor igual a F_G . Tras la implementación y pruebas del método de control de corte propuesto, se obtuvieron márgenes de error de entre 7.33 % y 41.34 %, los cuales son señalados por los autores como aceptables. En [31] se puede ver que el método propuesto no realiza un control directo de profundidad, sino un control de fuerza que parte de la existencia de una relación matemática previamente calculada entre la fuerza de corte y la profundidad de corte, para una determinada muestra de tejido. De existir otras variables que influyan en la profundidad de corte, estas representarían fuentes de error en la estimación de la misma. Además, aquella relación matemática es válida únicamente para la muestra de tejido analizada, siendo necesario realizar un nuevo análisis cada vez que se requiera cortar una nueva muestra. Cabe anotar que en [31], no se indica la técnica utilizada para medir la profundidad de corte real. Sin embargo, la figura 1.1 (b) da a suponer que esta se mide por medio del análisis de imágenes digitales.

En [1], se propone un método de control de la profundidad de corte con láser para microcirugía de tejido blando asistido por robot. Esta propuesta se resume en: realizar la incisión de acuerdo a los valores de la cantidad de pases n y la rapidez de movimiento del rayo láser, estimados para alcanzar la profundidad de corte d_{target} deseada. El cálculo se realiza en base a una relación matemática teórica propuesta y verificada en [1]. Las pruebas del método de control se hicieron empleando la plataforma de microcirugía μ RALP [8] para la realización de cortes con profundidad controlada sobre tejido de músculo de pollo con márgenes de error de entre 4.67 % y 5.8 %. Este método de control puede ser denominado control en lazo abierto, dado que no existe realimentación. Esto implica que no se tiene información, durante el la incisión, que indique si realmente la profundidad de corte alcanza el valor numérico de d_{target} .

En [7], se realiza un análisis de la respuesta a la fractura de tejido blando durante el corte con bisturí. En este, se define la respuesta a la fractura R , como la cantidad de trabajo mecánico necesario para hacer que una incisión se extienda por una unidad de longitud en una muestra de tejido blando de espesor unitario. Su estudio requirió de la medida de la fuerza de corte y de la determinación la profundidad de corte. La información necesaria para el análisis fue obtenida de experimentos de corte de tejido de hígado de cerdo, realizados utilizando una plataforma móvil de un grado de libertad. Se incorporó un sensor de fuerza al efector final de la plataforma, al que se acopló un bisturí, tal como se puede ver en la

figura 1.2. La medición de la fuerza fue realizada utilizando la información proveniente del sensor mencionado, mientras que la determinación de la profundidad de corte fue realizada en base al análisis de imágenes digitales, tomadas a 23 fotogramas por segundo, del bisturí incidiendo sobre el tejido. En estas imágenes se determinó la profundidad de corte en función a la distancia entre el centro de una marca visual rectangular ubicada sobre el mango del bisturí y el borde que separa la superficie del tejido, de la región del bisturí. Se puede ver en este procedimiento, utilizado para la determinación de la profundidad de corte, que localizar una marca visual permite reducir la región de análisis de la imagen completa, a únicamente una parte de ella que contiene la marca. Sin embargo, se puede suponer que el método es susceptible a error ocasionado por la posible presencia de objetos que agreguen bordes que se confundan con el borde de la superficie del tejido, y con ello dificulten la detección del límite que separa la superficie del tejido de la región del bisturí.

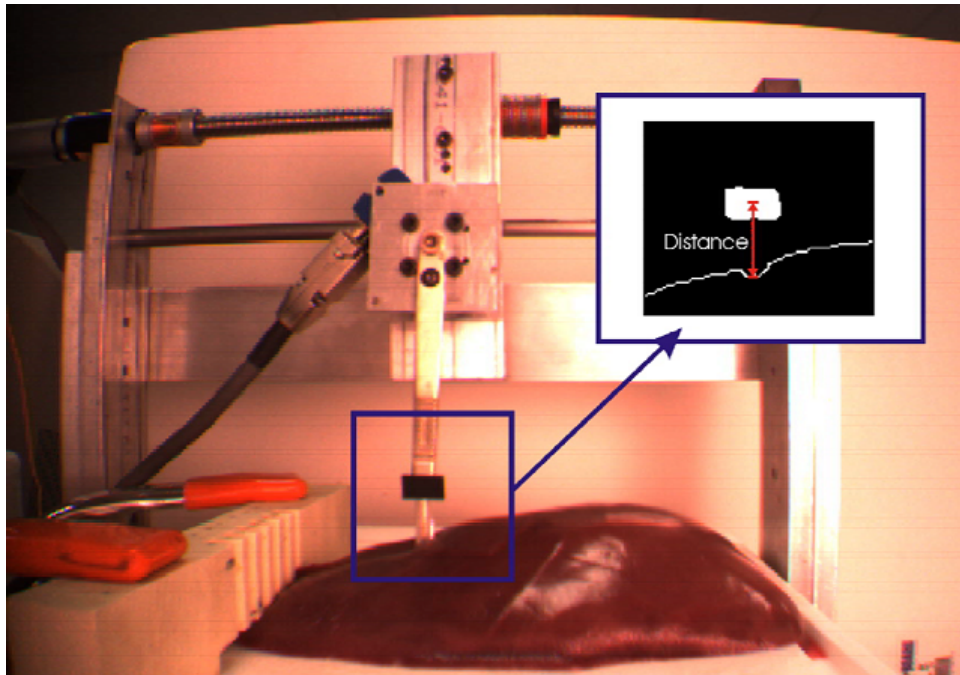


Figura 1.2: Fotografía tomada por la cámara instantánea.
(Fuente: Referencia [7])

En [10], se propone un método de control de la profundidad de corte en microcirugía láser de tejido blando. Este método se basa en el cálculo del tiempo de exposición t_{exp} , requerido para conseguir la profundidad de corte deseada, asumiendo valores previamente establecidos para los demás parámetros que intervienen en dicho cálculo. El método fue implementado en la plataforma robótica de microcirugía láser [23], para la realización de cortes con profundidad de corte controlada en muestras de tejido de pecho de pollo. En estas pruebas se presentó una raíz del error cuadrático medio de 0.12mm en incisiones del rango de hasta 1mm. La medición referencial de la profundidad de corte fue obtenida

por medio del análisis de imágenes tomadas utilizando un microscopio digital, mientras que las pruebas de corte fueron realizadas con diferentes tiempos de exposición de láser. Se consiguieron diferentes profundidades de corte, tal como se puede ver en la figura 1.3. De forma similar a [1], en [10] también se recurrió a una técnica de control en lazo abierto, dado que no existe realimentación. Por ello, no se tiene información, durante el corte, de si realmente se alcanza la profundidad de corte deseada.

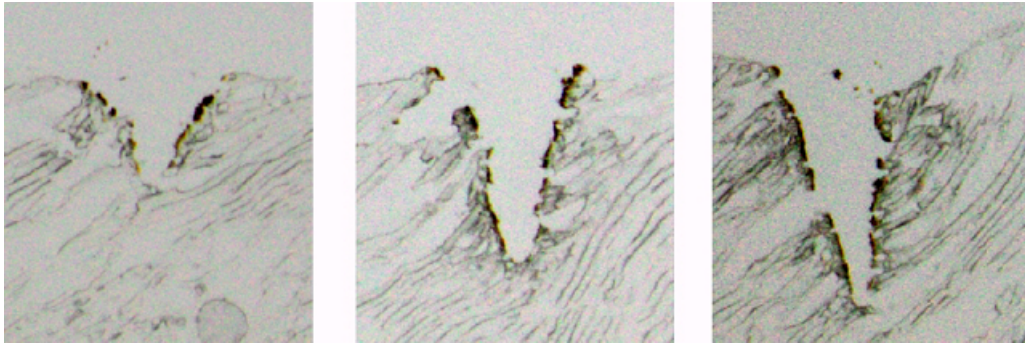


Figura 1.3: Perfiles de corte para tiempos incrementados de exposición (de izquierda a derecha: 2.5s, 3.5s, y 4.5s) y densidad de poder, frecuencia de escaneo y longitud constantes. (Fuente: Referencia [10])

En [3], se propone un efector final de un grado de libertad para cirugía, con movimiento generado por un IPMC (compuesto iónico de polímero-metal), para la realización de cortes con profundidad controlada. Una representación del efector final se puede ver en la figura 1.4, en donde la profundidad de corte D , puede ser calculada en función a la fuerza de corte F , en base a una relación matemática establecida y verificada en [3]. Ahí se mide la profundidad de corte en base al análisis de imágenes digitales tomadas por un microscopio, de perfiles de tejido cortados con diferentes valores de voltaje V aplicados al IPMC. El sistema con entrada V y salida F , es controlado utilizando un controlador PI (proporcional integral), cuyas ganancias de control proporcional e integral fueron calculadas mediante sintonización por 5 diferentes métodos, de los que se eligieron los valores de ganancias que permitieron obtener los mejores resultados en el sistema controlado. Las pruebas de corte fueron realizadas sobre muestras de silicona. Se puede ver que a pesar de que en las conclusiones de [3] se indica que el sistema muestra un rendimiento razonable, aquello no se puede contrastar dado que no se especifican valores numéricos de error. Es posible suponer que el método utilizado para medir la profundidad de corte no es afectado por fuentes de error de medición de fuerza, debido a que no utiliza información de fuerza. Sin embargo, tal como está planteado, no es aplicable durante el corte, dado que requiere imágenes de microscopio de secciones del tejido luego de haber sido cortado.

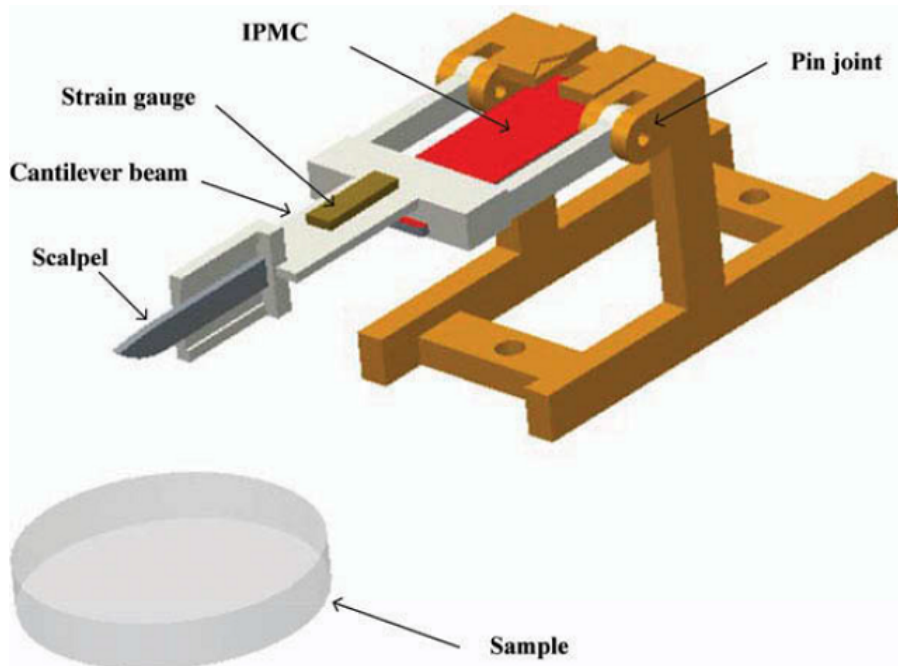


Figura 1.4: Mecanismo del efector final.
(Fuente: Referencia [3])

En [29], se propone un sistema robótico para la realización de cortes en tejido blando. El sistema incorpora el uso de ventosas que sostienen la superficie del tejido, con la finalidad de mantenerlo en una posición fija con respecto al bisturí y así poder realizar cortes más precisos. Este sistema que además consta de tres grados de libertad, dos de ellos lineales y uno angular, es representado en la figura 1.5 (a). Experimentalmente, el perfil de corte que define la profundidad y longitud de corte es obtenido por medio de la detección de bordes en imágenes de rayos-X, tomadas al tejido cortado, tal como se puede ver en la figura 1.5 (b). Las pruebas de corte se realizaron en tejido de lomo de cerdo. Se encontró que, con la utilización de ventosas, las curvas que aproximan los perfiles de corte son más consistentes y, en consecuencia, mejores. Se puede ver que el uso de rayos-X permitió obtener el perfil de corte sin necesidad de seccionar el tejido, como sí se hace cuando se analizan imágenes en el espectro de la luz visible en [10].

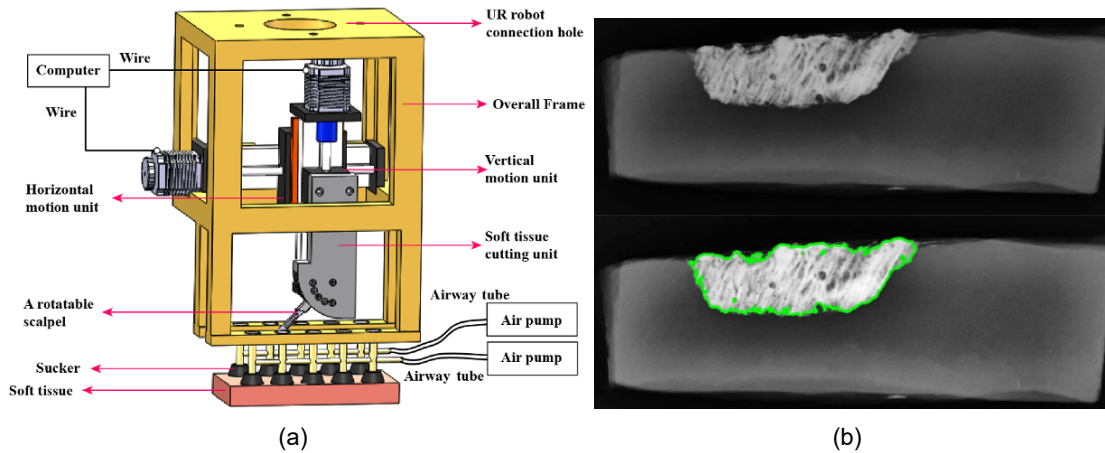


Figura 1.5: (a) Estructura del sistema robótico de corte. (b) Extracción de la curva de corte. (Fuente: Referencia [29])

En [40], se propone un método de control de corte de objetos deformables basado en visión. El método consiste en controlar el movimiento de un cuchillo para que siga una trayectoria previamente generada, con la finalidad de realizar una incisión en una superficie deformable. El control se ejerce con realimentación visual, en donde la localización y orientación del cuchillo se calculan en base a la localización de 3 marcas de color verde denominadas puntos de marca. Los parámetros del modelo de deformación del material son obtenidos en base a la localización de 7 marcas de color rojo ubicadas sobre la superficie deformable, denominadas puntos característicos. Las pruebas del método fueron para la realización de cortes en esponja, tal como se muestra en la figura 1.6. Se puede ver que el procedimiento propuesto no requiere información de fuerza y, en consecuencia, no precisa un estudio previo de la fuerza de corte como sí sucede en [31], [3]. Esto implica que el método es aplicable indistintamente al tipo de material, siempre que este pueda ser cortado por el sistema.

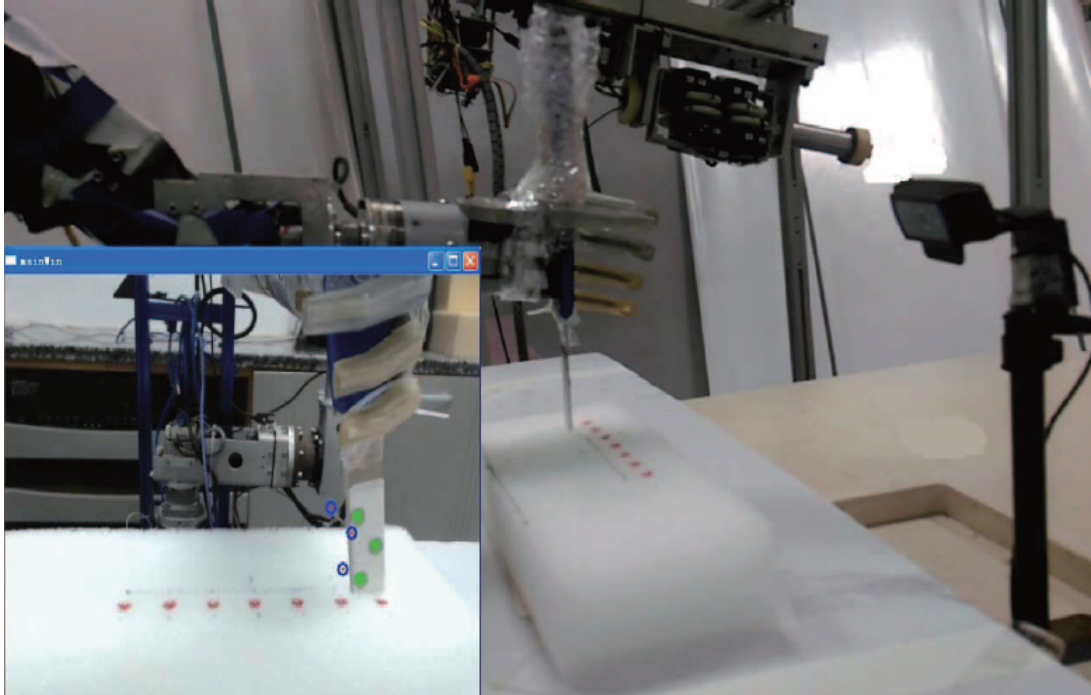


Figura 1.6: Configuración experimental.
(Fuente: Referencia [40])

En [6], se realiza un estudio de la respuesta mecánica que ofrece un tejido durante su corte. Esta respuesta es caracterizada por medio del módulo de Young efectivo local. Por su parte, este es obtenido mediante análisis de modelos de elementos finitos, en base a datos de fuerza de corte registrados haciendo uso de un sensor de fuerza. Las pruebas fueron realizadas utilizando el equipo de incisión de tejido hepático, mostrado en la figura 1.7 (a), para la realización de cortes en hígado de cerdo. Se encontró que dentro de una misma muestra de tejido, el módulo de Young efectivo local varía dependiendo de la distancia del límite restringido del espécimen, tal como se muestra en la figura 1.7 (b). Esta variación implica que la respuesta mecánica de un tejido puede cambiar en función de la posición específica en donde se esté cortando. Esto da a suponer que la realización de incisiones basada en información de fuerza requiere un estudio específico de la respuesta mecánica del tejido en las posiciones en donde se requiera cortar. Aquello no sucede cuando la incisión es realizada en base a información visual, como es el caso de [40].

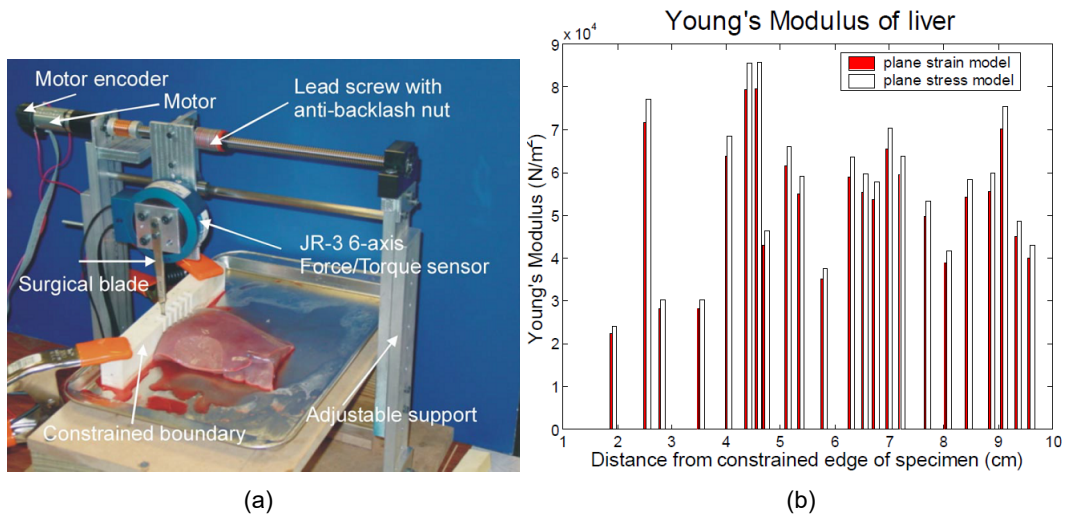


Figura 1.7: (a) Configuración experimental para medir las fuerzas de corte en hígado de cerdo. (b) Módulo de Young efectivo local para segmentos de carga de corte típico. (Fuente: Referencia [6])

En [38], se propone una plataforma mecánica para estudios biomédicos de fuerzas de corte en tejido: figura 1.8 (a). En dicho trabajo se mide la fuerza de corte en base a la información obtenida de un sensor de fuerza diseñado como una variación de [39], mientras que se calcula la profundidad de corte en base al análisis de imágenes tomadas por una cámara digital, a 4 fotogramas por segundo y 1600 por 1200 píxeles de resolución. Esta profundidad es calculada como la diferencia entre la longitud detectada de la hoja quirúrgica antes y después de que ingrese al tejido, tal como se muestra en la figura 1.8 (b). La plataforma fue probada para la realización de cortes en tejido porcino (pulmón, hígado y riñón). Se utilizó cianocrilato para pegar las muestras de tejido al porta-muestras, con la finalidad de tener los bordes más limitados y simular el entorno fisiológico de diferentes tejidos. Se puede inferir que, el pegar una muestra al porta-muestras implicaría que la superficie de esta sea más tensa y con ello menos deformable durante la incisión. Así sería menos probable que el borde que separa la región de la hoja quirúrgica y la muestra quede oculto dentro algún pliegue formado como consecuencia de la deformación de la muestra, permitiendo así la toma de imágenes del borde, por medio de la cámara digital.

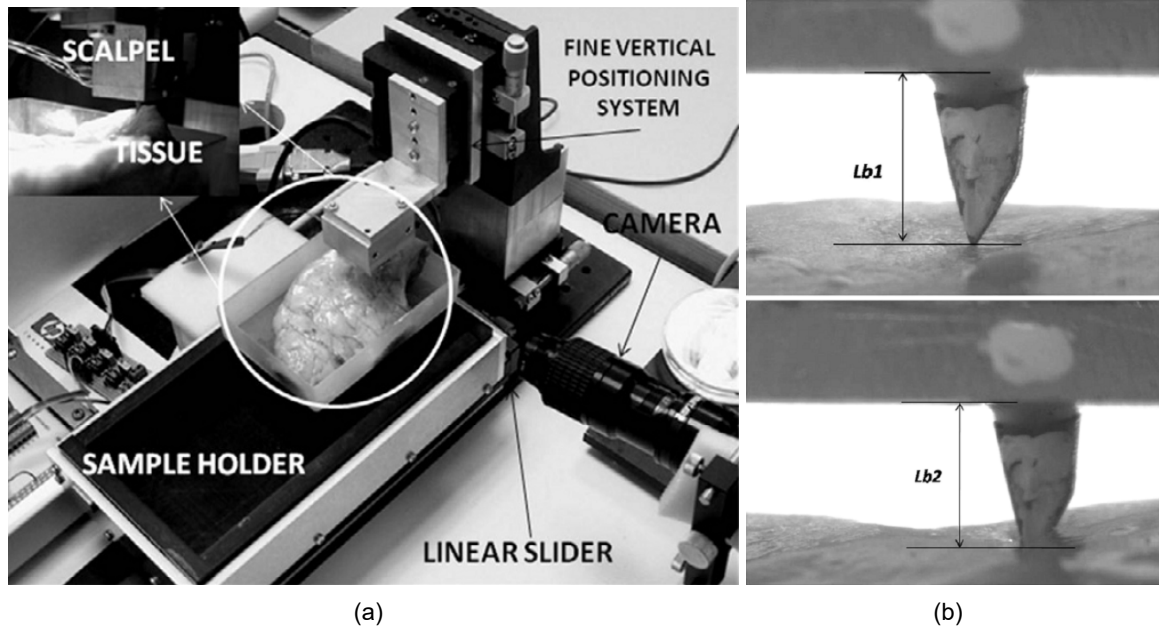


Figura 1.8: (a) Configuración experimental. (b) Fotografía para la estimación de la profundidad de corte usando procesamiento de imágenes, para una profundidad de $L_{b1} - L_{b2}$.
(Fuente: Referencia [38])

En [21], se propone una estrategia de control de corte robotizado para seccionar materiales blandos deformables, minimizando la fuerza de corte. Este método se basa en el uso de una plataforma de corte conformada por dos robots para la realización de una incisión a lo largo de una trayectoria curva. El primer robot, denominado robot de corte, se encarga de hacer la incisión, empleando un cuchillo. El segundo, denominado robot de tirado, se encarga sostener un extremo del material con la finalidad de aplicar fuerzas externas que faciliten la separación de la muestra, tal como se puede ver en la figura 1.9. La estrategia de control utiliza información de fuerza y torque, proveniente de sensores ubicados en los robots, además de información geométrica de la superficie deformable del material. Esta información es obtenida en base a la localización de marcas visuales colocadas sobre el material, a partir del uso de imágenes tomadas por una cámara digital, acoplada al robot de corte. Las pruebas fueron realizadas utilizando como muestra un bloque de espuma. Si bien el desarrollo de esta estrategia, descrita en [21], enfoca su atención al análisis cinemático del sistema, se puede ver que ha sido provechoso el uso de información visual y, específicamente, la localización de marcas visuales para la caracterización geométrica.

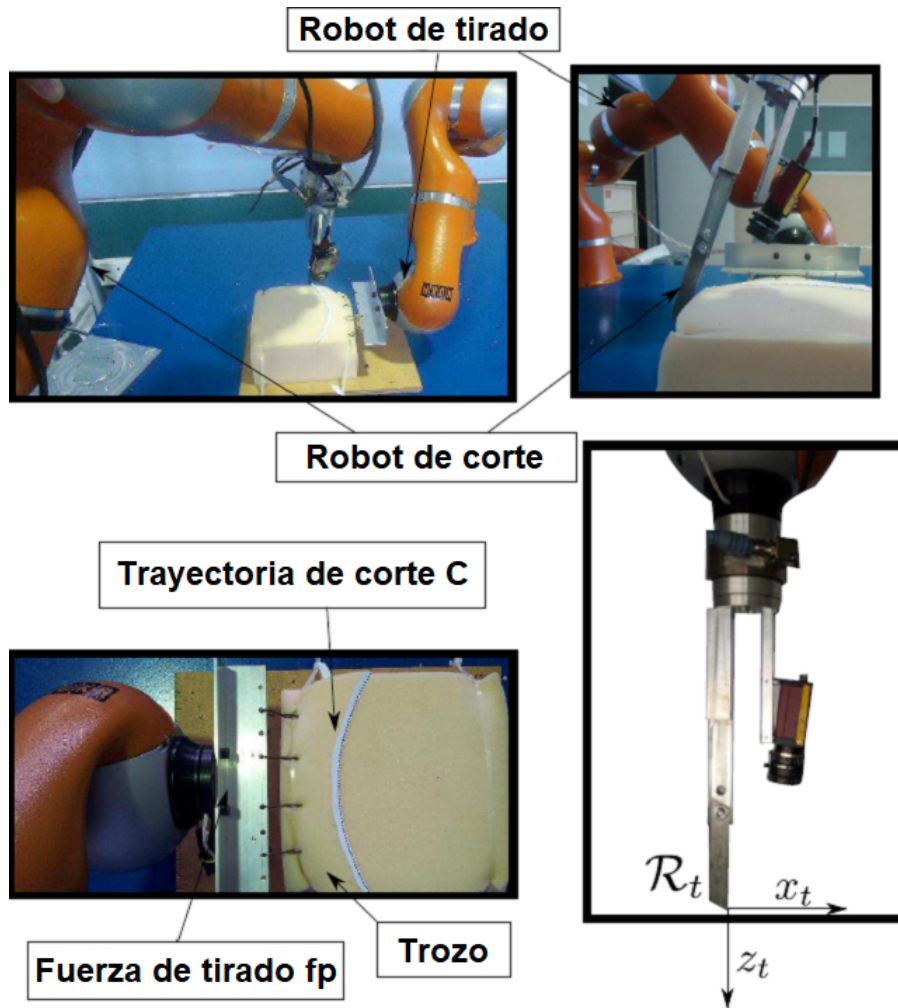


Figura 1.9: Plataforma experimental.
(Fuente: Referencia [21])

1.2. Descripción de la realidad problemática

El desarrollo de sistemas robóticos para la realización de cortes quirúrgicos controlados ha sido abordado en el ámbito comercial, como es el caso del robot Da Vinci [15], así como también en el ámbito de investigación, como en los ejemplos: [31], [7], [29], [6]. En diferentes trabajos se sostiene la premisa de que, normalmente, la realización de cortes quirúrgicos depende de la habilidad particular adquirida por medio de la experiencia de algún cirujano en específico. Esto implica que la falta de aquella práctica sea un problema para otros cirujanos, tal como se indica en [1]. Además, aquella dependencia, implica problemas con la precisión y repetitividad de los cortes, tal como se señala en [29].

En algunos de los trabajos que abordan la realización de cortes quirúrgicos controlados empleando bisturí, se busca una relación matemática entre la profundidad de corte y la

fuerza de corte [31], [3], de modo que luego pueda utilizarse aquella relación para estimar la profundidad de corte en función a información medida de la fuerza de corte. Sin embargo, la respuesta mecánica, de la que depende la fuerza de corte, no necesariamente es constante, incluso dentro de una misma muestra de tejido, tal como se evidencia en [6]. Aquella respuesta puede variar en función de la posición de la muestra que se esté cortando. Esto implica que sea necesario realizar un nuevo estudio de la relación entre la profundidad de corte y la fuerza de corte, cada vez que se trabaje con una nueva muestra o incluso si la incisión se hiciera en diferentes lugares de la misma. En otros trabajos, tales como [7] y [38], se realiza análisis de imágenes digitales para estimar la profundidad de corte como la diferencia entre la longitud detectada de la hoja del bisturí antes y después de que este penetre en el tejido, mientras que en otros casos, a pesar de que no se detalla el uso de algún algoritmo de análisis de imágenes, se da a entender que este se utiliza para el cálculo de la profundidad de corte, tal como se expone en [31], [3].

El presente trabajo de tesis se desarrolla como parte de una de las actividades que conforman el proyecto identificado con código 144-2018-FONDECYT/BM, llevado a cabo por la Universidad de Ingeniería y Tecnología (UTEC [37]) y financiado por Concytec - Fondecyt. Esta actividad, aborda además la propuesta de una técnica de control para la realización de cortes con penetración controlada, hecho que ha sido estudiado en [31], [1], y orienta su implementación a la realización de cortes verticales con bisturí en muestras de tejido blando de superficie plana, empleando un manipulador robótico, motivo por el que además fue necesario el diseño y construcción de un efector final que provea la capacidad de sostener un bisturí al manipulador robótico utilizado. Cabe resaltar que, para este trabajo se utilizó un manipulador robótico de dos grados de libertad, de los que únicamente fue necesario controlar uno, para la generación de movimiento vertical en el efector final.

1.3. Formulación del problema

En el presente trabajo se busca dar respuesta a la pregunta: ¿es posible diseñar un algoritmo de corte con penetración controlada con bisturí, empleando un manipulador robótico?

1.4. Justificación e importancia de la investigación

El presente trabajo de tesis se justifica desde la necesidad de un algoritmo de corte con penetración controlada con bisturí, empleando un manipulador robótico. Esto representa un aporte al desarrollo de un sistema para la realización de cortes quirúrgicos controlados que

no dependa de la habilidad particular adquirida por medio de la experiencia de un cirujano en específico (hecho que ocasiona inconvenientes para otros cirujanos [1], y problemas de precisión y repetitividad de los cortes [29]).

1.5. Objetivos

1.5.1. Objetivo general

- Diseñar un algoritmo de corte con penetración controlada con bisturí empleando un manipulador robótico, para la realización de cortes en muestras de tejido blando de superficie plana.

1.5.2. Objetivos específicos

- Diseñar e implementar un efector final para un manipulador robótico que le provea la capacidad de sostener un bisturí.
- Crear una base de datos de imágenes para el desarrollo de algoritmos, en la que las imágenes simulen cortes quirúrgicos con bisturí, a diferentes profundidades.
- Desarrollar e implementar, en un programa de computadora, un algoritmo de detección de píxeles de color, en base al análisis de imágenes en los espacios de color RGB y HSV.
- Desarrollar e implementar en un programa de computadora, un algoritmo de estimación de la profundidad de corte con bisturí en base al algoritmo de detección de píxeles de color.
- Seleccionar una técnica de control de la profundidad de corte con bisturí, entre la bibliografía existente, priorizando el empleo de controladores PI, PID, difuso.
- Desarrollar e implementar en un programa de computadora, un algoritmo de corte con penetración controlada en base al uso combinado del algoritmo de estimación de la profundidad de corte, con la técnica de control de la profundidad de corte seleccionada.

1.6. Hipótesis

1.6.1. Hipótesis principal

- Es posible diseñar un algoritmo de corte con penetración controlada con bisturí, empleando un manipulador robótico, para la realización de cortes en muestras de tejido blando de superficie plana.

1.6.2. Hipótesis secundarias

- Es posible diseñar un efector final para un manipulador robótico que le provea la capacidad de sostener un bisturí.
- El posible crear una base de datos de imágenes para el desarrollo de algoritmos, en la que las imágenes simulen cortes quirúrgicos con bisturí, a diferentes profundidades.
- El análisis de imágenes en los espacios de color RGB y HSV permitirá desarrollar un algoritmo de detección de píxeles de color.
- El empleo del algoritmo de detección de píxeles de color permitirá desarrollar un algoritmo de estimación de la profundidad de corte con bisturí.
- La revisión bibliográfica permitirá seleccionar una técnica de control de la profundidad de corte, priorizando el empleo de controladores PI, PID, difuso.
- El uso combinado del algoritmo de estimación de la profundidad de corte, con la técnica de control de la profundidad de corte, permitirá desarrollar un algoritmo de corte con penetración controlada.

1.7. Variables e indicadores

Variable Independiente

- Algoritmo de corte con bisturí empleando un manipulador robótico.

Indicadores de la Variable Independiente

- Capacidad del efector final para acoplarse a un manipulador robótico y sostener un bisturí.
- Región detectada del bisturí.

Variable Dependiente

- Corte con penetración controlada.

Indicadores de la Variable Dependiente

- Profundidad de corte.
- Ciclos de ejecución por segundo del bucle principal del programa de corte con bisturí.
- Coeficiente de correlación de Pearson entre las profundidades de corte controladas y sus respectivas referencias de control.

1.8. Unidad de análisis

Imagen digital de 1280 por 720 píxeles de resolución.

1.9. Tipo de investigación

El presente trabajo de tesis corresponde a una investigación de tipo aplicada, debido a que busca responder a una necesidad de la práctica quirúrgica. Esta es la necesidad de un algoritmo de corte con penetración controlada con bisturí que emplee un manipulador robótico. Para el desarrollo de la tesis, se diseñó un efector final que se acopló un manipulador robótico y le proveyó la capacidad de sostener un bisturí. Seguidamente, se desarrolló un algoritmo de estimación de la profundidad de corte. Aquel, posteriormente, derivó a un algoritmo de corte con bisturí con penetración controlada. Este algoritmo fue implementado en un programa de computadora y se probó satisfactoriamente para la realización de cortes quirúrgicos, empleando un manipulador robótico.

1.10. Nivel de investigación

El nivel de investigación del presente trabajo de tesis corresponde a un nivel de maestría en ciencias, dado que emplea conocimientos correspondientes al estudio de la misma tales como; inteligencia artificial, procesamiento de señales, e ingeniería de control.

Asimismo, el tema propuesto no ha sido abordado en la literatura científica, por lo cual contribuye con lo siguiente: desarrollo e implementación de un nuevo algoritmo de corte con penetración controlada y la construcción de un efector final para realizar cortes controlados.

1.11. Periodo de análisis

El presente trabajo de tesis se desarrolló dentro de un periodo de 12 meses.

1.12. Fuentes de información e instrumentos utilizados

Para el desarrollo del algoritmo de corte con bisturí se utilizaron imágenes digitales y videos de 1280 por 720 píxeles de resolución tomadas por una cámara de video Sony Handycam FDR-AX40 mientras que el efector final que sostuvo el bisturí fue acoplado a un manipulador robótico. Las pruebas de funcionamiento del algoritmo se realizaron en base a información de video de 1280 por 720 píxeles de resolución, capturada durante los experimentos de corte utilizando la cámara de video. Todas las operaciones por computadora se efectuaron en una computadora Asus Rog Strix g15 con microprocesador Intel(R) Core(TM)

i5-10300H, utilizando el software de programación Matlab R2021a y Microsoft Visual Studio 2019, para el desarrollo y pruebas del algoritmo de corte con bisturí respectivamente. Así mismo, se empleó un microscopio digital de fabricante genérico modelo 1600X para la medición de la profundidad de corte.

1.13. Técnicas de recolección y procesamiento de datos

La información utilizada para el desarrollo del algoritmo de corte con bisturí se compuso de imágenes digitales y videos, tomados manualmente empleando la cámara de video. Las pruebas de funcionamiento del algoritmo fueron realizadas por medio del análisis de la información de video, tomada y procesada en tiempo real, durante la realización de cortes quirúrgicos. Todos los datos fueron procesados empleando software funcionando en una computadora personal.

CAPÍTULO II

MARCO TEÓRICO Y CONCEPTUAL

2.1. Espacio de color RGB

De acuerdo a [14], el espacio de color RGB (Red Green Blue) es una representación de colores, en la que se parte de que es posible producir un determinado espectro de color como una combinación de los espectros de color de tres colores básicos: rojo, verde, y azul. Este espacio de color es considerado como modelo base para la mayoría de aplicaciones, dado que en él una imagen adquirida no necesita alguna transformación para mostrarse en una pantalla [11]. En este espacio de color, un color específico puede ser representado por tres componentes numéricos correspondientes a los canales rojo (R), verde (G), y azul (B), de modo que:

- r : Valor numérico en el canal R.
- g : Valor numérico en el canal G.
- b : Valor numérico en el canal B.

El rango de valores numéricos de r , g y b depende de los tipos de datos que se utilicen. Se puede tomar valores reales en el intervalo de $[0,1]$ o valores enteros en el intervalo de 0 a 255 [27], según sea el caso.

2.2. Espacio de color HSV

De acuerdo a [14], el espacio de color HSV (Hue Saturation Value) es una representación de colores utilizada en el análisis de imágenes para procesos de segmentación, en el que un determinado espectro de color es caracterizado por sus componentes en los canales de matiz (H), saturación (S), y valor (V), de modo que:

- h : Valor numérico en el canal H es la pureza del color y se puede representar como el ángulo de inclinación de un vector con origen en el centro de una sección circular.
- s : Valor numérico en el canal S es el grado en que el blanco está incrustado en un color específico. Se puede representar dentro de una sección circular de cono, como la división entre la distancia de un punto de la sección circular hacia el centro y el radio de aquella sección circular.

- v : Valor numérico en el canal V corresponde al brillo y se puede representar dentro de una sección circular de cono, como la distancia entre el centro de la sección circular y el vértice del cono.

2.3. Conceptos de lógica difusa

Los conceptos de lógica difusa descritos en esta sección, fueron extraídos de [20].

2.3.1. Conjunto crisp

Se denomina conjunto crisp a un conjunto cuya función de pertenencia μ_A mapea todos sus elementos en un conjunto universal X para al conjunto $\{0, 1\}$, lo que significa que un elemento únicamente puede pertenecer o no pertenecer a un conjunto crisp. De modo que:

$$\mu_A : X \rightarrow \{0, 1\} \quad (2.1)$$

2.3.2. Conjunto difuso

Se denomina conjunto difuso a aquel conjunto cuya función de pertenencia μ_A mapea todos sus elementos en un conjunto universal X al intervalo $[0, 1]$, lo que significa que un elemento puede pertenecer, no pertenecer o pertenecer parcialmente a un conjunto difuso. En consecuencia, un conjunto difuso es un “conjunto de límites vagos” en comparación con un conjunto crisp, de modo que:

$$\mu_A : X \rightarrow [0, 1] \quad (2.2)$$

2.3.3. Implicación difusa

Se define una relación difusa R como:

$$R = A \rightarrow B \quad (2.3)$$

En donde R puede ser vista como un conjunto difuso con una función de pertenencia bi-dimensional.

$$\mu_R(x, y) = f(\mu_A(x), \mu_B(y)) \quad (2.4)$$

Donde “ f ” es llamada “función de implicación difusa”, y hace el trabajo de transformar los grados de pertenencia de “ x ” en “ A ” y de “ y ” en “ B ”, en sus correspondientes (x, y) en $A \times B$. A continuación se muestran dos tipos de implicación difusa conocidos.

(1) Implicación difusa de regla de operación mínima (Mamdani): Se interpreta la implicación difusa como la operación “mínimo”, de modo que:

$$R_C = A \times B = \int_{X \times Y} \mu_A(x) \wedge \mu_B(y) / (x, y) \quad (2.5)$$

Donde; \wedge es el operador mínimo.

(2) Implicación difusa de regla de operación producto (Larsen): Se interpreta la implicación difusa como la operación “producto”, de modo que:

$$R_C = A \times B = \int_{X \times Y} \mu_A(x) \cdot \mu_B(y) / (x, y) \quad (2.6)$$

Donde; \cdot es el operador algebraico de producto.

2.4. El coeficiente de correlación de Pearson

El concepto de coeficiente de correlación de Pearson descrito en esta sección fue extraído de [19].

El coeficiente de correlación de Pearson es una medida de la asociación lineal de dos variables. Los valores de este varían de -1 a 1. Los valores positivos indican una correlación positiva, es decir, la tendencia a que el valor de una de las variables se incremente o decremente junto al valor de la otra. Por su parte, los valores negativos indican correlación negativa, es decir, la tendencia a que el incremento del valor de una de las variables se asocie al decremento del valor de la otra variable y viceversa. Los valores cercanos a 0 indican baja correlación y los valores cercanos a -1 o +1 indican una fuerte correlación entre las dos variables.

De acuerdo a [35], el valor del coeficiente de correlación de Pearson (r) puede ser calculado utilizando la ecuación (2.7).

$$r = \frac{n \sum xy - \sum x \sum y}{\sqrt{n \sum x^2 - (\sum x)^2} \sqrt{n \sum y^2 - (\sum y)^2}} \quad (2.7)$$

Donde; x e y son los valores de las variables, cuya correlación es evaluada y n es el tamaño de la muestra.

CAPÍTULO III

DESARROLLO DE LA TESIS

3.1. Especificaciones de diseño para la realización de cortes controlados

3.1.1. Definición del sistema dinámico a controlar

En el presente trabajo de tesis, se consideró como plataforma de control a la computadora personal utilizada para ejecutar el algoritmo de corte con penetración controlada (algoritmo de la figura 3.58). Por su parte, el sistema a controlar, en adelante denominado SRobot, estuvo conformado por el robot que ejerce los cortes verticales, el efector final empleado para sostener el bisturí y el bisturí.

Este sistema (SRobot) tiene como entrada la referencia de rapidez angular enviada al servomotor, que genera el movimiento vertical del efector final, y como salida tiene el grado de penetración, que es medido por la profundidad de corte del bisturí.

3.1.2. Especificaciones de los cortes a realizar

El presente trabajo únicamente aborda la realización de cortes verticales con bisturí, enfocándose exclusivamente en el grado de penetración de dichos cortes, el mismo que es medido por la profundidad de corte que alcanzan y se calcula de acuerdo a la ecuación (3.1). En este trabajo se dejó de lado el análisis del perfil de la incisión dejada por la hoja bisturí sobre la muestra, por no ser objeto de este estudio.

Asimismo, el algoritmo de corte con penetración controlada (algoritmo de la figura 3.58) desarrollado en el presente trabajo de tesis, no depende del número de hoja de bisturí empleado. Sin embargo, se infiere que el uso de hojas de bisturí muy pequeñas dificultaría su percepción por parte de la cámara de video, implicando así una reducción en la precisión de la estimación de la profundidad de corte. Se resalta que en este trabajo se utilizaron únicamente hojas de bisturí número 10.

El algoritmo de corte con penetración controlada no limita su aplicación a un modelo específico de robot, requiriendo fundamentalmente que este tenga la capacidad de realizar cortes verticales en base a referencias de rapidez angular del servomotor que genera

movimiento vertical. Aun así, se entiende que el grado de precisión de los movimientos controlados del robot repercutirá en el grado de precisión de los cortes controlados.

3.1.3. Especificaciones de control

El presente trabajo de tesis no se orientó a proponer un algoritmo de corte con penetración controlada óptimo o con características específicas de la variable controlada en el estado transitorio. En tal sentido, no se consideró necesario realizar un estudio profundo asociado al tiempo de asentamiento de la variable controlada, a la sobreelongación máxima porcentual de la misma o a minimizar algún funcional de costo.

Sin embargo, la elección de una técnica de control para formar parte del algoritmo de corte con penetración controlada (abordada en la sección 3.11), implicó la necesidad establecer criterios que permitan evaluar las alternativas de técnica de control a fin de elegir una.

Siendo así, en este trabajo se considera que una técnica de control funciona satisfactoriamente si la respuesta de un sistema controlado con dicha técnica, ante una entrada de tipo escalón, cumple con las siguientes características:

- 1 : t_s (el tiempo de asentamiento) es menor o igual a 90 segundos.
- 2 : M_p (la sobreelongación máxima porcentual) es menor o igual al 25 %.

Cabe resaltar que, si bien en [26], t_s y M_p se obtienen en base a la respuesta del sistema dinámico ante una entrada de tipo escalón unitario, aplicar una entrada de tipo escalón unitario al algoritmo de corte con penetración controlada, en este trabajo, significaría establecer una referencia de profundidad de corte (entrada del sistema controlado) de 1 mm. Sin embargo, experimentalmente, se encontró que surgen problemas al realizar cortes con referencias de profundidad de corte menores a 5 mm, tal como se indica en la sección 4.3. Debido a ello, en adelante, t_s y M_p serán obtenidos en base a la respuesta del sistema controlado ante una entrada de tipo escalón de 5 mm de amplitud.

Asimismo, t_s será el tiempo requerido para que la curva de respuesta alcance un rango en el intervalo de $\pm 5\%$ del valor final.

3.2. Desarrollo del efector final

En esta sección se describe el procedimiento seguido para diseñar y fabricar un efector final capaz de acoplarse a un manipulador robótico y proveerle a este la capacidad de

sostener un bisturí.

Si bien el producto del presente trabajo de tesis no orienta su aplicación a un modelo de manipulador robótico en específico, se consideró oportuno que el efector final sea diseñado con la capacidad de acoplarse a un manipulador robótico comercial. Para ello, se eligió realizar el acople de acuerdo al estándar internacional ISO 9409-1:2004 [16], aplicable a interfaces mecánicas para manipuladores robóticos industriales.

3.2.1. Diseño del efector final

El efector final consta de tres componentes principales y dos agarraderas. En la figura 3.1 se muestran dos vistas renderizadas del primer componente denominado acople general. Este componente se sujeta directamente al manipulador robótico por medio de cuatro tornillos, dispuestos de acuerdo con el patrón de acoplamiento ISO 9409-1-50-4-M6.

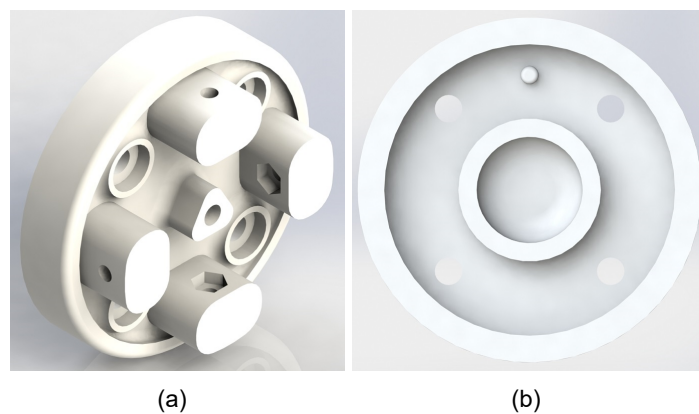


Figura 3.1: Vistas renderizadas isométrica (a), y posterior (b) del acople general.

En la figura 3.2 se muestran dos vistas renderizadas del segundo componente, denominado base del efector final. Este componente tiene como finalidad sujetarse al acople general por medio de cuatro tornillos y a su vez sujetar al tercer componente. Adicionalmente, la pieza incluye dos ranuras de uso opcional, las mismas que pueden ser utilizadas para acoplar una linterna o una cámara digital, según se requiera.

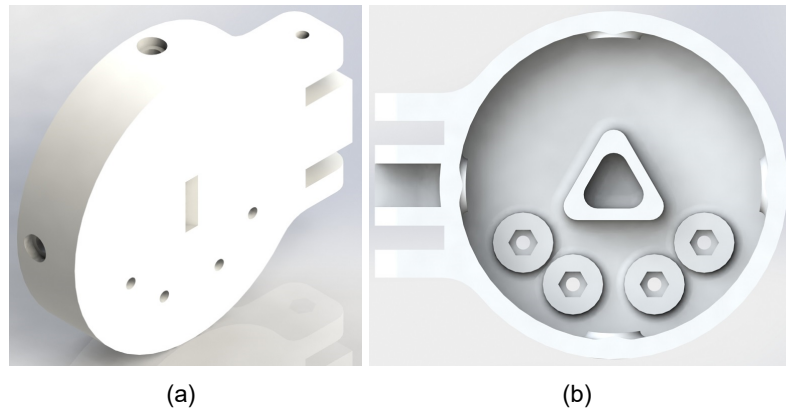


Figura 3.2: Vistas renderizadas isométrica (a), y posterior (b) de la base del efector final.

En la figura 3.3 se muestran dos vistas renderizadas del tercer componente denominado soporte de bisturí. Este componente se sujeta a la base del efector final por medio de cuatro tornillos y tiene como finalidad sujetar al bisturí con ayuda de las dos agarraderas, las que se sujetan a él por medio de dos tornillos cada una.

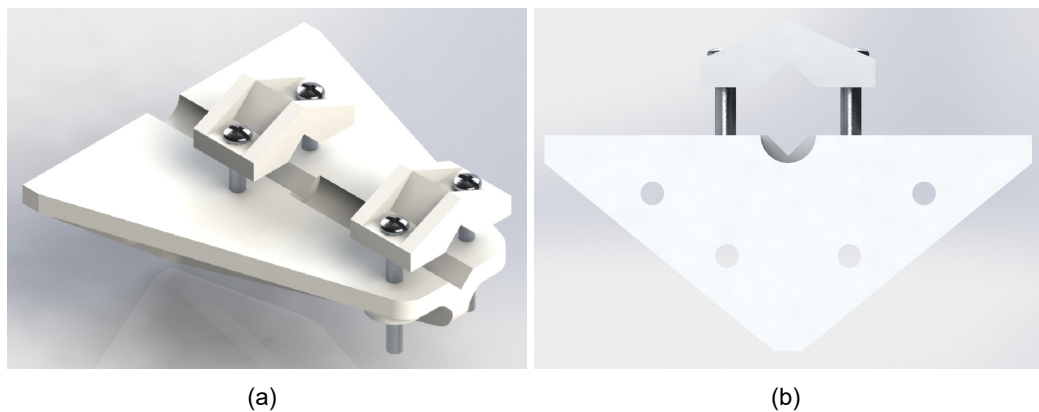


Figura 3.3: Vistas renderizadas isométrica (a), y posterior (b) del soporte del bisturí con dos agarraderas.

En la figura 3.4 se muestran tres vistas renderizadas del efector final diseñado.

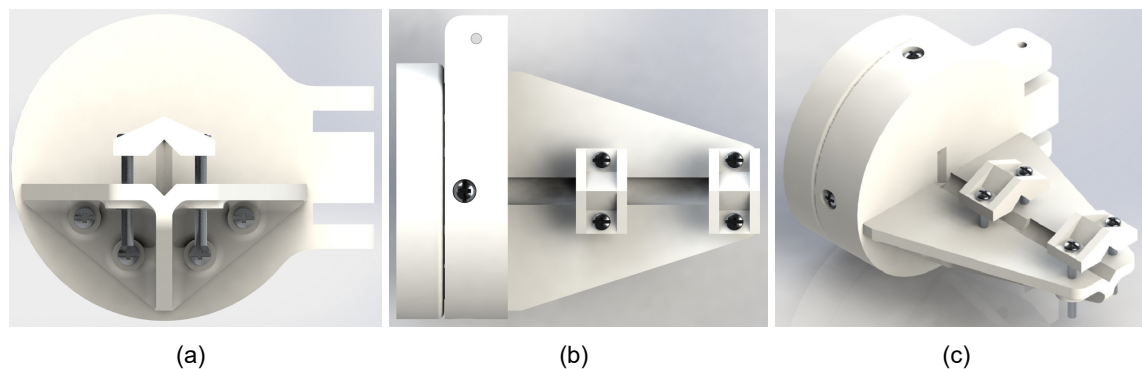


Figura 3.4: Vistas renderizadas frontal (a), superior (b), e isométrica (c) del efector final.

3.2.2. Diseño de la plataforma de muestras

Se desarrolló una plataforma física conformada por componentes diseñados por computadora y fabricados en material PLA, por medio de impresión 3D. Así mismo, se utilizó, como elemento base de la plataforma, una tabla de madera.

La plataforma diseñada, en adelante denominada plataforma de muestras, tiene como finalidad servir como recipiente para colocar las muestras, estando estas sujetas ella mediante hilos. Para tal fin, la plataforma cuenta con agujeros dispuestos a lo largo de la misma, tal como se muestra en la figura 3.5 (a). Así mismo, en la figura 3.5 (b), se muestra la plataforma de muestras fabricada.

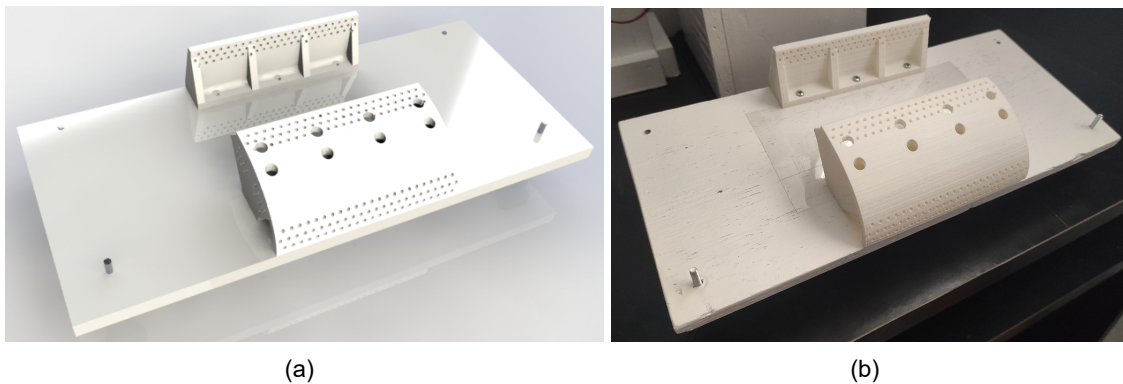


Figura 3.5: (a) Vista renderizada de la plataforma de muestras. (b) Plataforma de muestras.

3.2.3. El robot MK2 modificado

Si bien el efector final fue diseñado con la capacidad de acoplarse a un manipulador robótico, de acuerdo con el estándar ISO 9409-1:2004, durante el desarrollo del presente trabajo de tesis, no se dispuso de algún manipulador robótico comercial que cumpla con aquel estándar. Por ello, se optó por trabajar en base a un manipulador robótico basado en un robot de hardware abierto que permitiera generar movimiento vertical en el efector final para así poder realizar los cortes verticales controlados. Con este fin, se eligió utilizar el manipulador robótico EEZYbotARM MK2 [9], debido a que cumple con lo requerido y no necesita de equipo altamente sofisticado o costoso para su fabricación. Este robot se muestra en la figura 3.6. Cabe resaltar que, si bien el diseñador del robot proporciona los modelos en 3D de los elementos que componen la estructura mecánica del artefacto, no se precisa el modelo de servomotores a utilizar y tampoco se especifica la plataforma electrónica para controlar el robot. Se deja estos aspectos a elección de la persona que decida fabricar su diseño.



Figura 3.6: Robot EEZYbotARM MK2.
(Fuente: Referencia [9])

En el presente trabajo se utiliza una versión modificada del robot EEZYbotARM MK2, denominada en adelante como MK2 modificado, cuya estructura mecánica se muestra en forma de vista renderizada en la figura 3.7 (b). En esta versión se omite el uso de la base original del EEZYbotARM MK2, dado que esta no aporta al movimiento vertical. Asimismo, se reemplaza el gancho por un adaptador para el acople del efector final, el mismo que es mostrado en la figura 3.7 (a). El robot MK2 modificado cuenta únicamente con dos grados de libertad. Uno de ellos permite el movimiento vertical del efector final y el otro, la regulación del acercamiento de este efector hacia una cámara de video ubicada con línea de vista perpendicular al plano de la hoja del bisturí.

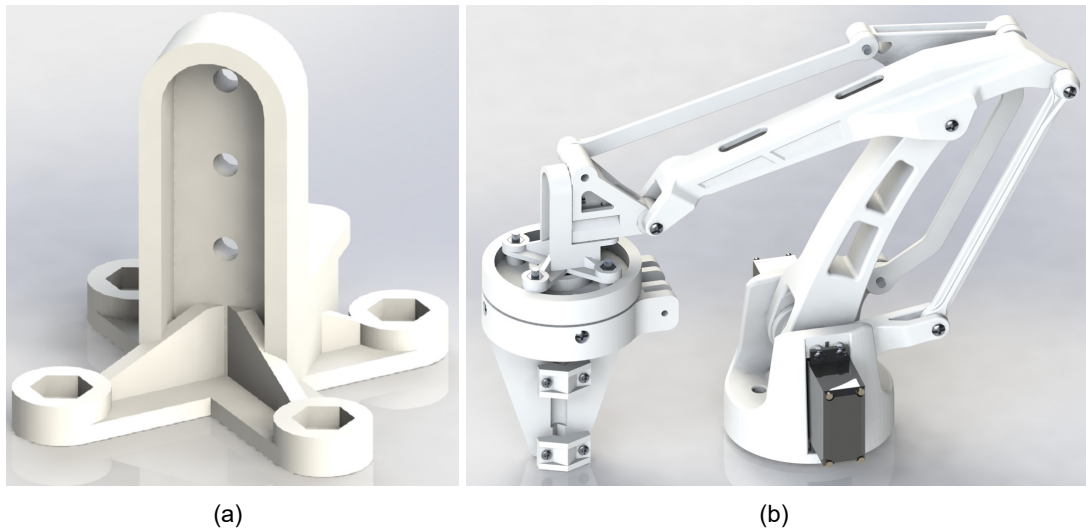


Figura 3.7: (a) Vista renderizada del adaptador para el acople del efector final. (b) Vista renderizada de la estructura mecánica del robot MK2 modificado, con el efector final acoplado.

Se eligió utilizar servomotores modelo MG996R para cumplir la función de actuadores del robot MK2 modificado. Así mismo, estos son alimentados con DC 5V y cuentan, cada uno, con un controlador de posición angular que opera en base a referencias, que son in-

gresadas en forma de señales de tipo PWM [13], de acuerdo con la información técnica del servomotor MG996R [2].

Se denomina servomotor 1 y $ang1$, al servomotor que genera movimiento vertical del efector final y a su variable de referencia de posición angular respectivamente. Del mismo modo, se denomina servomotor 2 y $ang2$ al servomotor que genera movimiento horizontal del efector final y a su variable de referencia de posición angular respectivamente.

En la figura 3.8 se muestra el robot MK2 modificado fabricado, con el efector final acoplado.

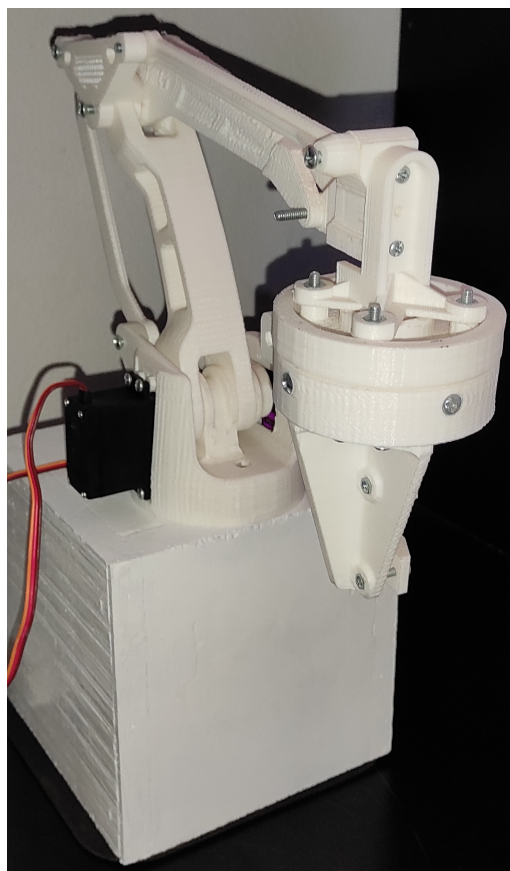


Figura 3.8: Robot MK2 modificado, con el efector final acoplado.

Es importante resaltar que, el autor del robot EEZYbotARM MK2 [9] no detalla características técnicas del mismo, tales como espacio de trabajo, capacidad de carga, entre otras. En consecuencia, tampoco se conocen aquellas características para el robot MK2 Modificado, al ser este una variación del robot EEZYbotARM MK2. Sin embargo, experimentalmente, el robot MK2 modificado fue capaz de generar movimiento vertical en el efector final dentro un rango aproximado de 50mm, soportando, en todo momento, el peso del efector final con

el bisturí acoplado más el acople (véase la figura 3.7 (b)), representando estos una carga de 120 gramos.

3.2.4. Interface electrónica de operación del robot MK2 modificado

El desarrollo de una interfaz electrónica de operación del robot MK2 modificado no forma parte de los objetivos del presente trabajo de tesis. Sin embargo, su realización fue necesaria a fin de poder operar al robot en base a órdenes provenientes de una computadora personal. En este trabajo se considera a dicha interfaz como parte del robot MK2 modificado.

La interfaz fue desarrollada como una tarjeta electrónica basada en el microcontrolador PIC18F4550 [24]. Esta tarjeta genera referencias de posición angular para la operación de los servomotores, mientras recibe órdenes provenientes de una computadora personal, a través de su puerto de comunicación en serie.

Cada orden está compuesta de un byte, denominado $data_{in}$, en donde el primer bit representa la variable denominada cmd_{in} , mientras que los siguientes 7 bits, representan la variable $param_{in}$. La lógica que rige el cálculo de las referencias de posiciones angulares $ang1$ y $ang2$, en función a sus valores anteriores, es descrita en el algoritmo de la figura 3.9.

Entrada: $data_{in}, ang1, ang2, Ts$

Salida: $ang1, ang2$

```
1:  $cmd_{in} :=$  obtener el primer bit de  $data_{in}$ 
2:  $param_{in} :=$  obtener los últimos 7 bits de  $data_{in}$ 
3: si  $param_{in} < 64$  entonces
4:   si  $cmd_{in} = 0$  entonces
5:      $ang1 := ang1 - Ts \cdot param_{in}/5$ 
6:   fin si
7:   si  $cmd_{in} = 1$  entonces
8:      $ang1 := ang1 + Ts \cdot param_{in}/5$ 
9:   fin si
10: si no
11:   si  $cmd_{in} = 0$  entonces
12:      $ang1 := 30 + (param_{in} - 64) \cdot 2$ 
13:   fin si
14:   si  $cmd_{in} = 1$  entonces
15:      $ang2 := 30 + (param_{in} - 64) \cdot 2$ 
16:   fin si
17: fin si
```

Figura 3.9: Algoritmo; generar $ang1$ y $ang2$.

Donde:

T_s : Tiempo de muestreo del bucle de control de movimiento del robot.

$ang1$: Referencia de posición angular para el servomotor 1.

$ang2$: Referencia de posición angular para el servomotor 2.

El diseño de la tarjeta electrónica es mostrado en el anexo B. Así mismo, el algoritmo de la figura 3.9 fue implementado en el programa para el microcontrolador PIC18F4550, detallado en [33]. Dicho programa fue redactado en el lenguaje de programación *C* y compilado en el entorno de desarrollo PCW [5]. Cabe resaltar que, durante los experimentos se procuró que las rapidezces de giro de los ejes de los servomotores se mantuvieran en el intervalo de $[-2, 2]$ dps (grados sexagesimales por segundo). Esto se debió a que trabajar con dicho intervalo permitió obtener resultados satisfactorios en el control de la profundidad de corte, resultando innecesario utilizar rapidezces de giro de mayor valor absoluto.

3.3. Análisis de color

El desarrollo de un algoritmo de corte con penetración controlada implica poder conocer aquella profundidad. Para ello se recurrió a [38], en donde dicha profundidad es calculada como la diferencia entre la longitud total de la hoja del bisturí L_{d1} y la del lado expuesto instantáneamente L_{b2} , tal como se indica en ecuación (3.1).

$$pc = L_{d1} - L_{d2} \quad (3.1)$$

Partiendo de que el problema de la estimación de la profundidad de corte puede ser resuelto en base a inspección visual, tal como se realiza en, [31], [1], [7], el presente trabajo de tesis aborda dicho problema por medio del análisis de imágenes digitales. En ese sentido, resulta fundamental poder detectar la región del bisturí dentro de una imagen digital, para así poder calcular L_{d1} y L_{d2} y con ello resolver la ecuación (3.1).

Es importante resaltar que parte del criterio empleado en el presente trabajo de tesis, para detectar la región del bisturí, fue introducido en [34].

En este trabajo se eligió limitar el área de búsqueda de la región del bisturí a una sub-imagen denominada imagen reducida, la misma que es obtenida utilizando tres marcas referenciales denominadas marcas de color (véase la sección 3.3.4)).

El análisis de una imagen digital para obtener la imagen reducida y posteriormente en ella, la región del bisturí, implicó comenzar realizando las definiciones de; imagen tomada,

píxel de color, región de color, píxel oscuro, y nivel de negro. Estas definiciones se muestran a continuación.

3.3.1. Imagen tomada

En el presente trabajo, se denomina imagen tomada a una fotografía realizada por medio de una cámara de video y representada en el espacio de color RGB. Toda imagen tomada se caracteriza por haberse realizado con una línea de vista perpendicular al plano al que pertenece la hoja del bisturí, así como también se caracteriza por mostrar las marcas de color (véase la sección 3.3.4) alineadas verticalmente. En la figura 3.10 se muestra un ejemplo de imagen tomada.

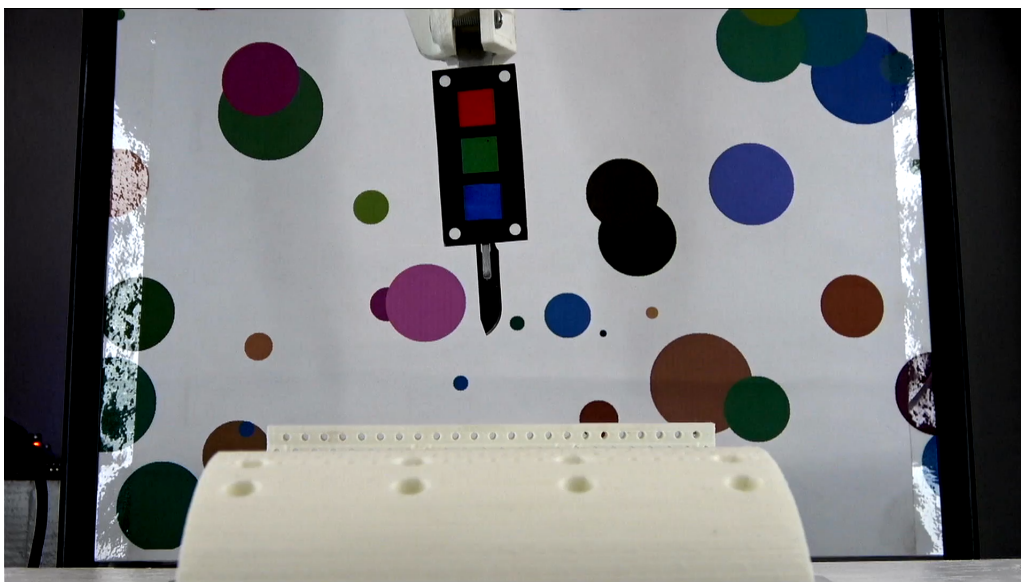


Figura 3.10: Imagen tomada.

3.3.2. Píxel de color y región de color

En el presente trabajo, se define un píxel de color como todo aquel cuyo aspecto visual frente al ojo humano es reconocido como uno de los colores rojo, verde o azul. Esta definición se acompaña de una caracterización matemática, la misma que es descrita en la sección 3.3.8. Así mismo, se denomina tipo de color de un píxel de color, al color reconocido en aquel píxel de color. De esta manera, se identifican píxeles de color de tipo rojo, verde, o azul. Cabe resaltar que, según sea el caso, se propone sobreentender la expresión "tipo de color", siendo así equivalente decir "píxel de color de tipo rojo", a decir "píxel de color rojo".

Se define una región de color, como un grupo de píxeles de color del mismo tipo, adyacentes entre sí.

3.3.3. Píxel negro

En el presente trabajo, se define como píxel negro a todo aquel cuyo aspecto visual sea oscuro. Estos píxeles (negros) tiene como característica que sus componentes numéricos en los canales R, G, y B, se encuentran por debajo de un umbral bl , siendo dicho umbral denominado nivel de negro e ingresado manualmente por el usuario.

La definición de píxel negro tiene como finalidad evitar el innecesario análisis de píxeles cuyo aspecto visual sea demasiado oscuro como para poder ser reconocidos visualmente como píxeles de color, considerándolos únicamente como píxeles negros y así descartando la posibilidad de que llegaran a ser considerados como píxeles de color. Por tal motivo, en el momento en el que un píxel entre en la denominación píxel negro, este ya no podrá ser denominado píxel de color.

3.3.4. La paleta de colores

Se define la paleta de colores como una imagen rectangular impresa físicamente y compuesta por cuatro círculos blancos dispuestos cerca de los vértices de la misma. Esta imagen con fondo negro contiene en su interior tres regiones de color cuadradas de iguales dimensiones y dispuestas verticalmente. Cada una de estas es de un tipo de color diferente y dos de ellas se ubican a igual distancia de la región de color central. Estas regiones se denominan marcas de color, y se nombran según su tipo de color, teniendo de esta forma la marca de color roja, verde y azul. La disposición de las tres marcas de color se realiza de modo que la marca de color roja se ubique por encima de la verde y esta a su vez se ubique por encima de la azul.

Es importante mencionar que los círculos blancos tienen como finalidad delimitar una región rectangular, dentro de la que se encuentren las marcas de color, denominada región interior de la paleta de colores. Así mismo, el fondo negro tiene como finalidad diferenciarse claramente de las marcas de color, facilitando así la detección de estas dentro de la paleta de colores.

En la figura 3.11 se muestra una representación gráfica de una paleta de colores.

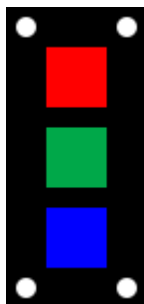


Figura 3.11: Representación gráfica de una paleta de colores.

La paleta de colores tiene como objetivo proporcionar los dos tipos de información descritos a continuación:

Información de color : Esta se obtiene en base al análisis de los componentes numéricos de los píxeles de color detectados en la paleta de colores, en los espacios de color RGB y HSV.

Información geométrica : Esta se obtiene en base al análisis de la información de posición, y sus derivados, obtenidos de los píxeles de color detectados en la paleta de colores.

3.3.5. Obtención de la región interior de la paleta de colores

En esta sección se describe el procedimiento seguido para la obtención de la región interior de la paleta de colores. Este procedimiento se basa en el hecho de que dicha región es delimitada por los círculos blancos contenidos en la paleta de colores. Este procedimiento, además, permite calcular las constantes *bas* y *has* de la imagen reducida (véase la sección 3.8.3). En tal sentido, requiere el señalamiento manual de 5 puntos referenciales por parte del usuario, los cuales son:

p1 : Centro del círculo blanco ubicado en la zona superior izquierda de la paleta de colores.

p2 : Centro del círculo blanco ubicado en la zona inferior derecha de la paleta de colores.

p3 : Centro de la marca de color verde.

p4 : Punta de del bisturí.

p5 : Punto que señala el límite hasta donde la hoja del bisturí puede ingresar verticalmente a la muestra.

Este procedimiento es descrito en el algoritmo de la figura 3.12. En donde se tienen las siguientes entradas y salidas:

IMG : Imagen tomada.

spalette : Sub-imagen que representa la región interior de la paleta de colores.

bas : Longitud de la base de la imagen reducida.

has : Longitud referencial para la altura de la imagen reducida.

dis : Distancia entre el centro de la marca de color verde y el punto medio entre la punta del bisturí y el límite hasta donde la hoja del bisturí puede ingresar verticalmente a la muestra.

p1 : Posición del centro del círculo blanco ubicado en la zona superior izquierda de la paleta de colores.

f_3 : Factor de relación entre *l2* y *has*.

f_{31} : Factor de relación entre *bas* y *has*.

Entrada: *IMG*, f_3 , f_{31}

Salida: *palette*, *bas*, *has*, *dis*, *p1*

1: $p1, p2, p3, p4, p5 :=$ puntos

2: *state* := 0

3: **mientras** *state* <= 4 **hacer**

4: *IMG* := obtener fotograma de video

5: *IMG* := aplicar filtro media a *IMG*

6: *pCursor* := obtener posición del cursor

7: **si** (*state* = 0) ∧ se oprimió el cursor **entonces**

8: *p1* := *pCursor*

9: *state* := 1

10: **fin si**

11: **si** (*state* = 1) ∧ se oprimió el cursor **entonces**

12: *p2* := *pCursor*

13: *state* := 2

14: **fin si**

15: **si** (*state* = 2) ∧ se oprimió el cursor **entonces**

16: *p3* := *pCursor*

17: *state* := 3

18: **fin si**

19: **si** (*state* = 3) ∧ se oprimió el cursor **entonces**

20: *p4* := *pCursor*

21: *state* := 4

22: **fin si**

23: **si** (*state* = 4) ∧ se oprimió el cursor **entonces**

24: *p5* := *pCursor*

25: *state* := 5

26: **fin si**

27: **fin mientras**

28: $l1 := \sqrt{(p4(1) - p3(1))^2 + (p4(2) - p3(2))^2}$

29: $l2 := \sqrt{(p5(1) - p4(1))^2 + (p5(2) - p4(2))^2}$

30: *has* := $l2 \cdot f_3$

31: *bas* := has / f_{31}

32: *dis* := $l1 - l2/2$

33: *palette* := Obtener la sub-imagen rectangular de *IMG* con vértices en *p1* y *p2*.

Figura 3.12: Algoritmo; obtener la paleta de colores.

En el presente trabajo de tesis no se establece algún criterio matemático para el cálculo f_3 y f_{31} . Sin embargo, experimentalmente, se encontraron buenos resultados considerando aquellos valores iguales a 1.3 y 1.5, respectivamente.

Cabe relatar que, tal como se puede ver en el algoritmo de la figura 3.12, la constante has se obtiene multiplicando la distancia entre $p4$ y $p5$ por f_3 . En ese sentido, has es la distancia entre la punta del bisturí y el límite hasta donde la hoja del bisturí puede ingresar verticalmente a la muestra, multiplicada por f_3 .

3.3.6. Detección de los píxeles de color en la paleta de colores

Al ser la paleta de colores diseñada de modo de que las marcas de color sean fácilmente diferenciadas del fondo. Se encontró, experimentalmente, que un píxel de color dentro de la paleta de colores cumple con las siguientes características:

1. El valor numérico del canal correspondiente a su tipo de color es superior a los valores numéricos de los otros dos canales en el espacio de color RGB.
2. El valor numérico del canal correspondiente a su tipo de color es superior a f_0 veces el valor numérico de alguno de los otros dos canales en el espacio de color RGB, siendo f_0 una constante ingresada por el usuario y obtenida mediante experimentación.

En el presente trabajo de tesis no se establece algún criterio matemático para el cálculo f_0 . Sin embargo, experimentalmente, se encontraron buenos resultados considerando aquel valor igual 1.5 para detectar píxeles de color rojos, 1.2 para detectar píxeles de color azules y 1.4 para detectar píxeles de color verdes. Valores que son empleados en el algoritmo de la figura 3.16.

En la figura 3.13 (b) se muestra una representación visual de los píxeles de color detectados en la paleta de colores, obtenidos en base a las dos características mencionadas.

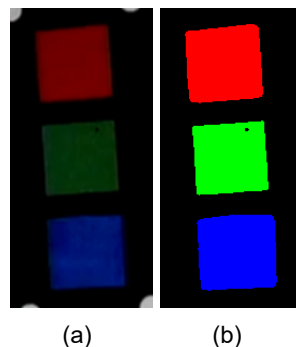


Figura 3.13: (a) Región interior de la paleta de colores. (b) Píxeles de color rojos, verdes y azules, detectados en la paleta de colores.

3.3.7. Análisis de la paleta de colores

El primer objetivo de la paleta de colores es proporcionar información de color. Por ello, resulta necesario analizar la información numérica que se pueda extraer de los píxeles de color detectados en la paleta de colores. Para esto, se eligió representar gráficamente los componentes numéricos de los diferentes canales de los píxeles de color en los espacios de color RGB y HSV. Esta representación se muestra en la figura 3.14.

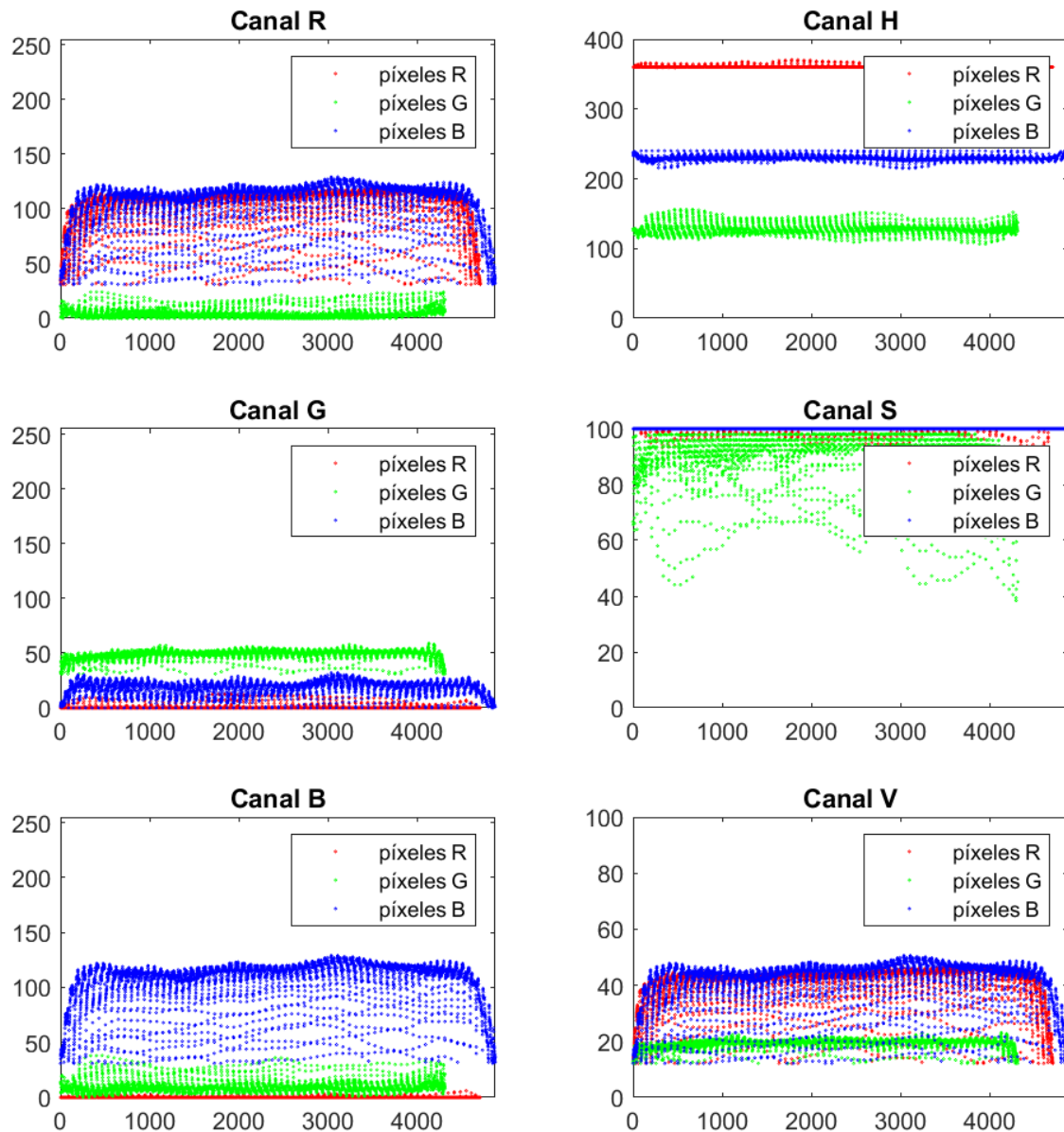


Figura 3.14: Representación de los valores numéricos de los píxeles de color de la paleta de colores en los canales R, G, B, H, S, y V. En los 6 sub-gráficos, el eje horizontal representa la cantidad de píxeles de color detectados en la paleta de colores, mientras que el eje vertical representa el valor numérico de cada píxel de color encontrado en la misma, en los canales R, G, B, H, S, y V.

Es importante resaltar que al ser la paleta de colores un objeto físico, los valores numéricos extraídos de los píxeles de color, detectados en ella, pueden variar dependiendo de factores tales como las condiciones de luz, características del sensor óptico de la cámara de video, entre otros.

El segundo objetivo de la paleta de colores es proporcionar información geométrica. Esta información es representada por las áreas de las marcas de color, las longitudes de los lados de las marcas de color y las varianzas de posición de los píxeles que componen las marcas de color, en los ejes horizontal y vertical. En ese sentido se definen las siguientes constantes:

RA_m, GA_m, BA_m : Áreas estimadas de las marcas de color roja, verde y azul, respectivamente. Estas se calculan como la cantidad de píxeles de color detectados de cada tipo en la paleta de colores.

RL_m, GL_m, BL_m : Longitudes estimadas de lados de las marcas de color roja, verde y azul, respectivamente. Estas se calculan como la raíz cuadrada del área estimada de cada marca de color.

RV_m, GV_m, BV_m : Varianzas estimadas de posición en los ejes horizontal y vertical (son iguales en ambos ejes) de las marcas de color roja, verde y azul, respectivamente. Para su cálculo se toma en cuenta que para un conjunto de puntos dispuestos uniformemente distribuidos sobre una región cuadrada, tal como se muestra en la figura 3.15, es posible calcular la varianza de posición de los píxeles en el eje horizontal de acuerdo a la ecuación (3.2).

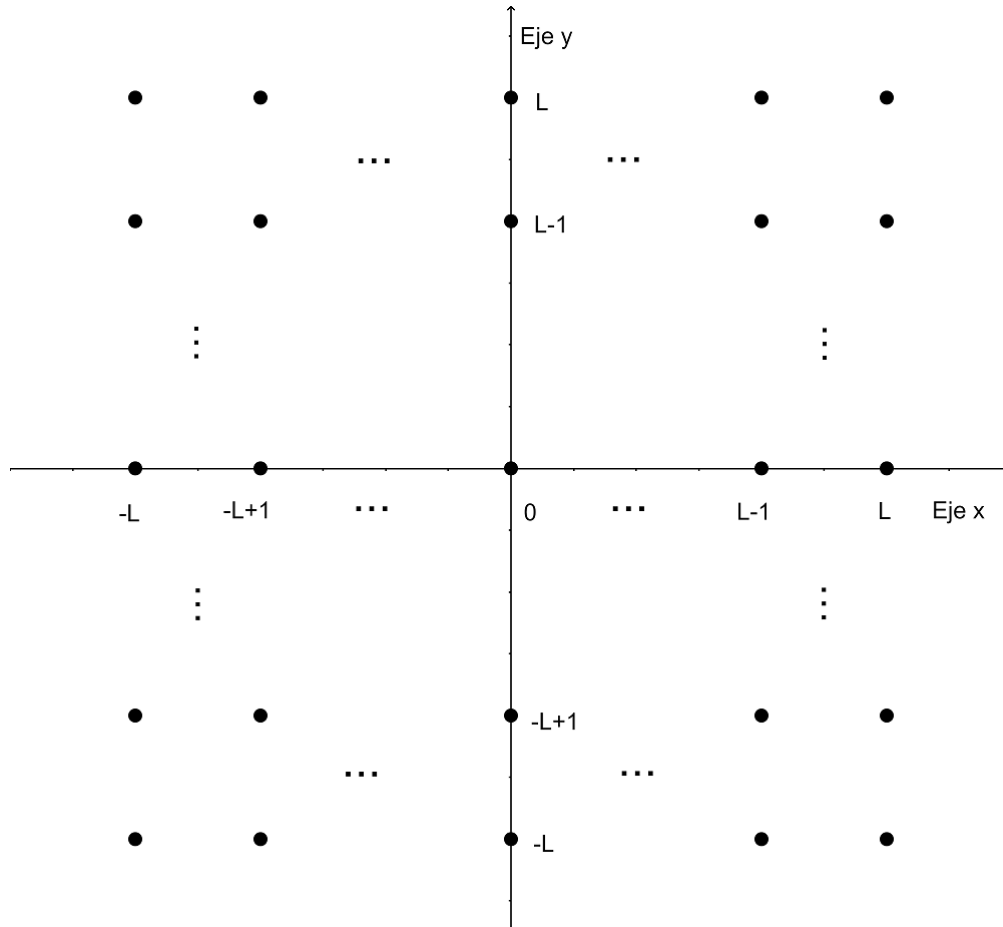


Figura 3.15: Región cuadrada con puntos.

$$\begin{aligned}
 Var &= (2L + 1) \frac{1}{n} \sum_{-L}^L (x - \bar{x})^2 = \frac{2L + 1}{(2L + 1)^2} 2 \sum_1^L x^2 \\
 &= 2 \frac{(2L + 1)}{(2L + 1)^2} \left(\frac{L(L + 1)(2L + 1)}{6} \right) = \frac{L(L + 1)}{3}
 \end{aligned} \tag{3.2}$$

Cabe resaltar que, si bien la ecuación (3.2) se utiliza para calcular la varianza en el eje horizontal. Al estar los puntos dispuestos uniformemente distribuidos en una región cuadrada, la varianza en el eje vertical será numéricamente igual a la del eje horizontal.

Para fines prácticos, siempre que se refiera a únicamente de un tipo de marca de color y se sepa claramente de que tipo de marca de color se trata, se optó por utilizar los siguientes términos.

A_m : Área estimada de la marca de color.

L_m : Longitud estimada del lado de la marca de color.

V_m : Varianza estimada de posición en los ejes horizontal y vertical de la marca de color.

En el algoritmo de la figura 3.16 se describe el procedimiento seguido para la detec-

ción de los píxeles de color contenidos en la región interior de la paleta de colores, además de la extracción de la información de color y geométrica requerida para la detección de los píxeles de color por el criterio de umbrales de color (véase la sección 3.3.8).

El algoritmo de la figura 3.16 tiene las siguientes entradas y salidas:

spalette : Sub-imagen de la región interior de la paleta de colores.

bl : Nivel de Negro.

f₁ : Factor de tolerancia de umbrales máximos.

R_p, G_p, B_p : Estructuras de parámetros comunes de los píxeles de color rojos, verdes y azules, respectivamente.

Los campos de las estructuras *R_p, G_p, y B_p*, son los siguientes:

A_m : Área estimada de la marca de color del tipo de color al que se refiere la estructura.

L_m : Longitud estimada del lado de la marca de color del tipo de color al que se refiere la estructura.

V_m : Varianza estimada de posición en los ejes horizontal y vertical de la marca de color del tipo de color al que se refiere la estructura.

H : Vector de valores numéricos en el canal H de los píxeles de color del tipo de color al que se refiere la estructura.

S : Vector de valores numéricos en el canal S de los píxeles de color del tipo de color al que se refiere la estructura.

V : Vector de valores numéricos en el canal V de los píxeles de color del tipo de color al que se refiere la estructura.

IDX_f : Vector de componentes en filas de las posiciones de los píxeles de color del tipo de color al que se refiere la estructura.

IDX_c : Vector de componentes en columnas de las posiciones de los píxeles de color del tipo de color al que se refiere la estructura.

h_{min} : Valor mínimo contenido en *H*.

h_{max} : Valor máximo contenido en *H*.

s_{min} : Valor mínimo contenido en *S*.

s_{max} : Valor máximo contenido en *V*.

Entrada: *palette*, *bl*, f_1
Salida: *Rp*, *Gp*, *Bp*, *LRp*, *LGp*, *LBp*

- 1: *Rp*, *Gp*, *Bp* := estructuras
- 2: *fil* := cantidad de filas de *palette*
- 3: *col* := cantidad de columnas de *palette*
- 4: *LRp*, *LGp*, *LBp* := puntos
- 5: *Rp.Am* := 0; *Gp.Am* := 0; *Bp.Am* := 0
- 6: **para** $i \leftarrow 1$, *fil* **hacer**
- 7: **para** $j \leftarrow 1$, *col* **hacer**
- 8: *r* := obtener componente en R del píxel *palette*(*i*, *j*)
- 9: *g* := obtener componente en G del píxel *palette*(*i*, *j*)
- 10: *b* := obtener componente en B del píxel *palette*(*i*, *j*)
- 11: *h* := obtener componente en H del píxel *palette*(*i*, *j*)
- 12: *s* := obtener componente en S del píxel *palette*(*i*, *j*)
- 13: *v* := obtener componente en V del píxel *palette*(*i*, *j*)
- 14: **si** $(r > g) \wedge (r > b) \wedge (r > 1,5g \vee r > 1,5b) \wedge (r > bl)$ **entonces**
- 15: *Rp.Am* := *Rp.Am* + 1
- 16: *Rp.H*(*Rp.Am*) := *h*, *Rp.S*(*Rp.Am*) := *s*, *Rp.V*(*Rp.Am*) := *v*
- 17: *Rp.IDXf*(*Rp.Am*) := *i*, *Rp.IDXc*(*Rp.Am*) := *j*
- 18: **fin si**
- 19: **si** $(g > r) \wedge (g > b) \wedge (g > 1,2r \vee g > 1,2b) \wedge (g > bl)$ **entonces**
- 20: *Gp.Am* := *Gp.Am* + 1
- 21: *Gp.H*(*Gp.Am*) := *h*, *Gp.S*(*Gp.Am*) := *s*, *Gp.V*(*Gp.Am*) := *v*
- 22: *Gp.IDXf*(*Gp.Am*) := *i*, *Gp.IDXc*(*Gp.Am*) := *j*
- 23: **fin si**
- 24: **si** $(b > r) \wedge (b > g) \wedge (b > 1,4r \vee b > 1,2g) \wedge (b > bl)$ **entonces**
- 25: *Bp.Am* := *Bp.Am* + 1
- 26: *Bp.H*(*Bp.Am*) := *h*, *Bp.S*(*Bp.Am*) := *s*, *Bp.V*(*Bp.Am*) := *v*
- 27: *Bp.IDXf*(*Bp.Am*) := *i*, *Bp.IDXc*(*Bp.Am*) := *j*
- 28: **fin si**
- 29: **fin para**
- 30: **fin para**
- 31: *Am* := (*Rp.Am* + *Gp.Am* + *Bp.Am*)/3
- 32: *Rp.Lm* := $\sqrt{Rp.Am}$, *Gp.Lm* := $\sqrt{Gp.Am}$, *Bp.Lm* := $\sqrt{Bp.Am}$
- 33: *Rp.Vm* := (*Rp.Lm*/2)((*Rp.Lm*/2) + 1)/3
- 34: *Gp.Vm* := (*Gp.Lm*/2)((*Gp.Lm*/2) + 1)/3
- 35: *Bp.Vm* := (*Bp.Lm*/2)((*Bp.Lm*/2) + 1)/3
- 36: *Rp.hmin* := (mínimo de *Rp.H*) f_1 , *Rp.hmax* := (máximo de *Rp.H*)/ f_1
- 37: *Rp.smin* := (mínimo de *Rp.S*) f_1 , *Rp.vmin* := (mínimo de *Rp.V*) f_1
- 38: *Gp.hmin* := (mínimo de *Gp.H*) f_1 , *Gp.hmax* := (máximo de *Gp.H*)/ f_1
- 39: *Gp.smin* := (mínimo de *Gp.S*) f_1 , *Gp.vmin* := (mínimo de *Gp.V*) f_1
- 40: *Bp.hmin* := (mínimo de *Bp.H*) f_1 , *Bp.hmax* := (máximo de *Bp.H*)/ f_1
- 41: *Bp.smin* := (mínimo de *Bp.S*) f_1 , *Bp.vmin* := (mínimo de *Bp.V*) f_1
- 42: *LRp*(1) := mediana de *Rp.IDXf*, *LRp*(2) := mediana de *Rp.IDXc*
- 43: *LGp*(1) := mediana de *Gp.IDXf*, *LGp*(2) := mediana de *Gp.IDXc*
- 44: *LBp*(1) := mediana de *Bp.IDXf*, *LBp*(2) := mediana de *Bp.IDXc*

Figura 3.16: Algoritmo; analizar la paleta de colores.

3.3.8. Detección de píxeles de color por el criterio de umbrales de color

En esta sección se propone un criterio general para la detección de píxeles de color, en base a la información de color extraída de los píxeles de color detectados en la paleta de colores, la misma que se muestra en la figura 3.14. En esta imagen se puede ver que los valores numéricos de dichos píxeles en los canales R, G, B, H, S, y V se encuentran dentro de intervalos acotados por umbrales mínimos y máximos. En la figura 3.14 se puede apreciar que los intervalos más estrechos y separados se dan en el canal H, mientras que en los canales S y V los intervalos son más anchos e incluso parcialmente superpuestos. Por su parte, se puede ver que los intervalos, en todos los canales del espacio de color RGB, son anchos y parcial o totalmente superpuestos.

Los hechos antes mencionados sugieren que es posible caracterizar los píxeles de color de forma más precisa al analizar en el espacio de color HSV y centrar el análisis en el canal H. Sin embargo, se considera poco prudente descartar la información proveniente de los canales S y V, dado que ello haría imposible diferenciar colores puros de opacos y colores brillantes de apagados. Esto daría paso a que píxeles muy opacos u oscuros puedan ser detectados como píxeles de color, a pesar de no ser visualmente identificables como tales. Es deseable descartar aquellos píxeles.

En ese sentido, se eligió establecer como criterio general para la detección de píxeles de color que todo píxel cuyo componente en el canal H se encuentre dentro del intervalo $\langle hmin, hmax \rangle$, al mismo tiempo de que su componente en el canal S se encuentre dentro del intervalo $\langle smin, \infty \rangle$ y su componente en el canal V se encuentre dentro del intervalo $\langle vmin, \infty \rangle$, será considerado píxel de color. Los valores de $hmin$, $hmax$, $smin$, $vmin$, dependen del canal que se esté analizando y estos valores son obtenidos utilizando el algoritmo de la figura 3.16.

Para fines prácticos, este criterio pasará a ser denominado criterio de umbrales de color. El procedimiento de aplicación del criterio de umbrales de color es descrito en el algoritmo de la figura 3.17 y tiene las siguientes entradas y salidas:

IMG : Imagen tomada.

Rp, Gp, Bp : Estructuras de parámetros comunes de los píxeles de color rojos, verdes y azules, respectivamente.

IMG_{RGB} : Imagen de píxeles de color detectados.

Entrada: IMG, Rp, Gp, Bp

Salida: IMG_{RGB}

```
1:  $fil :=$  cantidad de filas de  $IMG$ 
2:  $col :=$  cantidad de columnas de  $IMG$ 
3:  $IMG_{RGB} := IMG$ 
4: para  $i \leftarrow 1, fil$  hacer
5:   para  $j \leftarrow 1, col$  hacer
6:      $h :=$  obtener componente en H del píxel  $IMG(i, j)$ 
7:      $s :=$  obtener componente en S del píxel  $IMG(i, j)$ 
8:      $v :=$  obtener componente en V del píxel  $IMG(i, j)$ 
9:      $c := 0$ 
10:    si  $(h > Rp.hmin) \wedge (h < Rp.hmax) \wedge (s > Rp.smin) \wedge (v > Rp.vmin)$  entonces
11:      píxel  $IMG_{RGB}(i, j) :=$  píxel de color rojo
12:       $c := 1$ 
13:    fin si
14:    si  $(h > Gp.hmin) \wedge (h < Gp.hmax) \wedge (s > Gp.smin) \wedge (v > Gp.vmin)$  entonces
15:      píxel  $IMG_{RGB}(i, j) :=$  píxel de color verde
16:       $c := 1$ 
17:    fin si
18:    si  $(h > Bp.hmin) \wedge (h < Bp.hmax) \wedge (s > Bp.smin) \wedge (v > Bp.vmin)$  entonces
19:      píxel  $IMG_{RGB}(i, j) :=$  píxel de color azul
20:       $c := 1$ 
21:    fin si
22:    si  $c = 0$  entonces
23:      píxel  $IMG_{RGB}(i, j) :=$  píxel negro
24:    fin si
25:  fin para
26: fin para
```

Figura 3.17: Algoritmo; detectar píxeles de color.

En la figura 3.18 se representan gráficamente los píxeles de color detectados en una imagen tomada, utilizando el criterio de umbrales de color.

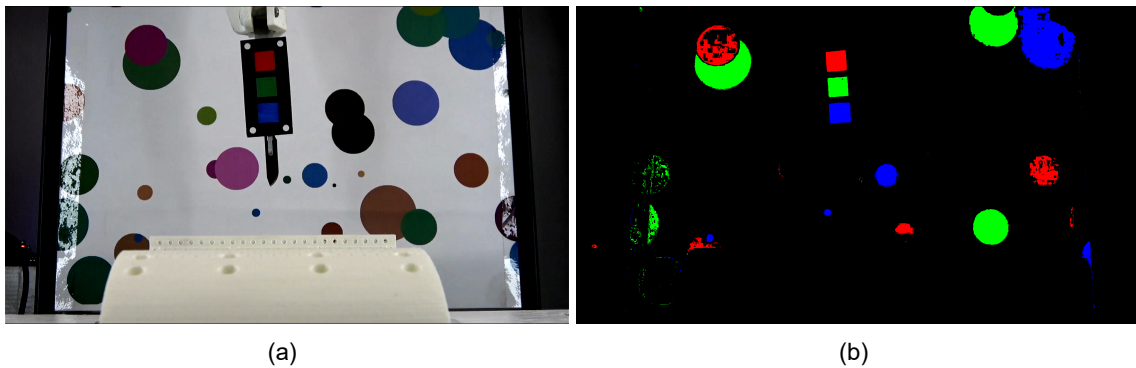


Figura 3.18: (a) Imagen tomada. (b) Representación gráfica de los píxeles de color detectados utilizando el criterio de umbrales de color.

3.4. Algoritmo de agrupación de píxeles por cuatro barridos

En esta sección se presenta un algoritmo que tiene como finalidad agrupar píxeles de valor lógico verdadero, adyacentes entre ellos, y contenidos en una imagen binaria. La agrupación se da por medio de la asignación y reasignación de números de grupo a cada píxel de valor lógico verdadero, de modo que cada píxel pueda ser identificado por medio de su número de grupo.

Este algoritmo se basa en la realización de cuatro barridos, entendiéndose como barrido al análisis secuencial de píxeles de una imagen, siguiendo un orden previamente establecido. En ese sentido, se eligió darle a este algoritmo el nombre de "Algoritmo de agrupación de píxeles por cuatro barridos".

En el primer barrido, el orden de análisis parte de la primera fila de píxeles completa, comenzando por el píxel ubicado en la columna inicial para después analizar uno por uno todos los píxeles de las columnas siguientes en orden ascendente. Tras terminar el análisis de toda la primera fila, se realiza la misma secuencia en la segunda y posteriormente en la tercera, hasta analizar la última fila. Durante este barrido, cada vez que se encuentre un píxel p , con valor lógico verdadero, se verificará si alguno de los píxeles adyacentes a p también tiene valor lógico verdadero y número de grupo. De ser así, se asignará a p como número de grupo, el menor número de grupo encontrado en los píxeles de valor lógico verdadero adyacentes a él. En caso contrario, se asignará a p un número de grupo mayor en una unidad al mayor número de grupo conocido. De no existir algún número de grupo conocido, se asignará a p , 1 como número grupo.

En los siguientes tres barridos, se realiza el mismo procedimiento que en el primero, pero con diferente orden de análisis.

En el segundo barrido, el orden de análisis parte de la primera columna, comenzando por el píxel ubicado en la fila inicial, para después analizar uno por uno todos los píxeles de las siguientes filas en orden ascendente. Tras terminar el análisis de toda la primera columna, se realiza la misma secuencia en la segunda y posteriormente en la tercera, hasta analizar la última columna.

En el tercer barrido, el orden de análisis parte de la primera fila completa, comenzando por el píxel ubicado en la última columna para después analizar uno por uno todos los píxeles ubicados en las columnas anteriores en orden descendente. Tras terminar el análisis

de toda la primera fila, se realiza la misma secuencia en la segunda y posteriormente en la tercera, hasta analizar la última fila.

En el cuarto barrido, el orden de análisis parte de la primera columna, comenzando por el píxel ubicado en la última fila para después analizar uno por uno todos los píxeles ubicados en las filas anteriores en orden descendente. Tras terminar el análisis de toda la primera columna, se realiza la misma secuencia en la segunda y posteriormente en la tercera, hasta analizar la última columna.

A continuación, se muestra un ejemplo de la aplicación del algoritmo de agrupación de píxeles por cuatro barridos. Este ejemplo se representa en la secuencia de figuras; 3.19 (a), 3.19 (b), 3.20 (a), y 3.20 (b). En el ejemplo se asignó un número de grupo a cada píxel de valor lógico verdadero siendo, G1 el grupo número 1, G2 el grupo número 2 y G3 el grupo número 3. Tal como se puede ver, conforme se van realizando los barridos, se van agregando píxeles a G1 y disminuyendo píxeles a G2 y G3 hasta la desaparición de G3.

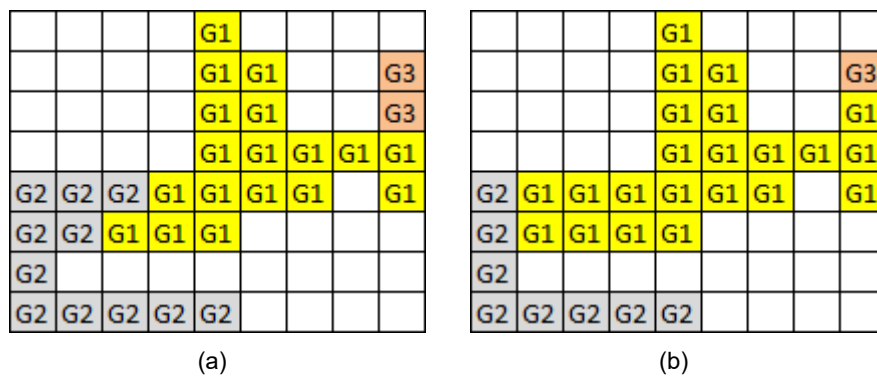


Figura 3.19: (a) Aplicación del primer barrido. (b) Aplicación del segundo barrido.

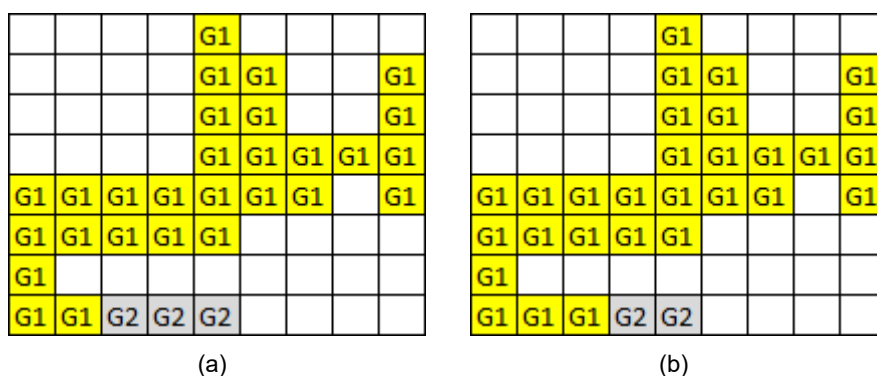


Figura 3.20: (a) Aplicación del tercer barrido. (b) Aplicación del cuarto barrido.

En la figura 3.20 (b), se puede ver que todos los píxeles de valor lógico verdadero son adyacentes entre ellos y, en consecuencia, se esperarí que conformen un mismo grupo.

Sin embargo, en la misma figura se puede corroborar que dichos píxeles conforman dos grupos diferentes (G1 y G2). Este hecho evidencia que, tras la aplicación del algoritmo de agrupación de píxeles por cuatro barridos, queda la posibilidad de que píxeles adyacentes entre ellos pertenezcan a grupos diferentes. Este resultado deficiente podría ser corregido aumentando la robustez del algoritmo. Sin embargo, hacerlo implicaría aumentar su tiempo de ejecución durante su funcionamiento un programa de computadora. Cabe resaltar que en la práctica se encontró que, a pesar de la posibilidad de ocurrencia de estos casos deficientes, el general, algoritmo logra agrupar visiblemente de forma satisfactoria los píxeles adyacentes.

El algoritmo de agrupación de píxeles por cuatro barridos, es descrito en el algoritmo de la figura 3.21 y tiene las siguientes entradas y salidas:

V : Imagen binaria.

$GROUP$: Matriz de números de grupo de los píxeles de valor lógico verdadero contenidos en V . Esta matriz tiene las mismas dimensiones que V .

Entrada: V

Salida: $GROUP$

```
1:  $fil :=$  cantidad de filas de  $V$ 
2:  $col :=$  cantidad de columnas de  $V$ 
3:  $GROUP :=$  matriz de ceros de  $fil$  filas y  $col$  columnas
4:  $nuevo := 0$ 
5: para  $i \leftarrow 1, fil$  hacer
6:   para  $j \leftarrow 1, col$  hacer
7:     si píxel  $V(i, j)$  es un píxel de valor lógico verdadero entonces
8:        $GROUP(i, j) :=$  menor número de grupo de su vecindad
9:     fin si
10:    si  $GROUP(i, j) = 0$  entonces
11:       $nuevo := nuevo + 1$ 
12:       $GRUPO(i, j) := nuevo$ 
13:    fin si
14:  fin para
15: fin para
16: para  $j \leftarrow 1, col$  hacer
17:   para  $i \leftarrow 1, fil$  hacer
18:     si píxel  $V(i, j)$  es un píxel de valor lógico verdadero entonces
19:        $GROUP(i, j) :=$  menor número de grupo de su vecindad
20:     fin si
21:   fin para
22: fin para
23: para  $i \leftarrow 1, fil$  hacer
24:   para  $j \leftarrow col, 1$  hacer
25:     si píxel  $V(i, j)$  es un píxel de valor lógico verdadero entonces
26:        $GROUP(i, j) :=$  menor número de grupo de su vecindad
27:     fin si
28:   fin para
29: fin para
30: para  $j \leftarrow 1, col$  hacer
31:   para  $i \leftarrow fil, 1$  hacer
32:     si píxel  $V(i, j)$  es un píxel de valor lógico verdadero entonces
33:        $GROUP(i, j) :=$  menor número de grupo de su vecindad
34:     fin si
35:   fin para
36: fin para
```

Figura 3.21: Algoritmo; agrupar píxeles por cuatro barridos.

3.5. Definición de las pusedo-marcas de color

Las marcas de color tienen como objetivo funcionar como referencias visuales para extraer la imagen reducida de una imagen tomada. En ese sentido, resulta ideal poder detectar cada una de las marcas de color. Sin embargo, teniendo en cuenta que el proceso de detección puede no ser perfecto, se asume que las regiones detectadas como resultado de

dicho proceso no necesariamente coincidan perfectamente con las marcas de color, pudiendo aquellas regiones incluir píxeles que no son parte de las marcas de color o incluso excluir píxeles que sí pertenecen a las marcas de color. Aquellas regiones obtenidas como resultado del proceso de intentar detectar las marcas de color son denominadas pseudo-marcas de color.

3.6. Obtención de las pseudo-marcas de color candidatas

En el presente trabajo de tesis, se propone un procedimiento compuesto por dos etapas para la obtención de las pseudo-marcas de color. Este procedimiento se basa en el análisis de una imagen IMG_{RGB} obtenida tras aplicar el criterio de umbrales de color a una imagen tomada.

La primera etapa consiste en analizar cada uno de los canales R, G, B de la imagen IMG_{RGB} . En cada canal se descartarán las regiones de color de áreas considerablemente diferentes al área estimada para la marca de color correspondiente al canal en análisis. Así mismo, se descartarán también las regiones de color cuyos centros geométricos estén ubicados en posiciones muy distantes al centro de la última pseudo-marca de color conocida, correspondiente al tipo de color del canal en análisis. Este procedimiento, aplicado a cada uno de los canales R, G, B, es descrito en el algoritmo de la figura 3.22, y tiene las siguientes entradas y salidas:

V : Imagen binaria.

$GROUP$: Matriz de números de grupo de los píxeles de valor lógico verdadero contenidos en V . Esta matriz es obtenida mediante el algoritmo de agrupación de píxeles por cuatro barridos.

A_{min} : Umbral inferior de área.

A_{max} : Umbral superior de área.

LC : Posición del centro de la última pseudo-marca de color conocida correspondiente al tipo de color del canal en análisis.

dms : Distancia entre los centros de dos marcas de color consecutivas. Este valor es previamente conocido e ingresado directamente por el usuario.

VC : Imagen contenedora de la pseudo-marca de color candidata.

Entrada: $V, GROUP, Amin, Amax, LC, dms$

Salida: VC

```
1: AREAS : vector
2: fil := cantidad de filas de V
3: col := cantidad de columnas de V
4: AREAS := asignar cada posición  $gr$  el área de cada grupo  $gr$  contenido en  $GROUP$ 
5: para  $i \leftarrow i, fil$  hacer
6:   para  $j \leftarrow i, col$  hacer
7:      $gr := GROUP(i, j)$ 
8:      $fi :=$  obtener media de las posiciones en filas de los píxeles del grupo  $gr$ 
9:      $co :=$  obtener media de las posiciones en columnas de los píxeles del grupo  $gr$ 
10:    si  $(AREAS(gr) < Amin) \vee (AREAS(gr) > Amax)$  entonces
11:      píxel  $V(i, j) :=$  píxel negro
12:    fin si
13:    si  $(|fi - LC(1)| > dms) \vee (|co - LC(2)| > dms)$  entonces
14:      píxel  $V(i, j) :=$  píxel negro
15:    fin si
16:  fin para
17: fin para
18:  $VC := V$ 
```

Figura 3.22: Algoritmo; obtener la pseudo-marca candidata.

La primera etapa tiene como resultado 3 imágenes binarias. Cada una en un canal del espacio de color RGB. Estas imágenes pueden contener una, algunas o ninguna región de color. Se denomina pseudo-región de color candidata a la unión de todas las regiones de color contenidas en cada una de estas 3 imágenes, teniendo de esta forma la pseudo-marca de color candidata roja, verde, y azul. Así mismo, las 3 imágenes se denominan imágenes contenedoras de las pseudo-marcas de color candidatas.

Experimentalmente se encontró que resulta provechoso aplicar operaciones de erosión y dilatación a una imagen IMG_{RGB} antes de aplicar el algoritmo de la figura 3.22, a los canales R, G, B.

En la figura 3.23, se muestra una representación gráfica de la aplicación del algoritmo de la figura 3.22 a los canales R, G, B de la imagen resultante de aplicar las operaciones de erosión y dilatación a una imagen IMG_{RGB} .

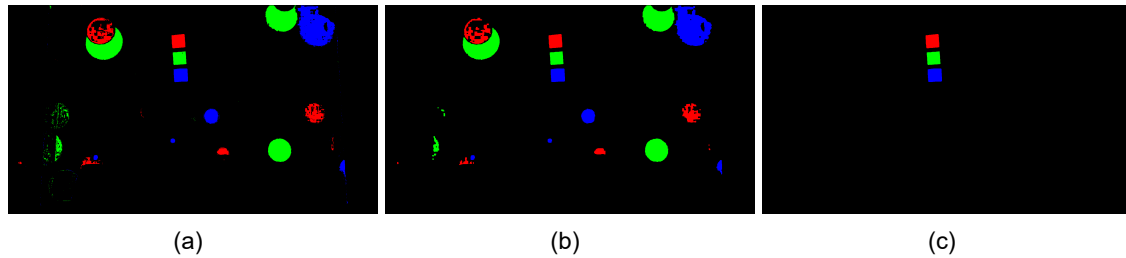


Figura 3.23: (a) Imagen IMG_{RGB} . (b) Aplicación de erosión y dilatación a IMG_{RGB} . (c) Imagen en color, contenedora de las 3 pseudo-marcas de color candidatas.

3.7. Análisis de confiabilidad de pseudo-regiones de color candidatas

En esta sección se propone un procedimiento para determinar si una pseudo-marca de color candidata es confiable o no. De acuerdo con ello, únicamente las pseudo-marcas de color candidatas confiables pasarán a denominarse pseudo-marcas de color.

Este procedimiento se conforma por la evaluación de confiabilidad de cada pseudo-marca de color candidata de acuerdo con los siguientes tres criterios:

- Criterio de medias y medianas.
- Criterio de varianzas.
- Criterio de áreas.

3.7.1. Criterio de medias y medianas

Al ser las marcas de color regiones cuadradas, se considera que idealmente los valores numéricos de la media y la mediana de posición de los píxeles que las conforman son iguales tanto en el eje horizontal como en el vertical. Siendo así, a pesar de que una pseudo-marca de color no necesariamente coincida perfectamente con una marca de color, se espera que los valores de la media y la mediana de posición de los píxeles que la conforman sean cercanos e, idealmente, iguales.

En ese sentido, el criterio de medias y medianas establece que; una pseudo-marca de color candidata es confiable si y solo si los valores absolutos de las diferencias entre sus medias y medianas de posición, tanto en el eje horizontal como vertical son pequeños. En el presente trabajo de tesis no se establece algún criterio matemático para el cálculo del valor máximo de aquella diferencia pequeña. Sin embargo, experimentalmente se encontraron buenos resultados considerando aquel valor igual a la longitud estimada de la marca de

color, L_m . En resumen, este criterio sostiene que una pseudo-marca de color candidata es confiable si y solo si cumple con la ecuación (3.3).

$$abs(X_{mda} - X_{mdn}) < L_m \wedge abs(Y_{mda} - Y_{mdn}) < L_m \quad (3.3)$$

Donde

X_{mda} : Valor numérico de la media de posición en el eje vertical de los píxeles que conforman la pseudo-marca de color candidata.

X_{mdn} : Valor numérico de la mediana de posición en el eje vertical de los píxeles que conforman la pseudo-marca de color candidata.

Y_{mda} : Valor numérico de la media de posición en el eje horizontal de los píxeles que conforman la pseudo-marca de color candidata.

Y_{mdn} : Valor numérico de la mediana de posición en el eje horizontal de los píxeles que conforman la pseudo-marca de color candidata.

3.7.2. Criterio de varianzas

En la sección 3.3.4, se indica que V_m es el valor numérico estimado de la varianza de posición en los ejes horizontal y vertical para una marca de color. En ese sentido, a pesar de que una pseudo-marca de color puede no coincidir perfectamente con una marca de color, se espera que el valor numérico de la varianza de posición V_c en los ejes horizontal y vertical, de los píxeles que conforman la pseudo-marca de color, sea cercana e idealmente igual a V_m . En la práctica puede suceder alguno de los siguientes tres escenarios:

Escenario 1, cuando $V_c < V_m$: Los píxeles que conforman la pseudo-marca de color candidata se encuentran menos dispersos que la estimación realizada para su correspondiente marca de color. Esto implica que no se ha detectado por completo la marca de color.

Escenario 2, cuando $V_c = V_m$: Los píxeles que conforman la pseudo-marca de color candidata son exactamente igual de dispersos que la estimación realizada para su correspondiente marca de color. Esto no implica necesariamente que se ha detectado la totalidad de la marca de color, pudiendo darse el caso de que la pseudo-marca de color candidata esté conformada también por píxeles que no pertenezcan a la marca de color y no contenga píxeles que sí pertenezcan a ella.

Escenario 3, cuando $V_c > V_m$: Los píxeles que conforman la pseudo-marca de color candidata se encuentran más dispersos que la estimación realizada para su correspondien-

te marca de color. Esto implica que existen píxeles no pertenecientes a la marca de color que han sido erradamente incluidos dentro de la pseudo-marca de color candidata.

Experimentalmente se encontró que el error más recurrente sucede cuando se da el escenario 3. Es por tal motivo que el criterio de varianzas se enfoca en rechazar las pseudo-regiones de color candidatas que cumplan con aquel escenario y se incluye un factor de tolerancia f_{var} . En resumen, este criterio establece que una pseudo-marca de color candidata es confiable si y solo si el cumple con la ecuación (3.4).

$$X_{var} < f_{var} \cdot V_m \wedge Y_{var} < f_{var} \cdot V_m \quad (3.4)$$

Donde

X_{var} : Varianza de posición de los píxeles que conforman la pseudo-marca de color candidata con respecto al eje vertical.

Y_{var} : Varianza de posición de los píxeles que conforman la pseudo-marca de color candidata con respecto al eje horizontal.

V_m : Varianza estimada de posición en los ejes horizontal y vertical de la marca de color.

f_{var} : Factor de tolerancia de la varianza.

El presente trabajo de tesis no establece algún criterio matemático para calcular f_3 . Sin embargo, experimentalmente se encontraron buenos resultados para $f_{var} = 2$, tal como se emplea en el algoritmo de la figura 3.24.

3.7.3. Criterio de áreas

En la sección 3.3.4 se indica que A_m es el valor numérico estimado del área de una marca de color. En ese sentido, a pesar de que una pseudo-marca de color puede no coincidir perfectamente con una marca de color, se espera que el valor numérico de su área denominada A_c sea cercano a A_m e idealmente igual. En la práctica se pueden dar alguno de los siguientes 3 escenarios:

Escenario 1, cuando $A_c < A_m$: Las regiones de color que conforman la pseudo-marca de color candidata suman un área inferior al área estimada de su correspondiente marca de color. Esto implica que no se ha detectado por completo la marca de color.

Escenario 2, cuando $A_c = A_m$: Las regiones de color que conforman la pseudo-marca de color candidata suman un área exactamente igual a su correspondiente marca de color.

Esto no implica necesariamente que se ha detectado la totalidad de la marca de color, pudiendo suceder que la pseudo-marca de color candidata esté conformada por regiones de color que no pertenecen a la marca de color, pero que suman la misma área.

Escenario 3, cuando $A_c > A_m$: Las regiones de color que conforman la pseudo-marca de color candidata suman un área superior al área estimada de su correspondiente marca de color. Esto implica que existen píxeles no pertenecientes a la marca de color que han sido erradamente incluidos dentro de la pseudo-marca de color candidata.

Experimentalmente se encontró que el error más recurrente es cuando se da el escenario 3. Es por ello que el criterio de áreas se enfoca en rechazar las pseudo-regiones de color candidatas que cumplan con aquel escenario y se incluye un factor de tolerancia f_{ar} . En resumen, este criterio establece que una pseudo-marca de color candidata es confiable si y solo si el cumple con la ecuación (3.5).

$$A_c < f_{ar} \cdot A_m \quad (3.5)$$

Donde

A_c : Área de la pseudo-marca de color candidata.

A_m : Área estimada de la marca de color.

f_{ar} : Factor de tolerancia de área.

El presente trabajo de tesis no establece algún criterio matemático para calcular f_{ar} . Sin embargo, experimentalmente se encontraron buenos resultados para $f_{ar}=1.5$, tal como se emplea en el algoritmo de la figura 3.24.

3.7.4. Obtención de las pseudo-marcas de color

El procedimiento seguido para la evaluación de la confiabilidad de las pseudo-marcas de color candidatas de acuerdo con los tres criterios de confiabilidad, es descrito en el algoritmo de la figura 3.24 y tiene las siguientes entradas y salidas:

CV_s : Imagen contenedora de las pseudo-marcas de color candidatas.

Rp, Gp, Bp : Estructuras de parámetros comunes de los píxeles de color rojos, verdes y azules, respectivamente.

M_{REST} : Matriz de confiabilidad de las pseudo-marcas de color candidatas.

Entrada: CV_s, Rp, Gp, Bp

Salida: M_{REST}

- 1: $CM, CV, CA :=$ vectores de ceros
- 2: $M_{REST} :=$ matriz de 3 filas y 3 columnas
- 3: $fil :=$ cantidad de filas de CV_s
- 4: $col :=$ cantidad de columnas de CV_s
- 5: obtener $xr_{mda}, xr_{mdn}, xr_{var}, yr_{mda}, yr_{mdn}, yr_{var}$ de CV_s
- 6: obtener $xg_{mda}, xg_{mdn}, xg_{var}, yg_{mda}, yg_{mdn}, yg_{var}$ de CV_s
- 7: obtener $xb_{mda}, xb_{mdn}, xb_{var}, yb_{mda}, yb_{mdn}, yb_{var}$ de CV_s
- 8: obtener $nr, ng, nb, xr_{dif}, yr_{dif}, xg_{dif}, yg_{dif}, xb_{dif}, yb_{dif}$ de CV_s
- 9: **si** $xr_{dif} < Rp.Lm \wedge yr_{dif} < Rp.Lm \wedge xr_{mda} > 0$ **entonces**
- 10: $CM(1) := 1$
- 11: **fin si**
- 12: **si** $xg_{dif} < Gp.Lm \wedge yg_{dif} < Gp.Lm \wedge xg_{mda} > 0$ **entonces**
- 13: $CM(2) := 1$
- 14: **fin si**
- 15: **si** $xb_{dif} < Bp.Lm \wedge yb_{dif} < Bp.Lm \wedge xb_{mda} > 0$ **entonces**
- 16: $CM(3) := 1$
- 17: **fin si**
- 18: **si** $xr_{var} \leq 2Rp.Vm \wedge yr_{var} \leq 2Rp.Vm \wedge xr_{var} > 0 \wedge yr_{var} > 0$ **entonces**
- 19: $CV(1) := 1$
- 20: **fin si**
- 21: **si** $xg_{var} \leq 2Gp.Vm \wedge yg_{var} \leq 2Gp.Vm \wedge xg_{var} > 0 \wedge yg_{var} > 0$ **entonces**
- 22: $CV(2) := 1$
- 23: **fin si**
- 24: **si** $xb_{var} \leq 2Bp.Vm \wedge yb_{var} \leq 2Bp.Vm \wedge xb_{var} > 0 \wedge yb_{var} > 0$ **entonces**
- 25: $CV(3) := 1$
- 26: **fin si**
- 27: **si** $nr < 1,5Rp.Am$ **entonces**
- 28: $CA(1) := 1$
- 29: **fin si**
- 30: **si** $ng < 1,5Gp.Am$ **entonces**
- 31: $CA(2) := 1$
- 32: **fin si**
- 33: **si** $nb < 1,5Bp.Am$ **entonces**
- 34: $CA(3) := 1$
- 35: **fin si**
- 36: $M_{REST}(1, 1) :=$ obtener la mediana de $\{CM(1), CV(1), CA(1)\}$
- 37: $M_{REST}(1, 2) :=$ obtener la mediana de $\{CM(2), CV(2), CA(2)\}$
- 38: $M_{REST}(1, 3) :=$ obtener la mediana de $\{CM(3), CV(3), CA(3)\}$
- 39: $M_{REST}(2, 1) := xr_{mda}$
- 40: $M_{REST}(2, 2) := xg_{mda}$
- 41: $M_{REST}(2, 3) := xb_{mda}$
- 42: $M_{REST}(3, 1) := yr_{mda}$
- 43: $M_{REST}(3, 2) := yg_{mda}$
- 44: $M_{REST}(3, 3) := yb_{mda}$

Figura 3.24: Algoritmo; evaluar la confiabilidad.

Donde

$xr_{mda}, xg_{mda}, yb_{mda}$: Media de posición en el eje vertical de los píxeles que conforman las pseudo-marcas de color roja, verde y azul, respectivamente.

$xr_{mdn}, xg_{mdn}, yb_{mdn}$: Mediana de posición en el eje vertical de los píxeles que conforman las pseudo-marcas de color roja, verde y azul, respectivamente.

$xr_{var}, xg_{var}, yb_{var}$: Varianza de posición en el eje vertical de los píxeles que conforman las pseudo-marcas de color roja, verde y azul, respectivamente.

$yr_{mda}, yg_{mda}, yb_{mda}$: Media de posición en el eje horizontal de los píxeles que conforman las pseudo-marcas de color roja, verde y azul, respectivamente.

$yr_{mdn}, yg_{mdn}, yb_{mdn}$: Mediana de posición en el eje horizontal de los píxeles que conforman las pseudo-marcas de color roja, verde y azul, respectivamente.

$yr_{var}, yg_{var}, yb_{var}$: Varianza de posición en el eje horizontal de los píxeles que conforman las pseudo-marcas de color roja, verde y azul, respectivamente.

nr, ng, nb : Cantidad de píxeles de color rojo, verde y azul, respectivamente, encontrados en CVs .

xr_{dif} : Valor absoluto de la diferencia entre xr_{mda} y xr_{mdn} .

yr_{dif} : Valor absoluto de la diferencia entre yr_{mda} y yr_{mdn} .

xg_{dif} : Valor absoluto de la diferencia entre xg_{mda} y xg_{mdn} .

yg_{dif} : Valor absoluto de la diferencia entre yg_{mda} y yg_{mdn} .

xb_{dif} : Valor absoluto de la diferencia entre xb_{mda} y xb_{mdn} .

yb_{dif} : Valor absoluto de la diferencia entre yb_{mda} y yb_{mdn} .

Se establece que toda pseudo-marca de color candidata pasará a denominarse pseudo-marca de color si y solo si la pseudo-marca de color candidata es considerada confiable de acuerdo con por lo menos 2 de los 3 criterios de confiabilidad.

En la figura 3.25, se muestra una representación gráfica de las pseudo-marcas de color encontradas en una imagen tomada.

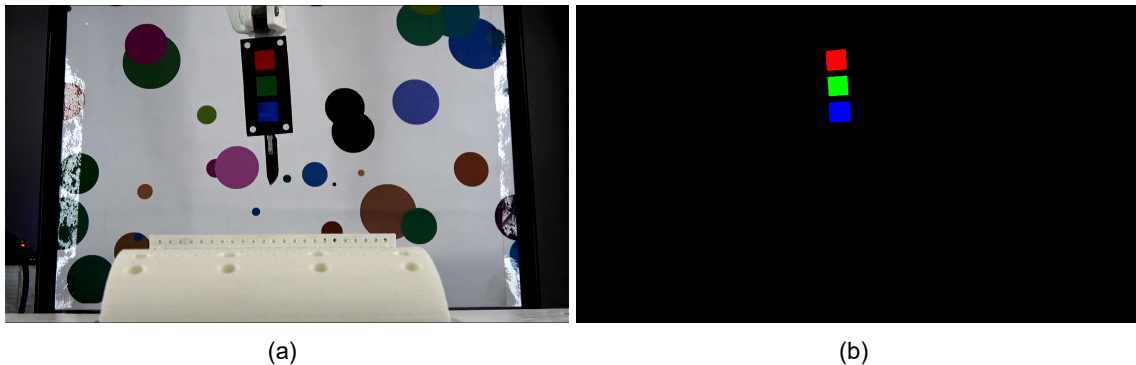


Figura 3.25: (a) Imagen tomada. (b) Pseudo-marcas de color.

3.8. Obtención de imagen reducida

3.8.1. La región del bisturí

Se define como región del bisturí al conjunto de los píxeles en color que conforman la representación visual del bisturí dentro de una imagen reducida (véase la sección 3.8.3). Esta definición se enmarca dentro de la situación ideal en la que es posible conocer todos los píxeles que conforman aquella representación visual, sin omitir o agregar algún otro píxel. En la figura 3.26 se muestra una imagen tomada (a) y la región del bisturí contenida en ella (b).

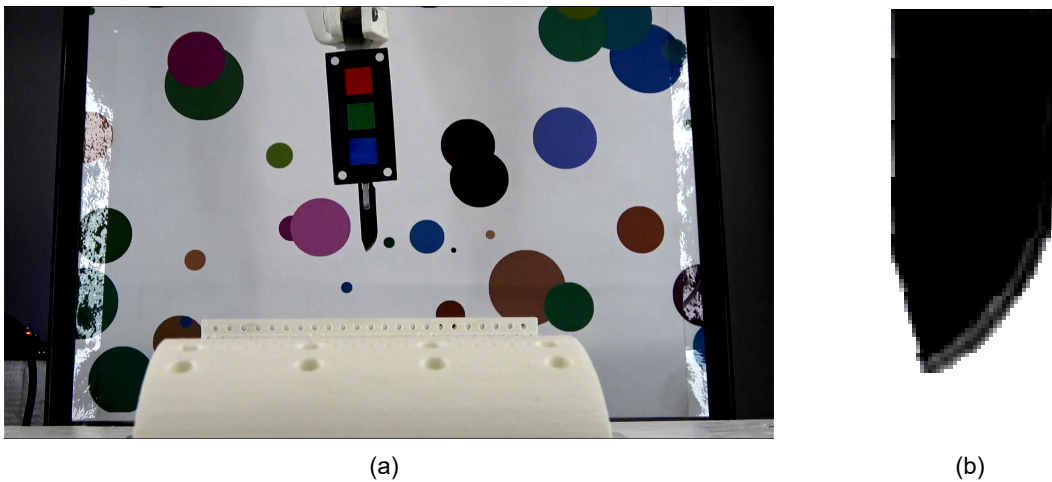


Figura 3.26: (a) Imagen tomada. (b) Región del bisturí.

3.8.2. La pseudo-región del bisturí

Se define como pseudo-región del bisturí al conjunto de píxeles obtenidos como resultado del proceso de intentar detectar los píxeles que conforman la región del bisturí. Esta definición implica que durante la obtención de la pseudo-región del bisturí han podido omitirse píxeles pertenecientes a la región del bisturí o se han podido agregar erradamente otros píxeles que no pertenezcan a dicha región.

A diferencia de la región del bisturí que está compuesta por píxeles en color, se decidió que la pseudo-región del bisturí esté compuesta por píxeles binarios. Esto debido a que el principio para la estimación de la profundidad de corte, indicado en la ecuación (3.1), se fundamenta en la variación geométrica de la región del bisturí percibida en una imagen tomada y no de los colores que conformen dicha región.

En la figura 3.27 se muestra una imagen tomada y la pseudo-región del bisturí extraída de ella.

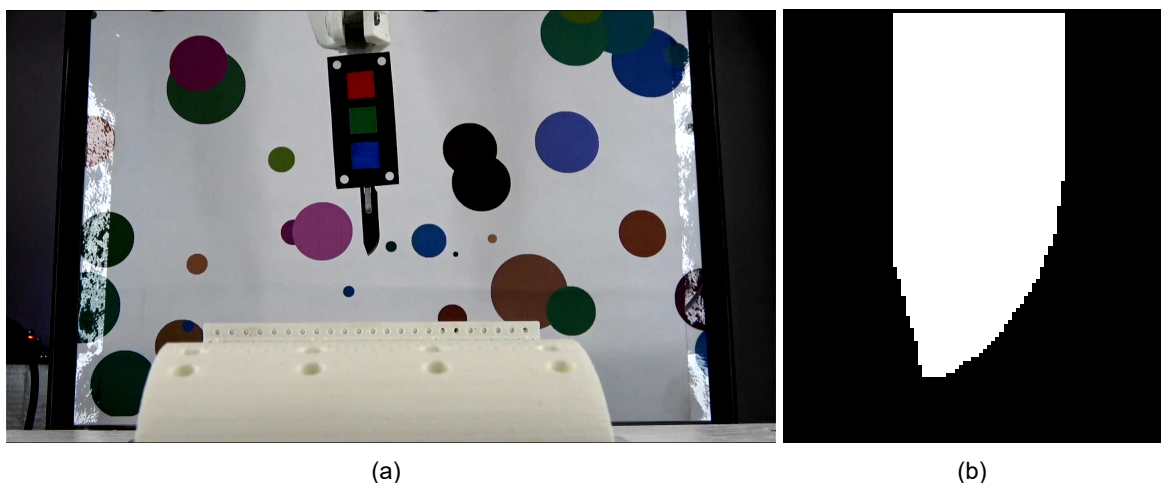


Figura 3.27: (a) Imagen tomada. (b) Representación binaria de la pseudo-región del bisturí en color blanco con fondo negro.

3.8.3. La imagen reducida

Se define como imagen reducida a una sub-imagen rectangular extraída de una imagen tomada, en la que se busca abarcar la totalidad de la región que conforma únicamente la parte de hoja del bisturí que puede ingresar a la muestra durante una incisión vertical, evitando a su vez abarcar regiones alejadas de la hoja y procurando que la hoja esté dispuesta verticalmente en la sub-imagen. La obtención de la imagen reducida tiene como objetivo reducir el área de análisis para la obtención de la región de bisturí a únicamente una parte de la imagen tomada.

Se obtiene la imagen recudida IMG_{SML2} como una sub-imagen de una imagen IMG_{SML1} , la que a su vez es obtenida rotando an grados una imagen tomada IMG con respecto a un punto P_3 . Asimismo, la obtención de IMG_{SML2} también depende de las constantes bas y has .

El algoritmo de la figura 3.12, describe un procedimiento para calcular bas y has . Por su parte, el cálculo de P_3 y an depende de la obtención de los puntos P_0 , P_1 y P_2 . Así mismo dms es la distancia conocida entre los centros de dos marcas de color consecutivas, y es ingresada directamente por el usuario.

Donde

P_r : Posición del centro de la pseudo-marca de color roja.

P_g : Posición del centro de la pseudo-marca de color verde.

P_b : Posición del centro de la pseudo-marca de color azul.

P_0 : Posición estimada del centro de la pseudo-marca de color verde.

P_1 : Posición del centro de la pseudo-marca de color que se ubique en la posición más alta, en caso se logre obtener más de una pseudo-marca de color.

P_2 : Posición del centro de la pseudo-marca de color que se ubique en la posición más baja, en caso se logre obtener más de una pseudo-marca de color.

P_3 : Posición del punto medio estimado entre la punta del bisturí y el límite hasta donde la hoja del bisturí puede ingresar verticalmente a la muestra.

dms : Distancia entre los centros de dos marcas de color consecutivas.

La obtención de la imagen reducida se puede dar dentro de alguno de los siguientes 4 posibles escenarios. Cabe anotar que, en estos escenarios, se indica entre paréntesis x para referirse al componente en el eje vertical, e y para referirse al componente horizontal de cada punto.

Escenario 1: No se tiene ni una pseudo-marca de color, entonces no es posible obtener la imagen reducida.

Escenario 2: Únicamente se tiene una pseudo-marca de color, entonces se pueden dar los siguientes 3 sub-escenarios:

- Sub-escenario 2.1: La pseudo-marca de color es roja:

$$P_0(x) = P_r(x) + dms, P_0(y) = P_r(y) \quad (3.6)$$

- Sub-escenario 2.2: La pseudo-marca de color es verde:

$$P_0(x) = P_g(x), P_0(y) = P_g(y) \quad (3.7)$$

- Sub-escenario 2.3: La pseudo-marca de color es azul:

$$P_0(x) = P_b(x) - dms, P_0(y) = P_b(y) \quad (3.8)$$

Al no tener más información, se asume que el bisturí se orienta de forma totalmente vertical con respecto a IMG , por lo tanto se asume que $an = 0$:

Escenario 3: Únicamente se tienen 2 pseudo-marcas de color, entonces pueden dar los siguientes 3 sub-escenarios:

- Sub-escenario 3.1: Las pseudo-marcas de color son la roja y la verde:

$$\begin{aligned}
 P_0(x) &= P_g(x), P_0(y) = P_g(y) \\
 P_1(x) &= P_r(x), P_1(y) = P_r(y) \\
 P_2(x) &= P_g(x), P_2(y) = P_g(y)
 \end{aligned} \tag{3.9}$$

- Sub-escenario 3.2: Las pseudo-marcas de color son la verde y la azul.

$$\begin{aligned}
 P_0(x) &= P_g(x), P_0(y) = P_g(y) \\
 P_1(x) &= P_g(x), P_1(y) = P_g(y) \\
 P_2(x) &= P_b(x), P_2(y) = P_b(y)
 \end{aligned} \tag{3.10}$$

- Sub-escenario 3.3: Las pseudo-marcas de color son la roja y la azul.

$$\begin{aligned}
 P_0(x) &= \frac{P_r(x) + P_g(x)}{2} \\
 P_0(y) &= \frac{P_r(y) + P_g(y)}{2} \\
 P_1(x) &= P_r(x), P_1(y) = P_r(y) \\
 P_2(x) &= P_b(x), P_2(y) = P_b(y)
 \end{aligned} \tag{3.11}$$

Escenario 4: Se tienen las tres pseudo-marcas de color, entonces:

$$\begin{aligned}
 P_0(x) &= P_g(x), P_0(y) = P_g(y) \\
 P_1(x) &= P_r(x), P_1(y) = P_r(y) \\
 P_2(x) &= P_b(x), P_2(y) = P_b(y)
 \end{aligned} \tag{3.12}$$

En los escenarios 3 y 4 se calcula an utilizando la ecuación (3.13).

$$an = asin\left(\frac{P_2(y) - P_1(y)}{\sqrt{(P_2(x) - P_1(x))^2 + (P_2(y) - P_1(y))^2}}\right) \tag{3.13}$$

En la sección 3.3.5, se obtiene la constante has , empleando el algoritmo de la figura 3.12. En dicha sección se indica que has es la distancia entre la punta del bisturí y el límite hasta donde la hoja del bisturí puede ingresar verticalmente a la muestra, multiplicada por un factor de escalamiento. En ese sentido, de obtener la imagen reducida como una sub-imagen de IMG_{SML1} concéntrica a IMG_{SML1} de longitudes de altura has y base bas , esta sub-imagen podría contener parte de la hoja del bisturí que no pueda ingresar a la muestra.

A fin de prevenir ello, se eligió obtener la imagen reducida como el resultado de eliminar la cuarta parte superior de la sub-imagen de IMG_{SML1} concéntrica a IMG_{SML1} y de longitudes de altura has y base bas .

El procedimiento seguido para la obtención de la imagen reducida IMG_{SML2} es descrito en el algoritmo de la figura 3.28 y tiene las siguientes entradas y salidas:

IMG : Imagen tomada.

M_{REST} : Matriz de confiabilidad de las pseudo-marcas de color candidatas.

bas : Longitud de la base de la imagen reducida.

has : Longitud referencial para la altura de la imagen reducida.

dis : Distancia entre el centro de la marca de color verde y el punto medio entre la punta del bisturí, y el límite hasta donde la hoja del bisturí puede ingresar verticalmente a la muestra.

dms : Distancia entre los centros de dos marcas de color consecutivas.

IMG_{SML2} : Imagen reducida.

Entrada: $IMG, M_{REST}, bas, has, dis, dms$

Salida: IMG_{SML2}

- 1: $fil :=$ cantidad de filas de IMG
- 2: $col :=$ cantidad de columnas de IMG
- 3: $P_0, P_1, P_2, P_3 :=$ vectores
- 4: **si** $M_{REST}(1, 1) = 1 \wedge M_{REST}(1, 2) = 0 \wedge M_{REST}(1, 3) = 0$ **entonces**
- 5: $P_0(1) := M_{REST}(2, 1) + dms$
- 6: $P_0(2) := M_{REST}(3, 1)$
- 7: **fin si**
- 8: **si** $M_{REST}(1, 1) = 0 \wedge M_{REST}(1, 2) = 1 \wedge M_{REST}(1, 3) = 0$ **entonces**
- 9: $P_0(1) := M_{REST}(2, 2)$
- 10: $P_0(2) := M_{REST}(3, 2)$
- 11: **fin si**
- 12: **si** $M_{REST}(1, 1) = 0 \wedge M_{REST}(1, 2) = 0 \wedge M_{REST}(1, 3) = 1$ **entonces**
- 13: $P_0(1) := M_{REST}(2, 3) - dms$
- 14: $P_0(2) := M_{REST}(3, 3)$
- 15: **fin si**
- 16: **si** $M_{REST}(1, 1) = 1 \wedge M_{REST}(1, 2) = 1 \wedge M_{REST}(1, 3) = 0$ **entonces**
- 17: $P_0(1) := M_{REST}(2, 2)$
- 18: $P_0(2) := M_{REST}(3, 2)$
- 19: $P_1(1) := M_{REST}(2, 1)$
- 20: $P_1(2) := M_{REST}(3, 1)$
- 21: $P_2(1) := M_{REST}(2, 2)$
- 22: $P_2(2) := M_{REST}(3, 2)$
- 23: **fin si**
- 24: **si** $M_{REST}(1, 1) = 0 \wedge M_{REST}(1, 2) = 1 \wedge M_{REST}(1, 3) = 1$ **entonces**
- 25: $P_0(1) := M_{REST}(2, 2)$
- 26: $P_0(2) := M_{REST}(3, 2)$
- 27: $P_1(1) := M_{REST}(2, 2)$
- 28: $P_1(2) := M_{REST}(3, 2)$
- 29: $P_2(1) := M_{REST}(2, 3)$
- 30: $P_2(2) := M_{REST}(3, 3)$
- 31: **fin si**
- 32: **si** $M_{REST}(1, 1) = 1 \wedge M_{REST}(1, 3) = 1$ **entonces**
- 33: $P_0(1) := (M_{REST}(2, 1) + M_{REST}(2, 3))/2$
- 34: $P_0(2) := (M_{REST}(3, 1) + M_{REST}(3, 3))/2$
- 35: $P_1(1) := M_{REST}(2, 1)$
- 36: $P_1(2) := M_{REST}(3, 1)$
- 37: $P_2(1) := M_{REST}(2, 3)$
- 38: $P_2(2) := M_{REST}(3, 3)$
- 39: **fin si**
- 40: $an = \text{asin}((P_2(2) - P_1(2))/\sqrt{(P_2(1) - P_1(1))^2 + (P_2(2) - P_1(2))^2})$
- 41: $P_3(1) := P_0(1) + dis \cdot \cos(an)$
- 42: $P_3(2) := P_0(2) + dis \cdot \sin(an)$
- 43: $IMG :=$ trasladar el centro de IMG a P_3
- 44: $IMG_{SML1} :=$ rotar IMG , an grados
- 45: $IMG_{SML2} :=$ extraer de IMG_{SML1} , una sub-imagen concéntrica a IMG_{SML1} , de bas columnas y has filas, y eliminar la cuarta parte superior.

Figura 3.28: Algoritmo; obtener la imagen reducida.

En la figura 3.29, se muestra un ejemplo de aplicación del algoritmo de la figura 3.28.

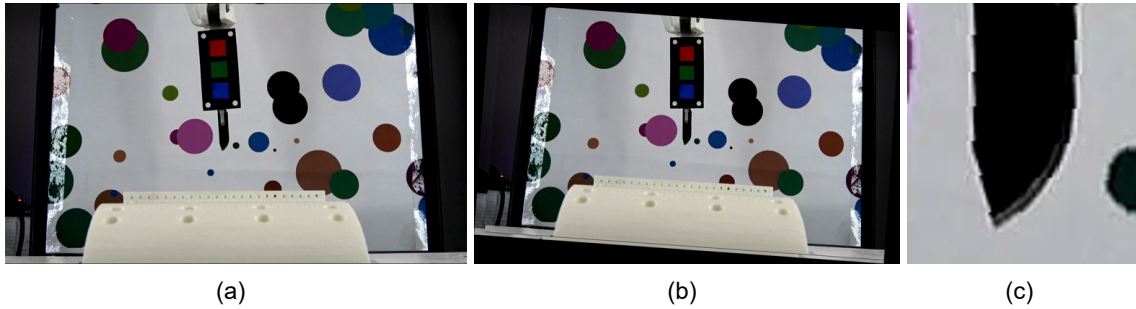


Figura 3.29: (a) Imagen tomada IMG . (b) IMG rotada an grados (c) Imagen Reducida.

3.9. Obtención de la pseudo-región del bisturí

En esta sección se propone un método para obtener la pseudo-región del bisturí contenida en una imagen reducida IMG_{SML2} . Para establecer este método, se optó por analizar la imagen reducida en los espacios de color RGB y HSV, a fin de establecer un criterio que permita diferenciar los píxeles pertenecientes a la pseudo-región del bisturí, del resto de la imagen reducida. En la figura 3.30, se muestra una representación en escala de grises de los canales que componen los espacios de color RGB y HSV de una imagen reducida.

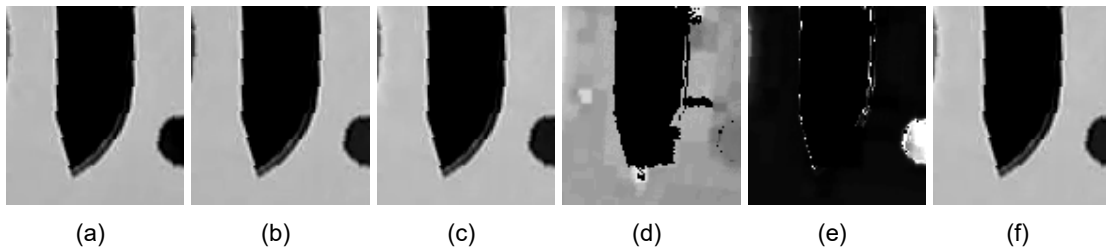


Figura 3.30: Canales extraídos de una imagen reducida; R (a), G (b), B (c), H (d), S (e) y V (f).

En la figura 3.30, se puede observar que en los canales R, G, B, H, S, y V, la región del bisturí aparentemente queda delimitada por los bordes que definen su perfil.

En la figura 3.31, se muestra una representación de los bordes detectados en los canales que componen los espacios de color RGB y HSV de la imagen reducida, bordes que fueron obtenidos utilizando el detector de bordes Canny [4].

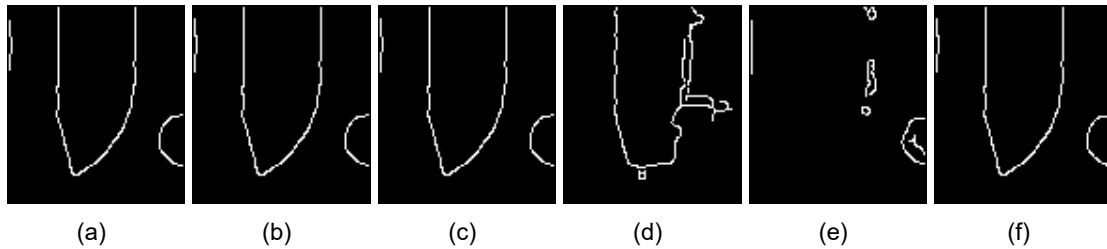


Figura 3.31: Bordes detectados en los canales extraídos de una imagen reducida; R (a), G (b), B(c), H(d), S(e), y V(f).

En la figura 3.31, se puede ver que todos los canales contienen parcialmente los bordes del bisturí. Esto sugiere que obtener los bordes que delimitan la región del bisturí en base a la información proveniente de un único canal puede ser insuficiente. En la misma figura se observa que, en los canales R, G, B, y V, existen bordes que delimitan bastante bien la región del bisturí, marcando clara separación entre aquella región y el resto de la imagen. Por otro lado, en el canal S existe carencia de bordes que delimiten la región del bisturí. Por su parte, en el canal H existen bordes que en lugar de separar, fusionan la región del bisturí con el resto de la imagen. Se considera que la posibilidad de delimitar exitosamente la región del bisturí aumenta en tanto se utilice la información de bordes proveniente de más canales; sin embargo, la posibilidad de inducir al error aumenta cuando se agregan bordes que en lugar de delimitar la región del bisturí, la fusionen con el resto de la imagen. Siendo así, se eligió fusionar los bordes obtenidos los canales R, G, B, S, y V, en una única imagen de fusión de bordes, tal como se muestra en la figura 3.32 (a).

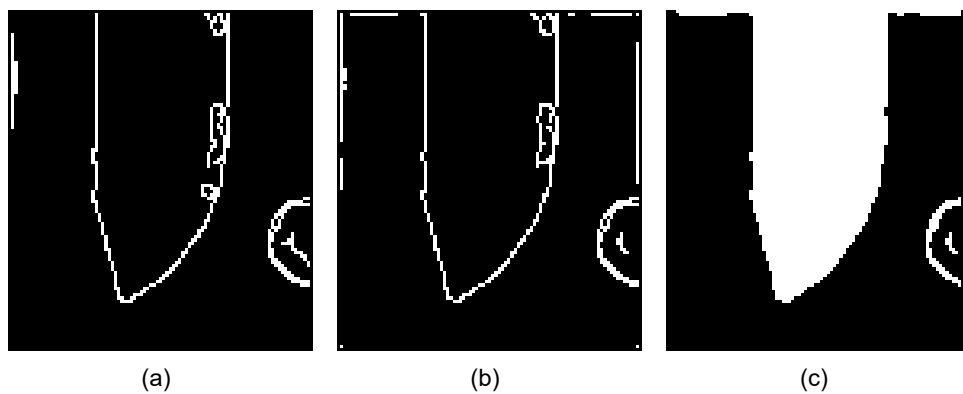


Figura 3.32: (a) Imagen de fusión de bordes. (b) Imagen de fusión de bordes con aplicación de un filtro media antes de detección de bordes. (c) Imagen con contornos cerrados rellenos, IMG_{SCL1} .

Cabe resaltar que en la práctica se encontró oportuno aplicar un filtro media a la imagen reducida antes de realizar la detección de bordes, a fin de atenuar el ruido que pudiera estar presente en la imagen tomada, obteniendo así la imagen RT mostrada en la figura 3.32 (b).

Tal como se puede ver en la figura 3.32 (b), ha sido posible obtener exitosamente los bordes que delimitan la región del bisturí. Este éxito, abre la posibilidad de obtener la pseudo-región del bisturí mediante el rellenado de los contornos cerrados presentes en RT . Sin embargo, durante la detección de estos no se encontraron bordes que completen la parte superior del contorno de la región del bisturí. Así mismo, es probable que otros objetos diferentes al bisturí se encuentren dentro de la imagen reducida y estos puedan definir otros contornos cerrados a los lados o debajo de la región del bisturí. Es por ello que se optó por agregar una línea en toda la primera fila de RT , que pudiera completar el contorno superior de la región del bisturí. Además, se borraron las primeras y las últimas 3 columnas, y, las últimas 3 filas de RT , a fin de evitar la conformación de otras regiones cerradas que lleguen a aquellos límites. Esta cantidad de filas y columnas a borrar fue elegida con el objetivo de evitar la conformación de regiones cerradas incluso si se aplicara la operación de erosión con radio de 2 unidades, pudiendo aumentar o disminuir dicha cantidad según sea el caso. Posteriormente, se utilizó el procedimiento descrito en [22], para rellenar los contornos cerrados restantes y así obtener la imagen IMG_{SCL1} , misma que se muestra en la figura 3.32 (c).

En la figura 3.32 (c) se puede ver que todavía quedan regiones indeseables que no pertenecen a la región del bisturí. Para descartarlas, se optó por realizar una operación de erosión que pudiera descartar regiones delgadas. Seguidamente, se agruparon los píxeles de valor lógico verdadero contenidos en IMG_{SCL1} por medio del algoritmo de agrupación de píxeles por cuatro barridos (véase la sección 3.4) y se descartaron las regiones pequeñas, basándose en que la región del bisturí es grande dentro de la imagen reducida. Finalmente, se realizó una operación de dilatación, a fin de atenuar el efecto de la operación de erosión antes realizada. El resultado de estas operaciones es una imagen binaria IMG_{SCL4} que contiene la pseudo-región del bisturí, esta última conformada por píxeles de valor lógico verdadero. Aquella imagen es representada en la figura 3.33 (d).

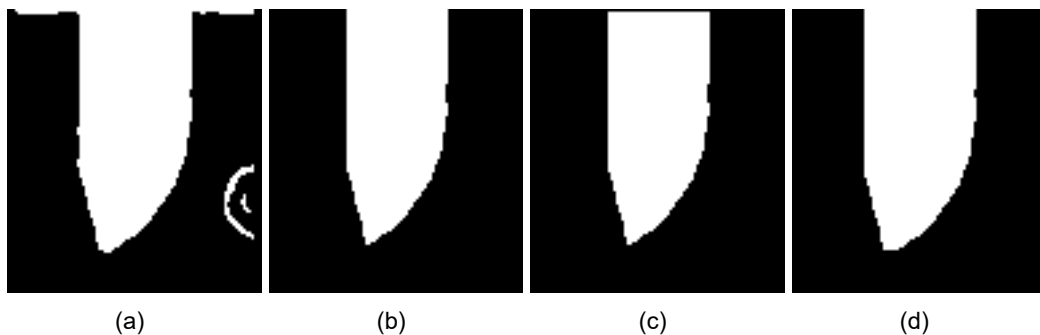


Figura 3.33: (a) Imagen IMG_{SCL1} . (b) Aplicación de erosión. (c) Descarte de grupos pequeños. (d) Aplicación de dilatación.

Cabe resaltar que, si bien en la figura 3.33 se puede ver que las regiones indeseables son descartadas, utilizando únicamente la operación de erosión, experimentalmente se encontró que la mayor presencia de irregularidades dentro de la imagen reducida puede ocasionar la generación de regiones indeseables que perduren a pesar de la operación de erosión. Estas regiones serían agrupadas utilizando el algoritmo de agrupación de píxeles por cuatro barridos (véase la sección 3.4) y luego descartadas por tamaño, tal como se muestra en los tres casos presentados en la figura 3.34.



(a) Ejemplo de caso exitoso número 1.



(b) Ejemplo de caso exitoso número 2.



(c) Ejemplo de caso exitoso número 3.

Figura 3.34: Ejemplos de casos exitosos de obtención de la pseudo-región del bisturí. De izquierda a derecha; imágenes reducidas representadas en escala de grises, imágenes RT , imágenes IMG_{SCL1} , imágenes IMG_{SCL4} .

Por otro lado, la presencia excesiva de irregularidades fuera de la región del bisturí puede ocasionar que dicha región sea confundida con el fondo y, en consecuencia, no se puedan obtener todos los bordes necesarios para definir la pseudo-región del bisturí dentro de un contorno cerrado. En la figura 3.35 se muestran tres ejemplos de estos casos fallidos.



(a) Ejemplo de caso fallido número 1.



(b) Ejemplo de caso fallido número 2.



(c) Ejemplo de caso fallido número 3.

Figura 3.35: Ejemplos de casos fallidos de obtención de la pseudo-región del bisturí. De izquierda a derecha; imágenes reducidas representadas en escala de grises, imágenes RT , imágenes IMG_{SCL1} , imágenes IMG_{SCL4} .

A fin de evitar casos fallidos, como los mostrados en la figura 3.35, se sugiere evitar la excesiva presencia de irregularidades fuera de la región del bisturí en una imagen tomada.

El procedimiento seguido para la obtención de la pseudo-región del bisturí en una imagen reducida es descrito en el algoritmo de la figura 3.36, y tiene las siguientes entradas, salidas y una sub-rutina:

IMG_{SML2} : Imagen reducida.

IMG_{SCLA} : Imagen contenedora de la pseudo-región del bisturí.

Agrupar píxeles por cuatro barridos : Sub-rutina descrita en el algoritmo de la figura 3.21.

Entrada: IMG_{SML2}

Salida: IMG_{SCL4}

- 1: $Rim :=$ obtener el canal R de IMG_{SCL2}
 - 2: $Gim :=$ obtener el canal G de IMG_{SCL2}
 - 3: $Bim :=$ obtener el canal B de IMG_{SCL2}
 - 4: $Sim :=$ obtener el canal S de IMG_{SCL2}
 - 5: $Vim :=$ obtener el canal V de IMG_{SCL2}
 - 6: $Rim :=$ aplicar un filtro media a Rim
 - 7: $Gim :=$ aplicar un filtro media a Gim
 - 8: $Bim :=$ aplicar un filtro media a Bim
 - 9: $Sim :=$ aplicar un filtro media a Sim
 - 10: $Vim :=$ aplicar un filtro media a Vim
 - 11: $Rim :=$ aplicar el detector de bordes Canny a Rim
 - 12: $Gim :=$ aplicar el detector de bordes Canny a Gim
 - 13: $Bim :=$ aplicar el detector de bordes Canny a Bim
 - 14: $Sim :=$ aplicar el detector de bordes Canny a Sim
 - 15: $Vim :=$ aplicar el detector de bordes Canny a Vim
 - 16: $RT := Rim|Gim|Bim|Sim|Vim$
 - 17: convertir el contenido de la primera fila de RT a 1 lógico
 - 18: convertir el contenido de las últimas 3 filas de RT a 0 lógico
 - 19: convertir el contenido de las primeras y últimas 3 columnas de RT a 0 lógico
 - 20: $IMG_{SCL1} :=$ rellenar contornos cerrados en RT
 - 21: $IMG_{SCL2} :=$ aplicar operación erosión a IMG_{SCL1}
 - 22: $IMG_{SCL3} :=$ agrupar píxeles por cuatro barridos de IMG_{SCL2} y descartar regiones pequeñas
 - 23: $IMG_{SCL4} :=$ aplicar operación de dilatación a IMG_{SCL3}
-

Figura 3.36: Algoritmo; obtener la pseudo-región en la imagen reducida.

3.9.1. Algoritmo de obtención de la pseudo-región del bisturí

En esta sección se propone un algoritmo para la obtención de la imagen contenedora de la pseudo-región del bisturí, a partir de una imagen tomada. Este algoritmo se conforma varias sub-rutinas, descritas anteriormente en forma de algoritmos. Además, se agrega el guardado de NR , NG , NB , que son las posiciones de los centros de las últimas pseudo-marcas de color conocidas, las mismas que son obtenidas únicamente cuando las correspondientes pseudo-marcas de color candidatas son confiables. Se agrega también un factor de escalamiento f_2 , ingresado por el usuario, para el cálculo de los parámetros A_{min} y A_{max} utilizados para obtener las pseudo-marcas de color candidatas.

En el presente trabajo de tesis no se establece algún criterio matemático para el cálculo f_2 ; sin embargo, experimentalmente, se encontraron buenos resultados considerando aquel valor igual a 0.7.

El algoritmo para la obtención de la imagen contenedora de la pseudo-región del bisturí, es representado en el algoritmo de la figura 3.37, y tiene las siguientes entradas, salidas y sub-rutinas:

IMG : Imagen tomada.

Rp, Gp, Bp : Estructuras de parámetros comunes de los píxeles de color rojos, verdes y azules, respectivamente.

LR : Posición del centro de la última pseudo-marca de color roja conocida.

LG : Posición del centro de la última pseudo-marca de color verde conocida.

LB : Posición del centro de la última pseudo-marca de color azul conocida.

bas : Longitud de la base de la imagen reducida.

has : Longitud referencial para la altura de la imagen reducida.

dis : Distancia entre el centro de la marca de color verde y el punto medio entre la punta del bisturí y el límite hasta donde la hoja del bisturí puede ingresar verticalmente a la muestra.

f₂ : Factor de tolerancia de áreas.

IMG_{SCLA} : Imagen contenedora de la pseudo-región del bisturí.

NR : Posición del centro de la nueva pseudo-marca de color roja conocida.

NG : Posición del centro de la nueva pseudo-marca de color verde conocida.

NB : Posición del centro de la nueva pseudo-marca de color azul conocida.

Agrupar píxeles por cuatro barridos : Sub-rutina descrita en el algoritmo de la figura 3.21.

Obtener la pseudo-marca candidata : Sub-rutina descrita en el algoritmo de la figura 3.22.

Evaluar la confiabilidad : Sub-rutina descrita en el algoritmo de la figura 3.24.

Obtener la imagen reducida : Sub-rutina descrita en el algoritmo de la figura 3.28.

Obtener la pseudo-región en la imagen reducida : Sub-rutina descrita en el algoritmo de la figura 3.36.

Entrada: $IMG, Rp, Gp, Bp, LR, LG, LB, bas, has, dis, dms, f_2$

Salida: IMG_{SCL4}, NR, NG, NB

- 1: $M_{REST} :=$ matriz de 3 filas y 3 columnas
- 2: $IMG_{RGB} \leftarrow$ detectar píxeles de color (IMG, Rp, Gp, Bp)
- 3: $IMG_{RGB} :=$ aplicar operación erosión a IMG_{RGB}
- 4: $IMG_{RGB} :=$ aplicar operación dilatación a IMG_{RGB}
- 5: $RI :=$ obtener canal R de IMG_{RGB}
- 6: $GI :=$ obtener canal G de IMG_{RGB}
- 7: $BI :=$ obtener canal B de IMG_{RGB}
- 8: $A := (Rp.Am + Gp.Am + Bp.Am)/3$
- 9: $A_{min} := A \cdot f_2$
- 10: $A_{max} := A/f_2$
- 11: $GR \leftarrow$ agrupar píxeles por cuatro barridos (RI)
- 12: $RI \leftarrow$ obtener la pseudo-marca candidata ($RI, GR, A_{min}, A_{max}, LR, dms$)
- 13: $GR \leftarrow$ agrupar píxeles por cuatro barridos (GI)
- 14: $GI \leftarrow$ obtener la pseudo-marca candidata ($GI, GR, A_{min}, A_{max}, LG, dms$)
- 15: $GR \leftarrow$ agrupar píxeles por cuatro barridos (BI)
- 16: $BI \leftarrow$ obtener la pseudo-marca candidata ($BI, GR, A_{min}, A_{max}, LB, dms$)
- 17: $IMG_{RGB} :=$ asignar RI, GI, BI a los canales R, G, B de IMG_{RGB}
- 18: $M_{REST} \leftarrow$ evaluar la confiabilidad (IMG_{RGB}, Rp, Gp, Bp)
- 19: $IMG_{SML2} \leftarrow$ obtener la imagen reducida ($IMG, M_{REST}, bas, has, dms$)
- 20: $IMG_{SCL4} \leftarrow$ obtener la pseudo-región en la imagen reducida (IMG_{SML2})
- 21: **si** $M_{REST}(1, 1) = 1$ **entonces**
- 22: $NR(1) := M_{REST}(2, 1)$
- 23: $NR(2) := M_{REST}(3, 1)$
- 24: **si no**
- 25: $NR := LR$
- 26: **fin si**
- 27: **si** $M_{REST}(1, 2) = 1$ **entonces**
- 28: $NG(1) := M_{REST}(2, 2)$
- 29: $NG(2) := M_{REST}(3, 2)$
- 30: **si no**
- 31: $NG := LG$
- 32: **fin si**
- 33: **si** $M_{REST}(1, 3) = 1$ **entonces**
- 34: $NB(1) := M_{REST}(2, 3)$
- 35: $NB(2) := M_{REST}(3, 3)$
- 36: **si no**
- 37: $NB := LB$
- 38: **fin si**

Figura 3.37: Algoritmo; obtener la pseudo-región del bisturí.

3.10. Estimación del grado de penetración del bisturí

En el presente trabajo de tesis se utiliza la profundidad de corte para medir el grado de penetración del bisturí. En la ecuación (3.1), se calcula la profundidad de corte como la

diferencia entre la longitud total de la hoja del bisturí y la del lado expuesto instantáneamente. Siendo a su vez esta diferencia, igual a la distancia entre la punta del bisturí y el borde que separa la región del bisturí, de la muestra, una vez la hoja hubiera penetrado en la muestra. Se entiende que mientras que la hoja del bisturí se encuentre fuera de la muestra, la posición de su punta será igual a la posición del punto más bajo de la región del bisturí. Así mismo, cuando la hoja hubiera penetrado en la muestra, el punto más bajo de la región del bisturí formaría parte del borde que la separa la región del bisturí, de la muestra. Este escenario es representado en la figura 3.38. En ella se define P como la punta de la pseudo-región del bisturí, mientras que se define P' como el punto más bajo de la pseudo-región del bisturí, ambas posiciones medidas IMG_{SCLA} (imagen contenedora de la pseudo-región del bisturí). En ese sentido, se estima la profundidad de corte pc como la distancia vertical entre P y P' de acuerdo a la ecuación (3.14).

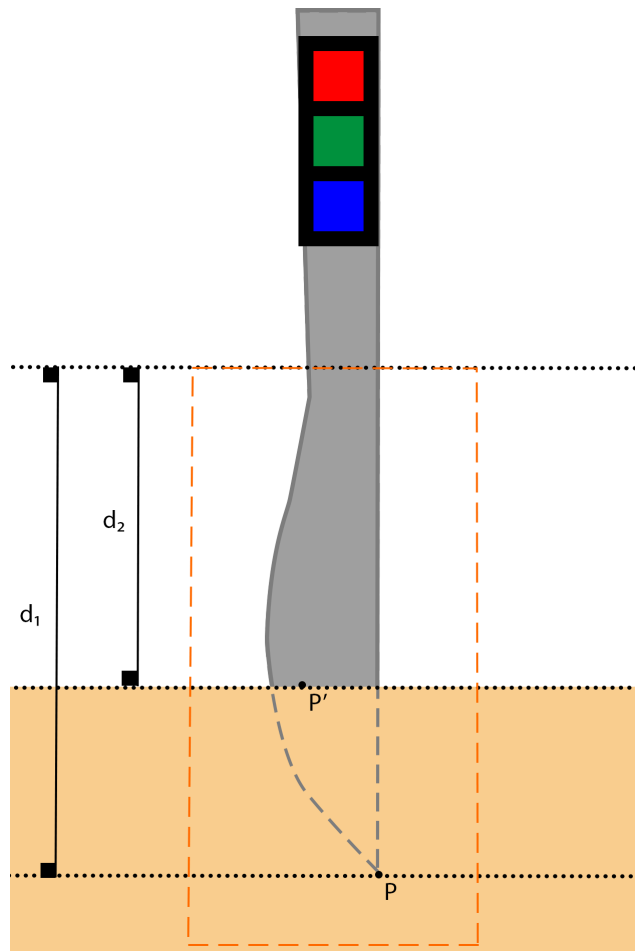


Figura 3.38: Representación del bisturí penetrando en la muestra.

Donde:

d_1 : Posición en el eje vertical de P en IMG_{SCLA} .

d_2 : Posición en el eje vertical de P' en IMG_{SCLA} .

$$pc = d_1 - d_2 \quad (3.14)$$

Cabe resaltar que, idealmente, P y P' coinciden siempre que el bisturí se encuentre fuera de la muestra. Es por ello que se calcula d_2 como la posición en el eje vertical del punto más bajo de la pseudo-región del bisturí y se calcula d_1 igual a d_2 cuando el bisturí se encuentre fuera de la muestra.

El procedimiento seguido para calcular P' , es descrito en el algoritmo de la figura 3.39 y tiene las siguientes entradas y salidas:

IMG_{SCLA} : Imagen contenedora de la pseudo-región del bisturí.

xm : Posición en el eje vertical de P' .

ym : Posición en el eje horizontal de P' .

Entrada: IMG_{SCLA}

Salida: xm, ym

```

1:  $fil :=$  cantidad de filas de  $IMG_{SCLA}$ 
2:  $col :=$  cantidad de columnas de  $IMG_{SCLA}$ 
3:  $X :=$  vector
4:  $Y :=$  vector
5:  $ma := 0$ 
6:  $xm := 0$ 
7:  $ym := 0$ 
8: para  $j \leftarrow 1, col$  hacer
9:    $X(j) = j$ 
10:   $Y(j) = 0$ 
11:  para  $i \leftarrow 1, fil$  hacer
12:    si  $IMG_{SCLA}(i, j) > 0$  entonces
13:       $Y(j) := i$ 
14:    fin si
15:    si  $Y(j) > ma$  entonces
16:       $xm = X(j)$ 
17:       $ym = Y(j)$ 
18:       $ma = ym$ 
19:    fin si
20:  fin para
21: fin para

```

Figura 3.39: Algoritmo; obtener el punto más bajo de la pseudo-región del bisturí.

3.11. Control de la profundidad de corte con bisturí

Esta sección tiene como objetivo proponer una técnica para el control de la profundidad de corte con bisturí. Para ello, se compararon tres técnicas de control diferentes con la finalidad de elegir una de ellas.

La comparación se basó en el cumplimiento de las especificaciones de control descritas en la sección 3.1.3. En ella se indica que, en este trabajo, se considera que una técnica de control funciona satisfactoriamente si la respuesta del sistema controlado con dicha técnica, ante una entrada de tipo escalón de 5 milímetros de amplitud, presenta un t_s (tiempo de asentamiento) menor o igual a 90 segundos y una M_p (sobreeLONGACIÓN MÁXIMA PORCENTUAL) menor o igual al 25 %.

Es importante resaltar que todas las técnicas de control indicadas en el presente trabajo fueron probadas en tiempo discreto. Esto, debido a que las técnicas fueron implementadas como parte de un programa de computadora que fue basado en la ejecución de un bucle. Siendo así, la estimación de la profundidad de corte y el cálculo de la señal de control se realizaron durante cada una de las iteraciones del bucle. Del mismo modo, las operaciones matemáticas de control no se realizaron en función a la integral o a la derivada del error de control, sino, en su lugar se utilizaron la suma se_k y la diferencia de_k del error de control e_k . Estas variables son calculadas en cada iteración, en base a la ecuación (3.15).

$$\begin{aligned}e_k &:= ref_k - pc_k \\se_k &:= se_{k-1} + e_k T_s \\de_k &:= \frac{e_k - e_{k-1}}{T_s}\end{aligned}\tag{3.15}$$

Dónde:

ref_k : Referencia de control.

pc_k : Profundidad de corte estimada.

e_k : Error de control.

e_{k-1} : Error de control en la iteración anterior.

se_k : Suma del error de control.

se_{k-1} : Suma del error de control hasta la iteración anterior.

de_k : Diferencia del error de control.

T_s : Tiempo de muestreo.

3.11.1. Aplicación de un controlador proporcional integral derivativo

Se eligió como primera alternativa de técnica de control un controlador proporcional integral derivativo, también denominado controlador PID. En la figura 3.40 se esquematiza el sistema, SRobot + Estimador de la profundidad de corte, controlado con esta técnica.

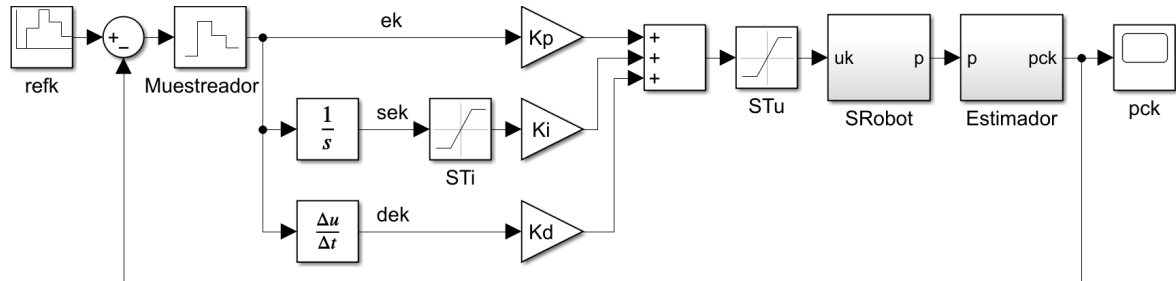
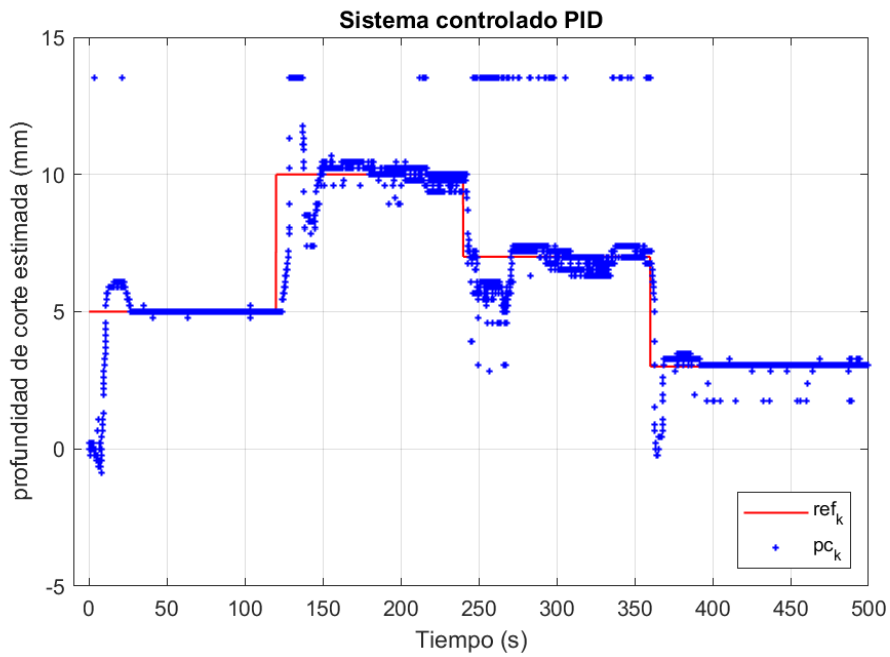


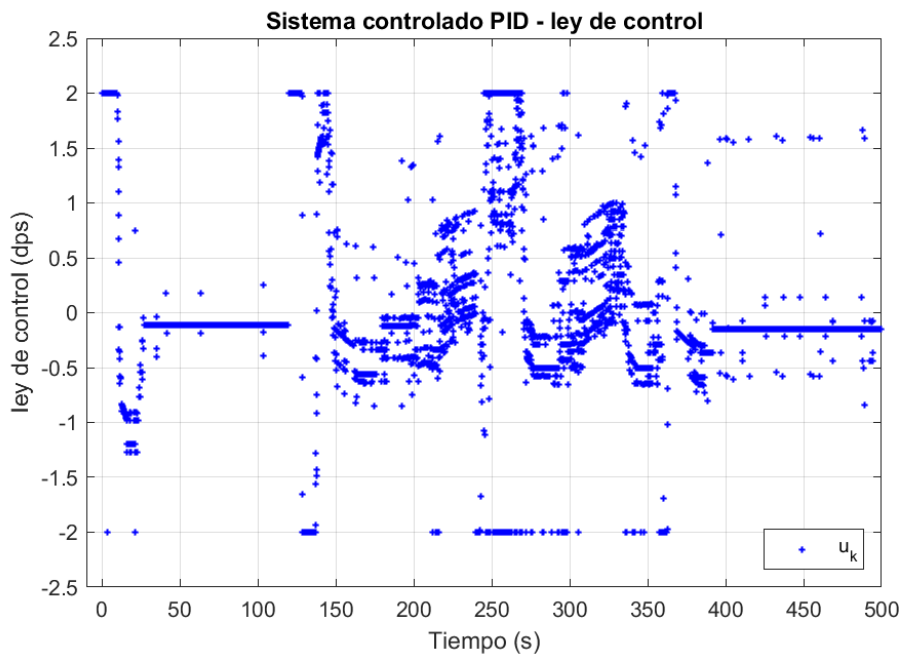
Figura 3.40: Esquema de la aplicación del controlador PID.

La figura 3.40 SRobot (véase la sección 3.1.1) representa el sistema que se desea controlar. Por su parte, el bloque Estimador representa todas las acciones tomadas para estimar la profundidad de corte, mientras que el bloque Muestreador representa la acción de realizar el muestreo dentro del lazo de control. En la misma figura, los operadores de saturación ST_i y ST_u tienen como finalidad limitar los valores que adopten las señales se_k (suma del error de control) y u_k (señal de control) a intervalos de previamente establecidos. Estos saturadores fueron agregados, debido a que, si bien teóricamente se_k y u_k pueden tomar cualquier valor numérico, en la práctica el hardware empleado puede no operar correctamente para ciertos valores de ambas señales, fundamentalmente valores extremadamente altos. En este trabajo no se establece algún criterio para calcular los límites de saturación de dichos saturadores. Sin embargo, se sugiere que los límites de ST_i sean determinados tomando en cuenta que valores muy elevados de se_k pueden ser consecuencia de que, al no conocerse la distancia entre la punta del bisturí y la superficie de la muestra, al comenzar las acciones de control. Tampoco se conoce el tiempo que tarde la punta en llegar a la superficie, tiempo durante el cual e_k se mantiene constante y se_k aumenta a ritmo constante pudiendo alcanzar valores muy elevados que se desea evitar. Por su parte, los valores de los límites de ST_u deben ser tales que eviten que u_k adopte valores que estén fuera del rango de operación de SRobot.

A fin de verificar el funcionamiento del controlador PID, se implementó el mismo en un programa de computadora, en el que las ganancias de control fueron reguladas manualmente. En la figura 3.41 (a), se muestra los resultados de esta implementación, para la secuencia de referencias de profundidad de corte de 5mm, 10mm, 7mm, y 3mm, para una ley de control limitada al intervalo de $[-2, 2]$ dps (grados sexagesimales por segundo).



(a) Referencia de control ref_k y profundidad estimada de corte pc_k .



(b) Ley de control u_k .

Figura 3.41: Profundidad de corte controlada con el controlador PID.

Esta técnica de control, expresada en tiempo discreto, es descrita en el algoritmo de la figura 3.42, y tiene las siguientes entradas y salidas:

e_k : Error de control.

se_k : Suma del error de control saturada.

de_k : Diferencia del error de control.

Kp : Ganancia proporcional del controlador PID.
 Ki : Ganancia integral del controlador PID.
 Kd : Ganancia derivativa del controlador PID.
 u : Salida del controlador PID.

Entrada: $e_k, se_k, de_k, Kp, Ki, Kd$

Salida: u

$$1: u := Kp \cdot e_k + Ki \cdot se_k + Kd \cdot de_k$$

Figura 3.42: Algoritmo; controlador PID.

Se extrajeron los primeros 120 segundos de los resultados mostrados en la figura 3.41 (a), que corresponden a la respuesta del sistema controlado ante una entrada de tipo escalón de 5mm de amplitud. De esta respuesta, representada en la figura 3.43, se obtuvo un t_s de 26.04s y una M_p de 21.99%, valores que cumplen con las especificaciones planteadas en la sección 3.1.3. Por tal motivo, en este trabajo se considera que se ha corroborado que el controlador PID funciona satisfactoriamente.

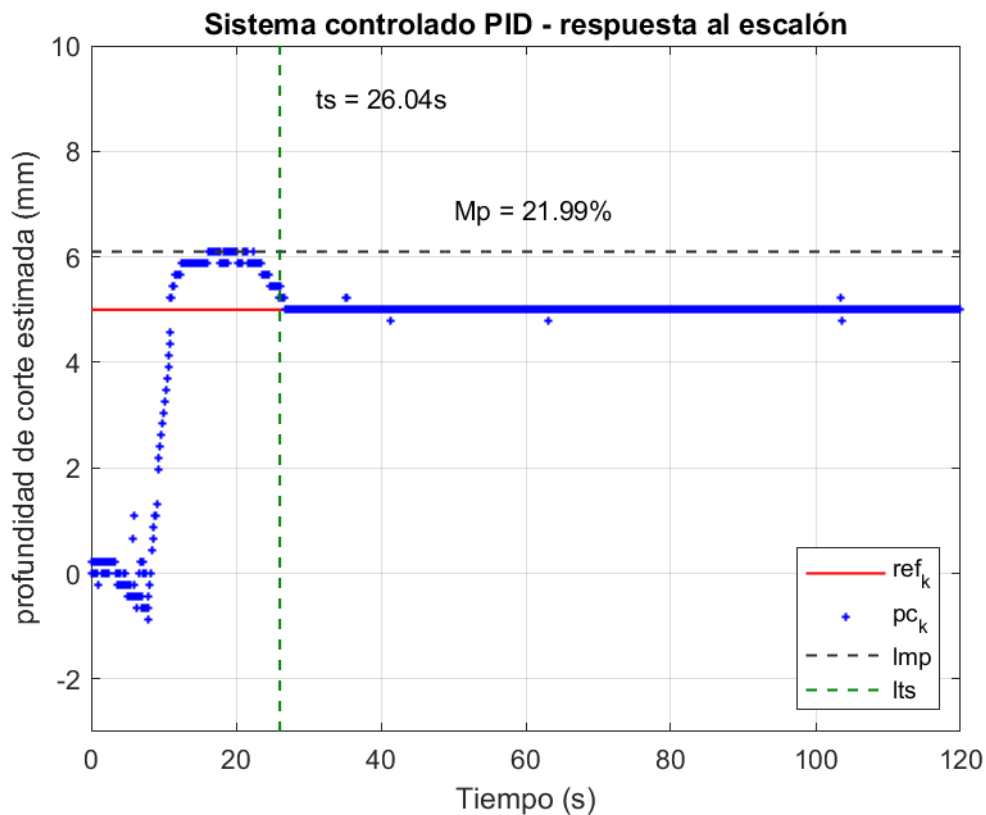


Figura 3.43: Respuesta del sistema controlado PID ante una entrada de tipo escalón de 5mm de amplitud.

Dónde:

ref_k : Referencia de control de profundidad de corte.

pc_k : Profundidad de corte estimada.

lts : Línea vertical que señala t_s .

lmp : Línea horizontal que señala M_p .

3.11.2. Aplicación de un controlador difuso

Se eligió como segunda alternativa de técnica de control un controlador difuso. Este controlador se basa en el uso de una máquina de 25 reglas difusas, extraídas de [18]. Por su parte, los conjuntos difusos son versiones triangulares normalizadas a dominios de -1 a 1, de los conjuntos difusos descritos en [30].

Esta máquina de reglas difusas, en adelante denominada *FRM*, utiliza como reglas silogismos de tipo Mamdani y, además, utiliza el método del centroide para la defuzificación. La máquina de reglas difusas tiene como variables de entrada e y ce , que se calculan escalando el error (con una ganancia f_e) y la diferencia del error de control de profundidad de corte (con una ganancia f_{ce}) respectivamente. Por su parte, *FRM* tiene como variable de salida u , que posteriormente es escalada para calcular la señal de control. En la figura 3.44 se esquematiza el sistema, SRobot + Estimador de la profundidad de corte, controlado con esta técnica.

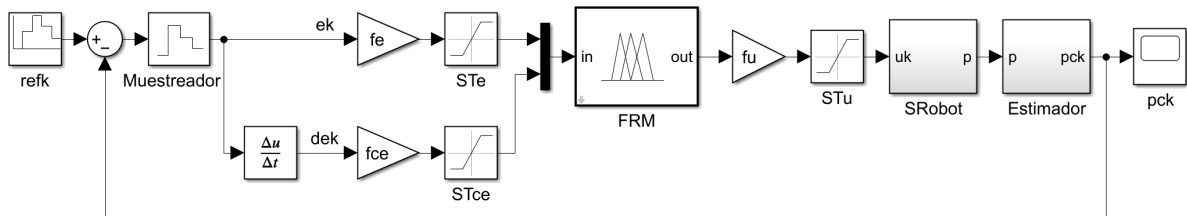


Figura 3.44: Esquema de la aplicación del controlador difuso.

En la figura 3.44 SRobot (véase la sección 3.1.1) representa el sistema que se desea controlar. Por su parte, el bloque Estimator representa todas las acciones tomadas para estimar la profundidad de corte, mientras que el bloque Muestreador representa la acción de realizar el muestreo dentro del lazo de control. En la misma figura, los operadores de saturación ST_e , ST_{ce} tienen como finalidad limitar los valores que adopten las señales las señales e_k (error de control) y de_k (diferencia del error de control) al intervalo de 0 a 1, que es el intervalo de valores de las variables de entrada de *FRM*, y en consecuencia ambos saturadores utilizan como límite inferior y límite superior, los valores de 0 y 1 respectivamente. Por otro lado, ST_u tiene como finalidad limitar los valores que adopte u_k . En este trabajo

no se establece algún criterio para calcular los límites de saturación de ST_u ; sin embargo, se sugiere tomar en cuenta que los valores de los límites de ST_u deben ser tales que eviten que u_k adopte valores que estén fuera del rango de operación SRobot.

En *FRM*, las variables e , ce , y u , son operadas con respecto a los conjuntos difusos NL , NS , ZE , PS , y PL . Estos conjuntos son representados en la figura 3.45.

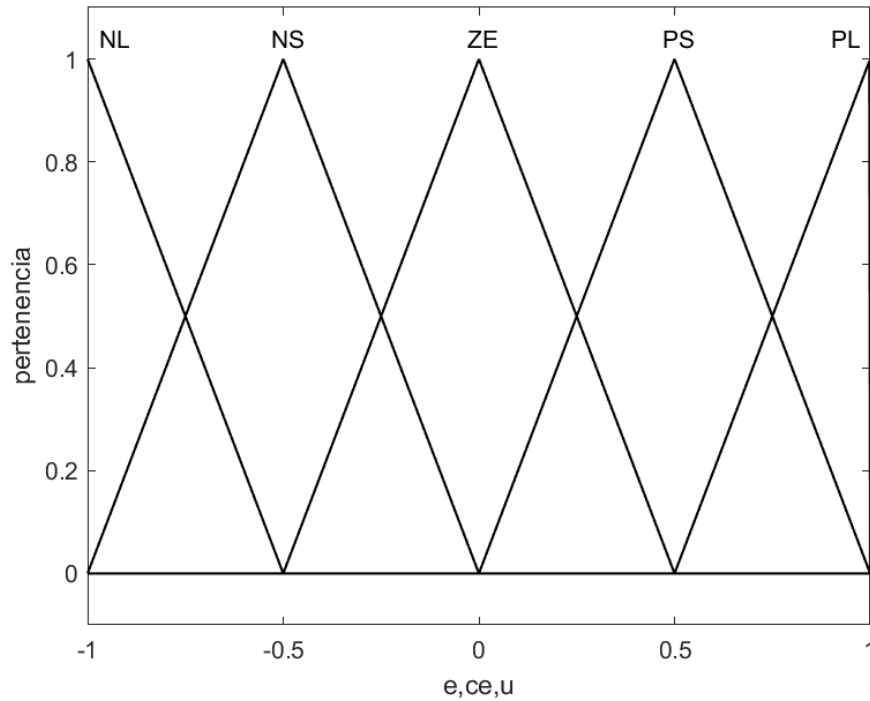


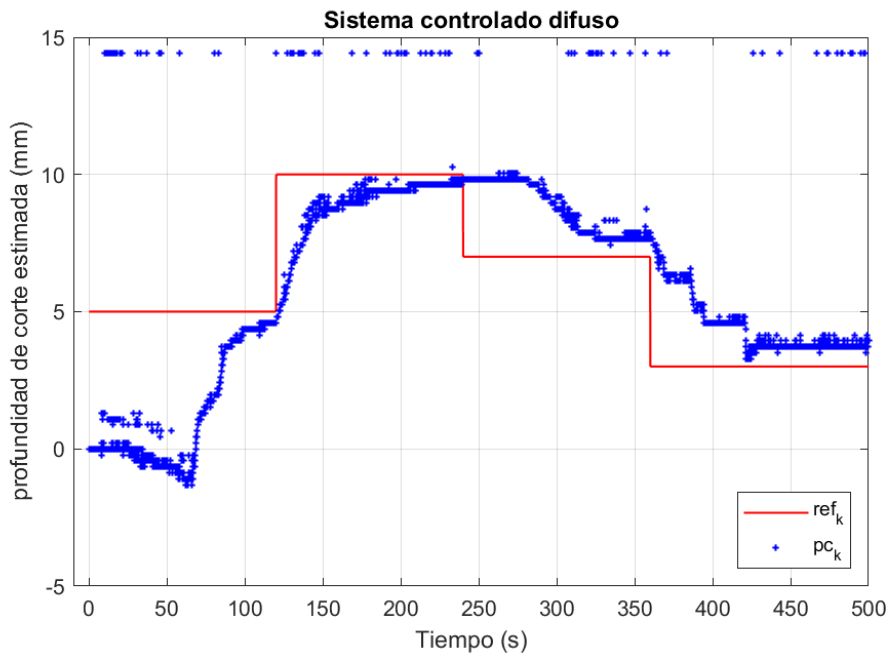
Figura 3.45: Conjuntos difusos para e , ce y u .

FRM, se compone de las 25 reglas difusas descritas en la tabla 3.1.

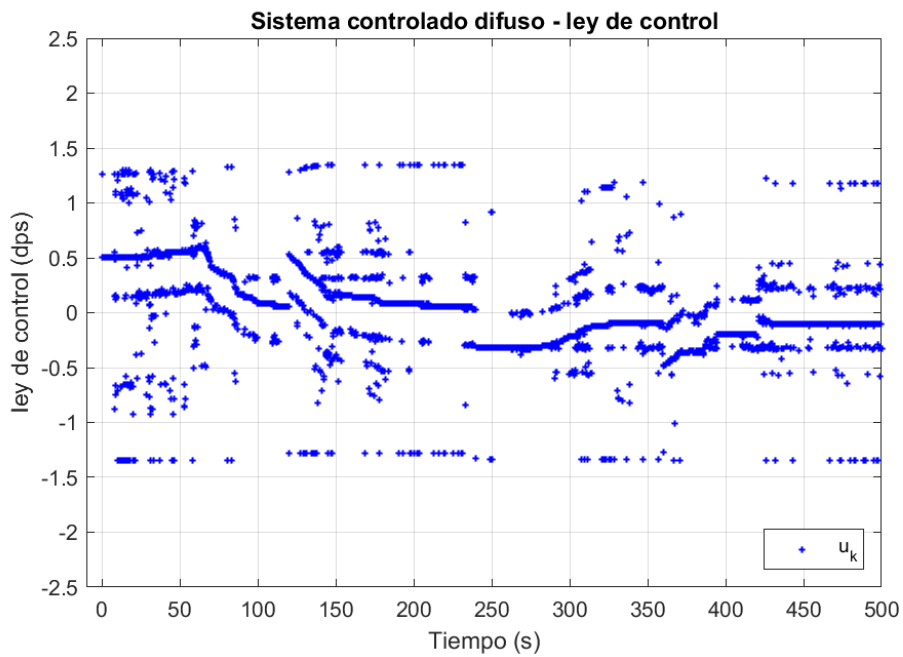
Nombre	Regla
R1	Si e es NL y ce es NL , entonces u es NL
R2	Si e es NL y ce es NS , entonces u es NL
R3	Si e es NL y ce es ZE , entonces u es NL
R4	Si e es NL y ce es PS , entonces u es NS
R5	Si e es NL y ce es PL , entonces u es ZE
R6	Si e es NS y ce es NL , entonces u es NL
R7	Si e es NS y ce es NS , entonces u es NS
R8	Si e es NS y ce es ZE , entonces u es NS
R9	Si e es NS y ce es PS , entonces u es ZE
R10	Si e es NS y ce es PL , entonces u es PS
R11	Si e es ZE y ce es NL , entonces u es NL
R12	Si e es ZE y ce es NS , entonces u es NS
R13	Si e es ZE y ce es ZE , entonces u es ZE
R14	Si e es ZE y ce es PS , entonces u es PS
R15	Si e es ZE y ce es PL , entonces u es PL
R16	Si e es PS y ce es NL , entonces u es NS
R17	Si e es PS y ce es NS , entonces u es ZE
R18	Si e es PS y ce es ZE , entonces u es PS
R19	Si e es PS y ce es PS , entonces u es PS
R20	Si e es PS y ce es PL , entonces u es PL
R21	Si e es PL y ce es NL , entonces u es ZE
R22	Si e es PL y ce es NS , entonces u es PS
R23	Si e es PL y ce es ZE , entonces u es PL
R24	Si e es PL y ce es PS , entonces u es PL
R25	Si e es PL y ce es PL , entonces u es PL

Tabla 3.1: Reglas difusas de $FRM1$.

A fin de verificar el funcionamiento del controlador difuso, se implementó el mismo en un programa de computadora. En la figura 3.46 (a) se muestra los resultados de esta implementación para la secuencia de referencias de profundidad de corte de 5mm, 10mm, 7mm, y 3mm, para una ley de control limitada al intervalo de $[-2, 2]$ dps.



(a) Referencia de control ref_k y profundidad estimada de corte pc_k .



(b) Ley de control u_k .

Figura 3.46: Profundidad de corte controlada con el controlador difuso.

Esta técnica de control, expresada en tiempo discreto, es descrita en el algoritmo de la figura 3.47, y tiene las siguientes entradas, salidas, y sub-rutinas:

- e : Error de control escalado y saturado.
- ce : Diferencia del error de control escalada y saturada.
- NL : Vector de 3 elementos que define el conjunto difuso triangular NL .

NS : Vector de 3 elementos que define el conjunto difuso triangular NS .

ZE : Vector de 3 elementos que define el conjunto difuso triangular ZE .

PS : Vector de 3 elementos que define el conjunto difuso triangular PS .

PL : Vector de 3 elementos que define el conjunto difuso triangular PL .

u : Salida de FRM .

$rule$: Implicación difusa de Mamdani para conjuntos difusos triangulares.

Entrada: $e, ce, NL, NS, ZE, PS, PL$

Salida: u

```
1:  $V := \text{vector}$ 
2:  $nv := \text{obtener la cantidad de elementos de } V$ 
3:  $dt := 2/(nv)$ 
4:  $t := -1$ 
5:  $PA := 0$ 
6:  $A := 0$ 
7:  $V_1 \leftarrow \text{rule}(NL, NL, NL, e, ce)$ 
8:  $V_2 \leftarrow \text{rule}(NL, NS, NL, e, ce)$ 
9:  $V_3 \leftarrow \text{rule}(NL, ZE, NL, e, ce)$ 
10:  $V_4 \leftarrow \text{rule}(NL, PS, NS, e, ce)$ 
11:  $V_5 \leftarrow \text{rule}(NL, PL, ZE, e, ce)$ 
12:  $V_6 \leftarrow \text{rule}(NS, NL, NL, e, ce)$ 
13:  $V_7 \leftarrow \text{rule}(NS, NS, NS, e, ce)$ 
14:  $V_8 \leftarrow \text{rule}(NS, ZE, NS, e, ce)$ 
15:  $V_9 \leftarrow \text{rule}(NS, PS, ZE, e, ce)$ 
16:  $V_{10} \leftarrow \text{rule}(NS, PL, PS, e, ce)$ 
17:  $V_{11} \leftarrow \text{rule}(ZE, NL, NL, e, ce)$ 
18:  $V_{12} \leftarrow \text{rule}(ZE, NS, NS, e, ce)$ 
19:  $V_{13} \leftarrow \text{rule}(ZE, ZE, ZE, e, ce)$ 
20:  $V_{14} \leftarrow \text{rule}(ZE, PS, PS, e, ce)$ 
21:  $V_{15} \leftarrow \text{rule}(ZE, PL, PL, e, ce)$ 
22:  $V_{16} \leftarrow \text{rule}(PS, NL, NS, e, ce)$ 
23:  $V_{17} \leftarrow \text{rule}(PS, NS, ZE, e, ce)$ 
24:  $V_{18} \leftarrow \text{rule}(PS, ZE, PS, e, ce)$ 
25:  $V_{19} \leftarrow \text{rule}(PS, PS, PS, e, ce)$ 
26:  $V_{20} \leftarrow \text{rule}(PS, PL, PL, e, ce)$ 
27:  $V_{21} \leftarrow \text{rule}(PL, NL, ZE, e, ce)$ 
28:  $V_{22} \leftarrow \text{rule}(PL, NS, PS, e, ce)$ 
29:  $V_{23} \leftarrow \text{rule}(PL, ZE, PL, e, ce)$ 
30:  $V_{24} \leftarrow \text{rule}(PL, PS, PL, e, ce)$ 
31:  $V_{25} \leftarrow \text{rule}(PL, PL, PL, e, ce)$ 
32: para  $i \leftarrow 1, nv$  hacer
33:    $V(i) := \text{obtener el máximo de entre } V_1(i), V_2(i), \dots, V_{25}(i)$ 
34:    $PA := PA + t \cdot dt \cdot V(i)$ 
35:    $A := A + dt \cdot V(i)$ 
36:    $t := t + dt$ 
37: fin para
38: si  $A = 0$  entonces
39:    $u := 0$ 
40: si no
41:    $u := PA/A$ 
42: fin si
```

Figura 3.47: Algoritmo; controlador difuso.

Tal como se puede ver, el algoritmo de la figura 3.47 utiliza la sub-rutina *rule*. Esta es

descrita en el anexo A.2.

Se extrajeron los primeros 120 segundos de los resultados mostrados en la figura 3.46 (a), que corresponden a la respuesta del sistema controlado ante una entrada de tipo escalón de 5mm de amplitud. De esta respuesta, representada en la figura 3.48, no se tiene información suficiente para corroborar que se ha alcanzado el estado estacionario y, en consecuencia, no se obtuvieron valores para t_s y M_p . Por tal motivo, en este trabajo se considera que no se ha corroborado que el controlador difuso funcione satisfactoriamente.

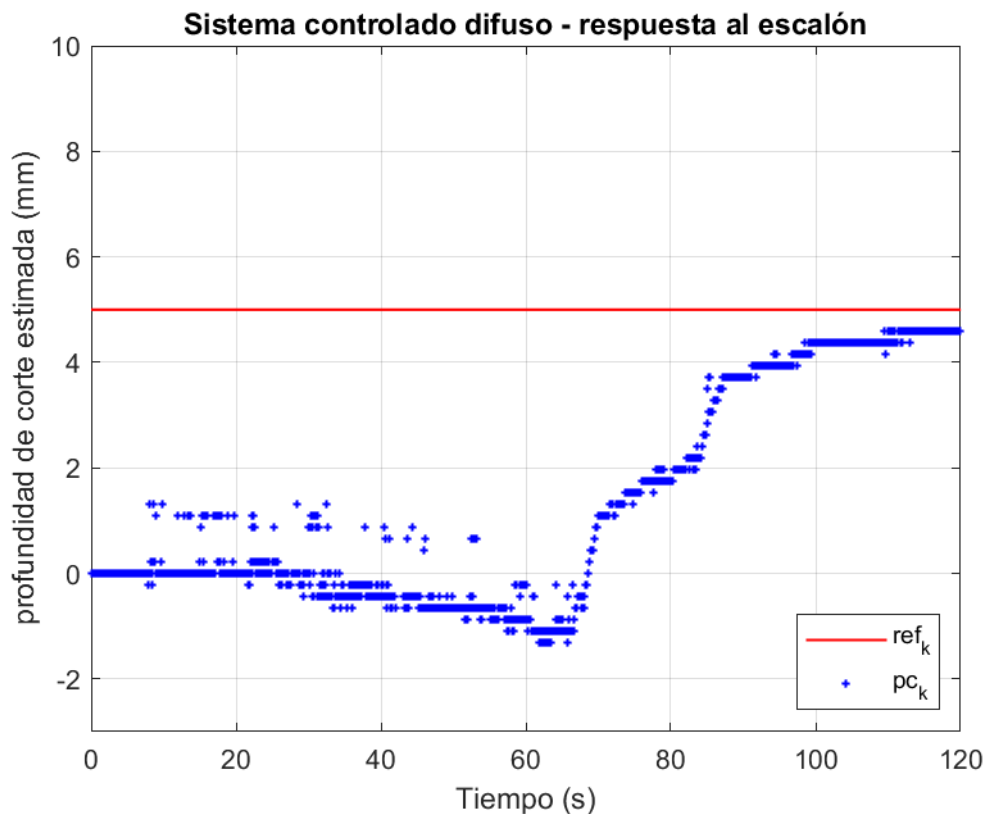


Figura 3.48: Respuesta del sistema controlado difuso ante una entrada de tipo escalón de 5mm de amplitud.

Dónde:

ref_k : Referencia de control de profundidad de corte.

pc_k : Profundidad de corte estimada.

3.11.3. Aplicación de un controlador proporcional integral difuso

Se eligió como tercera alternativa de técnica de control un controlador proporcional integral difuso, también denominado controlador PI-difuso. Este controlador, que es una adaptación del controlador empleado en [17], es un controlador proporcional integral en el

que cada una de las ganancias de control es calculada utilizando una máquina de reglas difusas diferente. Estas máquinas de reglas difusas, denominadas en adelante $FRMp$ y $FRMi$, tienen como variables de entrada e y ua , que a su vez se calculan escalando el error de control (con una ganancia fie) y la señal de control en la interacción anterior del lazo de control (con una ganancia fiu) respectivamente. Por su parte, las salidas de las máquinas de reglas difusas, op y oi corresponden a $FMRp$ y $FMRi$ respectivamente. Estas salidas son posteriormente escaladas para calcular las ganancias Kp (con una ganancia gp) y Ki (con una ganancia gi), tal como se esquematiza en la figura 3.49.

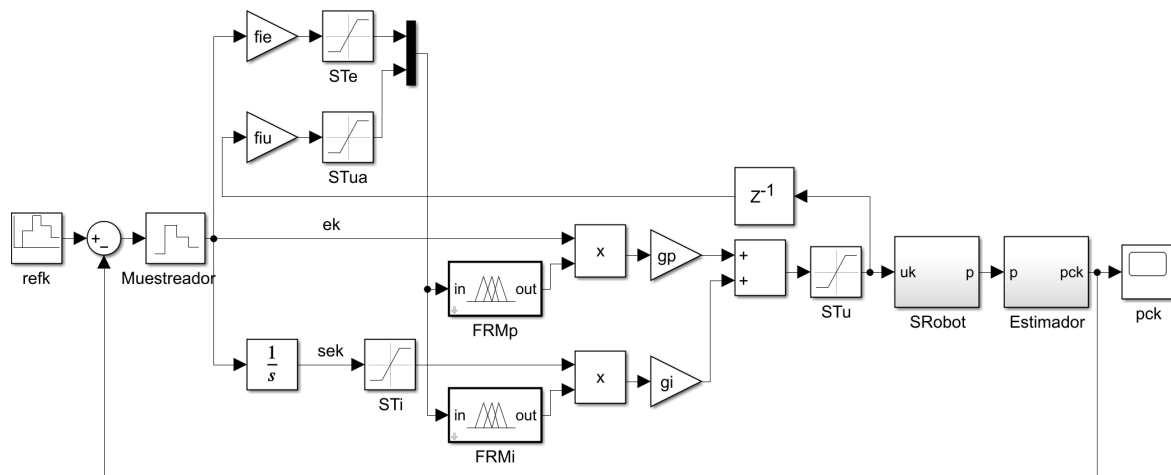


Figura 3.49: Esquema de la aplicación del control PI-difuso.

En la figura 3.49 SRobot (véase la sección 3.1.1) representa el sistema que se desea controlar. Por su parte, el bloque Estimador representa todas las acciones tomadas para estimar la profundidad de corte, mientras que el bloque Muestreador representa la acción de realizar el muestreo dentro del lazo de control. En la misma figura, los operadores de saturación ST_e y ST_{ua} tienen como finalidad limitar los valores que adopten las señales e y ua al intervalo de -1 a 1 , que es el intervalo de valores de las variables de entrada de $FRMp$ y $FRMi$, y, en consecuencia, ambos saturadores utilizan como límite inferior y límite superior los valores de -1 y 1 respectivamente. Por su parte, los saturadores ST_i y ST_u tienen como finalidad limitar los valores que adopten las señales se_k (suma del error de control) y u_k (señal de control) a intervalos de previamente establecidos. Estos saturadores fueron agregados, debido a que, si bien teóricamente se_k y u_k pueden tomar cualquier valor numérico, en la práctica el hardware empleado puede no operar correctamente para ciertos valores de ambas señales, fundamentalmente valores extremadamente altos. En este trabajo no se establece algún criterio para calcular los límites de saturación de dichos saturadores. Sin embargo, se sugiere que los límites de ST_i sean determinados tomando en cuenta que valores muy elevados de se_k pueden ser consecuencia de que, al no conocerse

la distancia entre la punta del bisturí y la superficie de la muestra, al comenzar las acciones de control. Tampoco se conoce el tiempo que tarde la punta en llegar a la superficie, tiempo durante el cual e_k se mantiene constante y se_k aumenta a ritmo constante pudiendo alcanzar valores muy elevados que se desea evitar. Por su parte, los valores de los límites de ST_u deben ser tales que eviten que u_k adopte valores que estén fuera del rango de operación de SRobot.

En ambas máquinas de reglas difusas, la entrada e es operada con respecto a los conjuntos difusos NB_{eK} , NS_{eK} , ZE_{eK} , PS_{eK} , y PB_{eK} , mientras que la entrada ua es operada con respecto a los conjuntos difusos LOW_{uK} , MED_{uK} , HIG_{uK} . Por su parte, las salidas op y oi son operadas con respecto a los conjuntos difusos $VLOW_K$, LOW_K , MED_K , HIG_K , y $VHIG_K$. Estos conjuntos difusos se representan en las figuras 3.50, 3.51, 3.52.

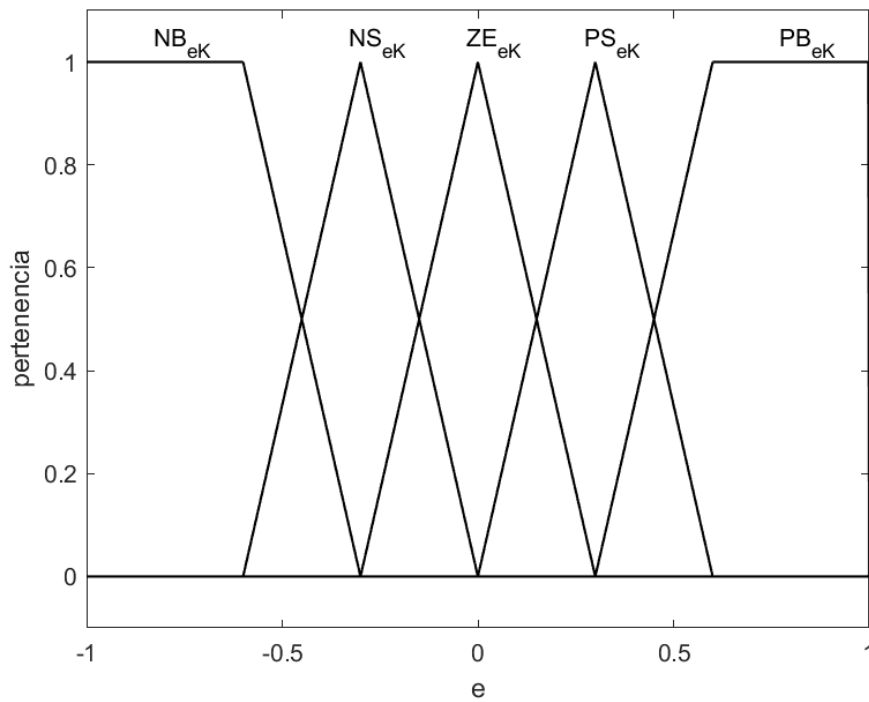


Figura 3.50: Conjuntos difusos para e .

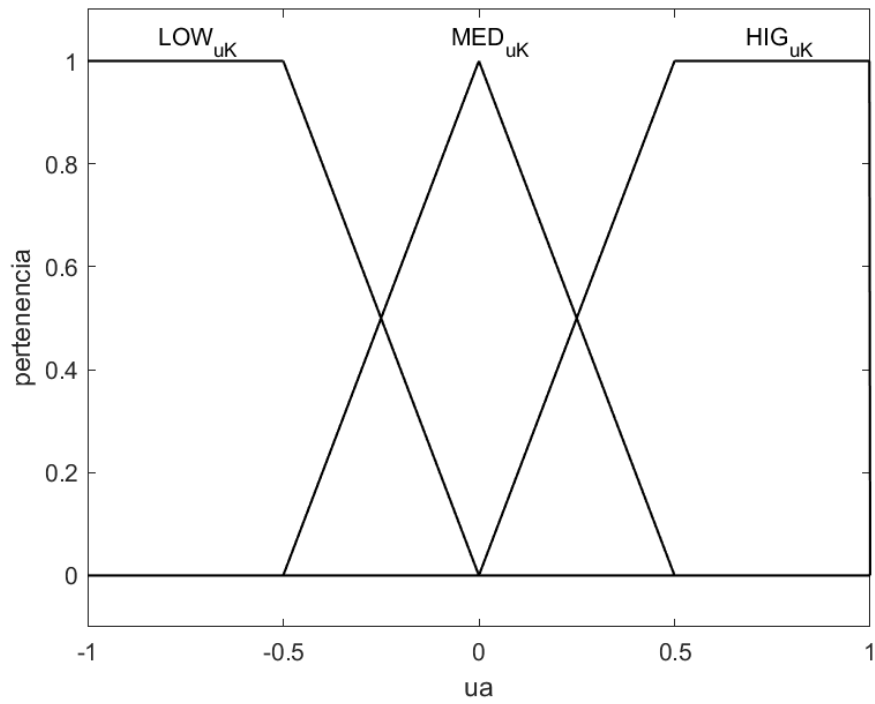


Figura 3.51: Conjuntos difusos para ua .

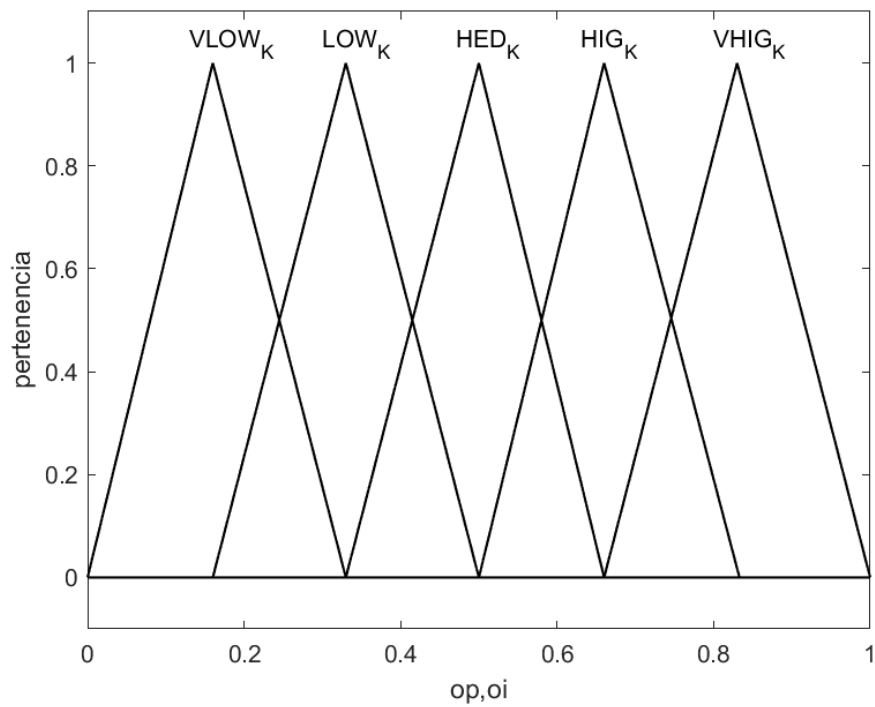


Figura 3.52: Conjuntos difusos para op y oi .

$FRMp$, se compone de las 15 reglas difusas descritas en la tabla 3.2.

Nombre	Regla
R1	Si e es NB_{eK} y ua es LOW_{uK} , entonces op es $VHIG_K$
R2	Si e es NB_{eK} y ua es MED_{uK} , entonces op es HIG_K
R3	Si e es NB_{eK} y ua es HIG_{uK} , entonces op es MED_K
R4	Si e es NS_{eK} y ua es LOW_{uK} , entonces op es HIG_K
R5	Si e es NS_{eK} y ua es MED_{uK} , entonces op es MED_K
R6	Si e es NS_{eK} y ua es HIG_{uK} , entonces op es LOW_K
R7	Si e es ZE_{eK} y ua es LOW_{uK} , entonces op es MED_K
R8	Si e es ZE_{eK} y ua es MED_{uK} , entonces op es LOW_K
R9	Si e es ZE_{eK} y ua es HIG_{uK} , entonces op es $VLOW_K$
R10	Si e es PS_{eK} y ua es LOW_{uK} , entonces op es LOW_K
R11	Si e es PS_{eK} y ua es MED_{uK} , entonces op es LOW_K
R12	Si e es PS_{eK} y ua es HIG_{uK} , entonces op es $VLOW_K$
R13	Si e es PB_{eK} y ua es LOW_{uK} , entonces op es MED_K
R14	Si e es PB_{eK} y ua es MED_{uK} , entonces op es HIG_K
R15	Si e es PB_{eK} y ua es HIG_{uK} , entonces op es HIG_K

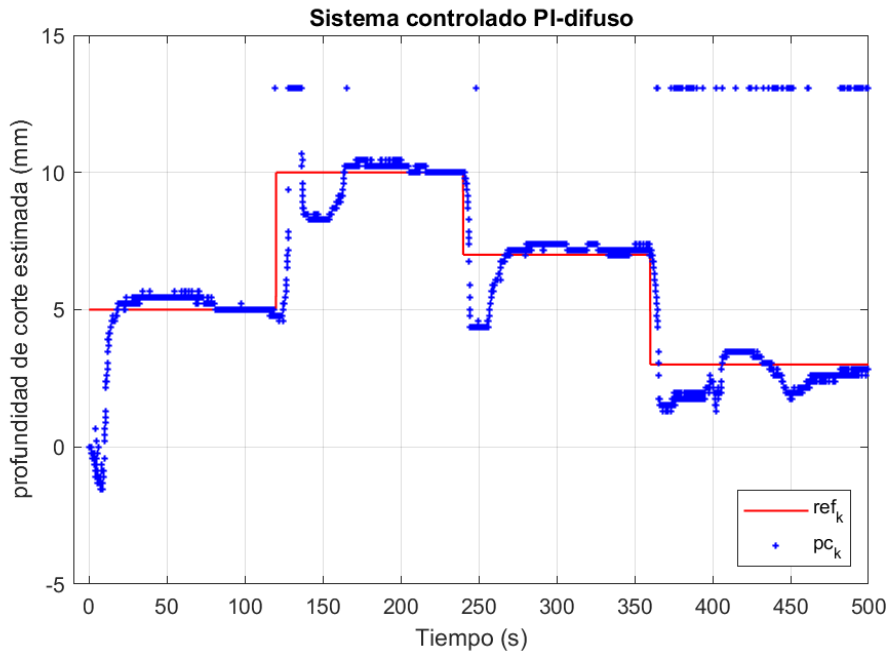
Tabla 3.2: Reglas difusas de $FRMp$.

$FRMi$, se compone de las 15 reglas difusas descritas en la tabla. 3.3.

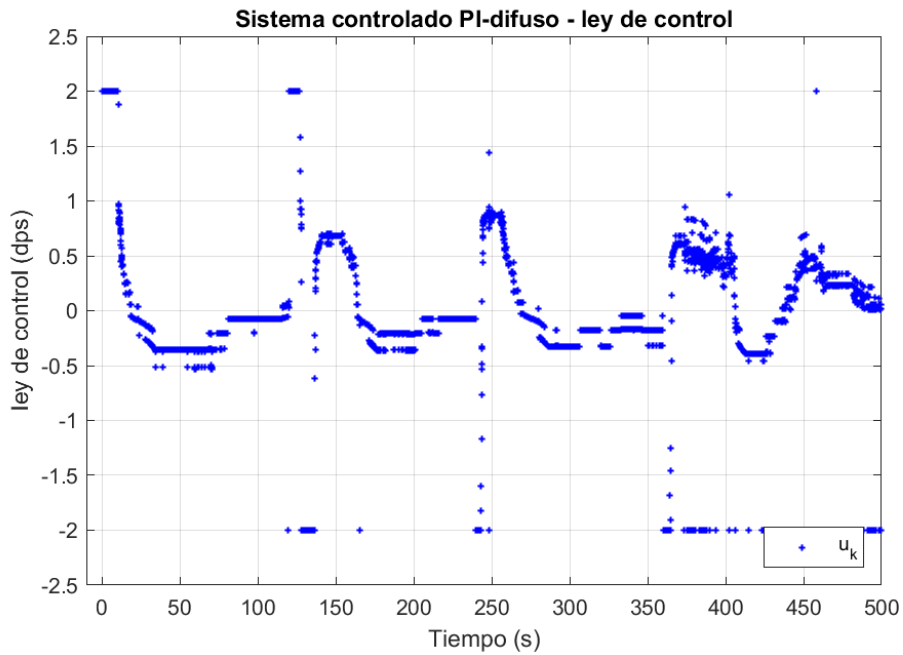
Nombre	Regla
R1	Si e es NB_{eK} y ua es LOW_{uK} , entonces oi es HIG_K
R2	Si e es NB_{eK} y ua es MED_{uK} , entonces oi es MED_K
R3	Si e es NB_{eK} y ua es HIG_{uK} , entonces oi es MED_K
R4	Si e es NS_{eK} y ua es LOW_{uK} , entonces oi es MED_K
R5	Si e es NS_{eK} y ua es MED_{uK} , entonces oi es LOW_K
R6	Si e es NS_{eK} y ua es HIG_{uK} , entonces oi es $VLOW_K$
R7	Si e es ZE_{eK} y ua es LOW_{uK} , entonces oi es LOW_K
R8	Si e es ZE_{eK} y ua es MED_{uK} , entonces oi es LOW_K
R9	Si e es ZE_{eK} y ua es HIG_{uK} , entonces oi es $VLOW_K$
R10	Si e es PS_{eK} y ua es LOW_{uK} , entonces oi es $VLOW_K$
R11	Si e es PS_{eK} y ua es MED_{uK} , entonces oi es MED_K
R12	Si e es PS_{eK} y ua es HIG_{uK} , entonces oi es HIG_K
R13	Si e es PB_{eK} y ua es LOW_{uK} , entonces oi es $VLOW_K$
R14	Si e es PB_{eK} y ua es MED_{uK} , entonces oi es $VLOW_K$
R15	Si e es PB_{eK} y ua es HIG_{uK} , entonces oi es MED_K

Tabla 3.3: Reglas difusas de $FRMi$.

A fin de verificar el funcionamiento del controlador PI-difuso, se implementó el mismo en un programa de computadora. En la figura 3.53 (a) se muestra los resultados de esta implementación, para la secuencia de referencias de profundidad de corte de 5mm, 10mm, 7mm, y 3mm, para una ley de control limitada al intervalo de $[-2, 2]$ dps.



(a) Referencia de control ref_k y profundidad estimada de corte pc_k .



(b) Ley de control u_k .

Figura 3.53: Profundidad de corte controlada con el controlador PI-difuso.

Esta técnica de control, expresada en tiempo discreto, es descrita en el algoritmo de la figura 3.54, y tiene las siguientes entradas y salidas:

- e_k : Error de control.
- se_k : Suma del error de control saturada.
- e : Error de control escalado y saturado.

ua : Señal de control de la iteración anterior escalada y saturada.
 gp : Factor de escalamiento para obtener Kp .
 gi : Factor de escalamiento para obtener Ki .
 get_fKp : Sub-rutina para calcula op , utilizando la $FRMp$.
 get_fKi : Sub-rutina para calcula oi , utilizando la $FRMi$.
 u : Salida del controlador PI-difuso.

Entrada: $e_k, se_k, e, ua, gp, gi, NB_{eK}, NS_{eK}, ZE_{eK}, PS_{eK}, PB_{eK}$
Entrada: $LOW_{uK}, MED_{uK}, HIG_{uK}, VLOW_K, LOW_K, MED_K, HIG_K, VHIG_K$
Salida: u
 1: $Kp := gp \cdot get_fKp(e, ua, NB_{eK}, NS_{eK}, ZE_{eK}, PS_{eK}, \dots)$
 2: $Ki := gi \cdot get_fKi(e, ua, NB_{eK}, NS_{eK}, ZE_{eK}, PS_{eK}, \dots)$
 3: $u := Kp \cdot e_k + Ki \cdot se_k$

Figura 3.54: Algoritmo; controlador PI-difuso.

Es importante resaltar que el algoritmo de la figura 3.54 utiliza las sub-rutinas get_fKp y get_fKi , como base para calcular Kp y Ki . Estas sub-rutinas son descritas en el algoritmo de la figura 3.55 y algoritmo de la figura 3.56 respectivamente, y tienen las siguientes entradas, salidas y sub-rutinas.

e : Error de control escalado y saturado.
 ua : Señal de control de la iteración anterior escalada y saturada.
 NB_{eK} : Vector de 4 elementos que define el conjunto difuso trapezoidal NB_{eK} .
 NS_{eK} : Vector de 3 elementos que define el conjunto difuso triangular NS_{eK} .
 ZE_{eK} : Vector de 3 elementos que define el conjunto difuso triangular ZE_{eK} .
 PS_{eK} : Vector de 3 elementos que define el conjunto difuso triangular PS_{eK} .
 PB_{eK} : Vector de 4 elementos que define el conjunto difuso trapezoidal PB_{eK} .
 LOW_{uK} : Vector de 4 elementos que define el conjunto difuso trapezoidal LOW_{uK} .
 MED_{uK} : Vector de 3 elementos que define el conjunto difuso triangular MED_{uK} .
 HIG_{uK} : Vector de 4 elementos que define el conjunto difuso trapezoidal HIG_{uK} .
 $VLOW_K$: Vector de 3 elementos que define el conjunto difuso triangular $VLOW_K$.
 LOW_K : Vector de 3 elementos que define el conjunto difuso triangular LOW_K .
 MED_K : Vector de 3 elementos que define el conjunto difuso triangular MED_K .
 HIG_K : Vector de 3 elementos que define el conjunto difuso triangular HIG_K .
 $VHIG_K$: Vector de 3 elementos que define el conjunto difuso triangular $VHIG_K$.
 op : Salida de $FRMp$.
 oi : Salida de $FRMi$.

$rule_K$: Implicación difusa de Mamdani para conjuntos difusos triangulares y trapezoidales.

Entrada: $e, ua, NB_{eK}, NS_{eK}, ZE_{eK}, PS_{eK}, PB_{eK}$

Entrada: $LOW_{uK}, MED_{uK}, HIG_{uK}, VLOW_K, LOW_K, MED_K, HIG_K, VHIG_K$

Salida: oi

- 1: $V := \text{vector}$
- 2: $nv := \text{obtener la cantidad de elementos de } V$
- 3: $dt := 1/(nv)$
- 4: $t := 0$
- 5: $PA := 0$
- 6: $A := 0$
- 7: $V_1 \leftarrow rule_K(NB_{eK}, 2, LOW_{uK}, 2, VHIG_K, e, ua)$
- 8: $V_2 \leftarrow rule_K(NB_{eK}, 2, MED_{uK}, 1, HIG_K, e, ua)$
- 9: $V_3 \leftarrow rule_K(NB_{eK}, 2, HIG_{uK}, 2, MED_K, e, ua)$
- 10: $V_4 \leftarrow rule_K(NS_{eK}, 1, LOW_{uK}, 2, HIG_K, e, ua)$
- 11: $V_5 \leftarrow rule_K(NS_{eK}, 1, MED_{uK}, 1, MED_K, e, ua)$
- 12: $V_6 \leftarrow rule_K(NS_{eK}, 1, HIG_{uK}, 2, LOW_K, e, ua)$
- 13: $V_7 \leftarrow rule_K(ZE_{eK}, 1, LOW_{uK}, 2, MED_K, e, ua)$
- 14: $V_8 \leftarrow rule_K(ZE_{eK}, 1, MED_{uK}, 1, LOW_K, e, ua)$
- 15: $V_9 \leftarrow rule_K(ZE_{eK}, 1, HIG_{uK}, 2, VLOW_K, e, ua)$
- 16: $V_{10} \leftarrow rule_K(PS_{eK}, 1, LOW_{uK}, 2, LOW_K, e, ua)$
- 17: $V_{11} \leftarrow rule_K(PS_{eK}, 1, MED_{uK}, 1, LOW_K, e, ua)$
- 18: $V_{12} \leftarrow rule_K(PS_{eK}, 1, HIG_{uK}, 2, VLOW_K, e, ua)$
- 19: $V_{13} \leftarrow rule_K(PB_{eK}, 2, LOW_{uK}, 2, MED_K, e, ua)$
- 20: $V_{14} \leftarrow rule_K(PB_{eK}, 2, MED_{uK}, 1, HIG_K, e, ua)$
- 21: $V_{15} \leftarrow rule_K(PB_{eK}, 2, HIG_{uK}, 2, HIG_K, e, ua)$
- 22: **para** $i \leftarrow 1, nv$ **hacer**
- 23: $V(i) := \text{obtener el máximo de entre } V_1(i), V_2(i), \dots, V_{15}(i)$
- 24: $PA := PA + t \cdot dt \cdot V(i)$
- 25: $A := A + dt \cdot V(i)$
- 26: $t := t + dt$
- 27: **fin para**
- 28: **si** $A = 0$ **entonces**
- 29: $op := 0$
- 30: **si no**
- 31: $op := PA/A$
- 32: **fin si**

Figura 3.55: Algoritmo; get_fKp .

Entrada: $e, ua, NB_{eK}, NS_{eK}, ZE_{eK}, PS_{eK}, PB_{eK}$
Entrada: $LOW_{uK}, MED_{uK}, HIG_{uK}, VLOW_K, LOW_K, MED_K, HIG_K, VHIG_K$
Salida: oi

- 1: $V := \text{vector}$
- 2: $nv := \text{obtener la cantidad de elementos de } V$
- 3: $dt := 1/(nv)$
- 4: $t := 0$
- 5: $PA := 0$
- 6: $A := 0$
- 7: $V_1 \leftarrow \text{rule}_K(NB_{eK}, 2, LOW_{uK}, 2, HIG_K, e, ua)$
- 8: $V_2 \leftarrow \text{rule}_K(NB_{eK}, 2, MED_{uK}, 1, MED_K, e, ua)$
- 9: $V_3 \leftarrow \text{rule}_K(NB_{eK}, 2, HIG_{uK}, 2, MED_K, e, ua)$
- 10: $V_4 \leftarrow \text{rule}_K(NS_{eK}, 1, LOW_{uK}, 2, MED_K, e, ua)$
- 11: $V_5 \leftarrow \text{rule}_K(NS_{eK}, 1, MED_{uK}, 1, LOW_K, e, ua)$
- 12: $V_6 \leftarrow \text{rule}_K(NS_{eK}, 1, HIG_{uK}, 2, VLOW_K, e, ua)$
- 13: $V_7 \leftarrow \text{rule}_K(ZE_{eK}, 1, LOW_{uK}, 2, LOW_K, e, ua)$
- 14: $V_8 \leftarrow \text{rule}_K(ZE_{eK}, 1, MED_{uK}, 1, LOW_K, e, ua)$
- 15: $V_9 \leftarrow \text{rule}_K(ZE_{eK}, 1, HIG_{uK}, 2, VLOW_K, e, ua)$
- 16: $V_{10} \leftarrow \text{rule}_K(PS_{eK}, 1, LOW_{uK}, 2, VLOW_K, e, ua)$
- 17: $V_{11} \leftarrow \text{rule}_K(PS_{eK}, 1, MED_{uK}, 1, MED_K, e, ua)$
- 18: $V_{12} \leftarrow \text{rule}_K(PS_{eK}, 1, HIG_{uK}, 2, HIG_K, e, ua)$
- 19: $V_{13} \leftarrow \text{rule}_K(PB_{eK}, 2, LOW_{uK}, 2, VLOW_K, e, ua)$
- 20: $V_{14} \leftarrow \text{rule}_K(PB_{eK}, 2, MED_{uK}, 1, VLOW_K, e, ua)$
- 21: $V_{15} \leftarrow \text{rule}_K(PB_{eK}, 2, HIG_{uK}, 2, MED_K, e, ua)$
- 22: **para** $i \leftarrow 1, nv$ **hacer**
- 23: $V(i) := \text{obtener el máximo de entre } V_1(i), V_2(i), \dots, V_{15}(i)$
- 24: $PA := PA + t \cdot dt \cdot V(i)$
- 25: $A := A + dt \cdot V(i)$
- 26: $t := t + dt$
- 27: **fin para**
- 28: **si** $A = 0$ **entonces**
- 29: $oi := 0$
- 30: **si no**
- 31: $oi := PA/A$
- 32: **fin si**

Figura 3.56: Algoritmo; get_fKi .

Tal como se puede ver en los algoritmos 3.55 y 3.56, éstos utilizan la sub-rutina $rule_K$, la misma que es descrita en el anexo A.2.

Se extrajeron los primeros 120 segundos de los resultados mostrados en la figura 3.53 (a), que corresponden a la respuesta del sistema controlado ante una entrada de tipo escalón de 5mm de amplitud. De esta respuesta, representada en la figura 3.57, se obtuvo un t_s de 78.49s y una M_p de 12.23 %, valores que cumplen con las especificaciones planteadas en la sección 3.1.3. Por tal motivo, en este trabajo se considera que se ha corroborado que el

controlador PI-difuso funciona satisfactoriamente.

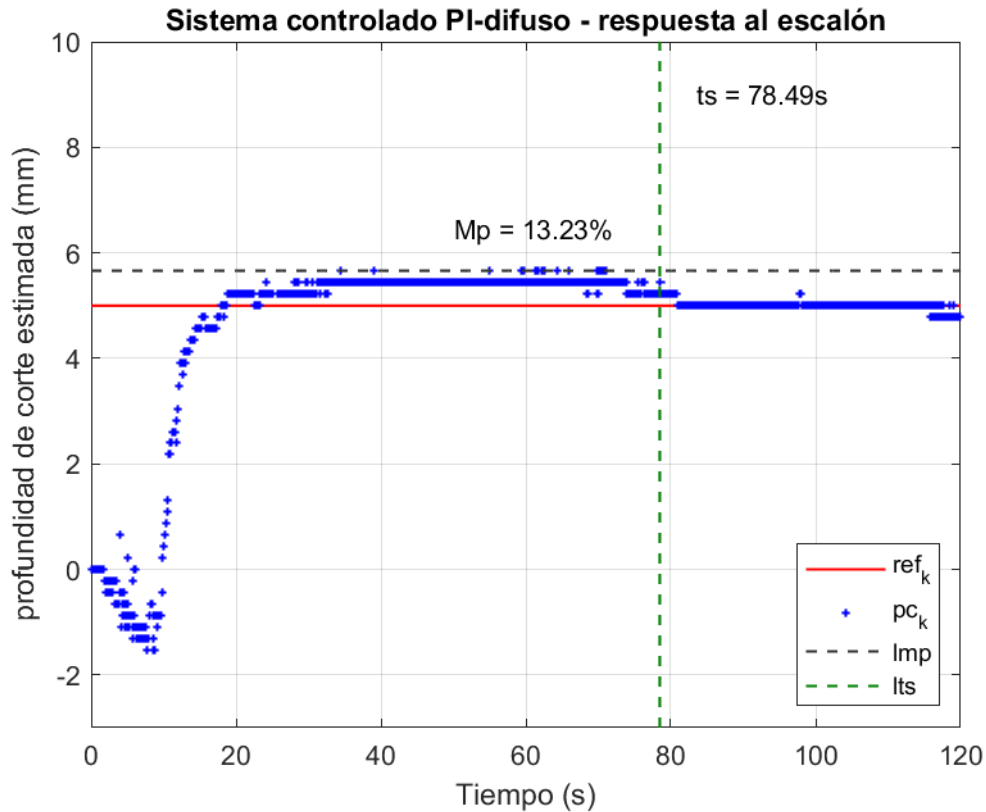


Figura 3.57: Respuesta del sistema controlado PI-difuso ante una entrada de tipo escalón de 5mm de amplitud.

Dónde:

ref_k : Referencia de control de profundidad de corte.

pc_k : Profundidad de corte estimada.

lts : Línea vertical que señala t_s .

lmp : Línea horizontal que señala M_p .

3.11.4. Elección de la técnica de control de profundidad de corte

En las secciones 3.11.1, 3.11.2, y 3.11.3, se evaluó el uso de 3 posibles técnicas de control, las cuales fueron: un controlador PID, un controlador difuso y un controlador PI-difuso. En dichas secciones se mostró experimentalmente que se pudo corroborar el funcionamiento satisfactorio únicamente de los controladores PID y PI-difuso.

Del mismo modo, si bien se optó por regular manualmente las ganancias de control del controlador PID, existen procedimientos que permiten calcular valores apropiados para aquellas ganancias. Sin embargo, aquellos procedimientos usualmente requieren conocer

algún modelo matemático que represente la dinámica del sistema a controlar o hacer múltiples pruebas experimentales hasta sintonizar los valores las ganancias, a diferencia del controlador PI-difuso, que en principio únicamente requirió escalar y limitar (saturar) los valores de algunas de las variables implicadas, mostrando así este controlador mayor versatilidad de implementación que el controlador PID.

En base a lo mencionado en esta sección, se concluye que tanto un controlador PID como un controlador PI-difuso podrían funcionar satisfactoriamente como parte del algoritmo de corte con penetración controlada. Sin embargo, la mayor versatilidad de implementación implicó que, finalmente, se elija utilizar el controlador PI-difuso para cumplir dicha función.

3.12. Algoritmo de corte con penetración controlada

En esta sección se presenta el algoritmo de corte con penetración controlada con bistrú empleando un manipulador robótico, que es objetivo principal del presente trabajo de tesis. Con fines prácticos, este algoritmo será denominado en adelante como algoritmo de corte con penetración controlada. El algoritmo es descrito en el algoritmo de la figura 3.58 y utiliza como sub-rutinas a los algoritmos desarrollados en secciones anteriores del presente trabajo de tesis. Siendo así, a continuación se indican las constantes, variables y sub-rutinas involucradas en el algoritmo de corte con penetración controlada.

NB_{eK} : Vector de 4 elementos que define el conjunto difuso trapezoidal NB_{eK} .

NS_{eK} : Vector de 3 elementos que define el conjunto difuso triangular NS_{eK} .

ZE_{eK} : Vector de 3 elementos que define el conjunto difuso triangular ZE_{eK} .

PS_{eK} : Vector de 3 elementos que define el conjunto difuso triangular PS_{eK} .

PB_{eK} : Vector de 4 elementos que define el conjunto difuso trapezoidal PB_{eK} .

LOW_{uK} : Vector de 4 elementos que define el conjunto difuso trapezoidal LOW_{uK} .

MED_{uK} : Vector de 3 elementos que define el conjunto difuso triangular MED_{uK} .

HIG_{uK} : Vector de 4 elementos que define el conjunto difuso trapezoidal HIG_{uK} .

$VLOW_K$: Vector de 3 elementos que define el conjunto difuso triangular $VLOW_K$.

LOW_K : Vector de 3 elementos que define el conjunto difuso triangular LOW_K .

MED_K : Vector de 3 elementos que define el conjunto difuso triangular MED_K .

HIG_K : Vector de 3 elementos que define el conjunto difuso triangular HIG_K .

$VHIG_K$: Vector de 3 elementos que define el conjunto difuso triangular $VHIG_K$.

IMG : Imagen tomada.

$palette$: Sub-imagen de la región interior de la paleta de colores.

T_s : Tiempo de muestreo.

R_p, G_p, B_p : Estructuras de parámetros comunes de los píxeles de color rojos, verdes y azules, respectivamente.

IMG_{SCLA} : Imagen contenedora de la pseudo-región del bisturí.

LR : Vector de posición del centro de la última pseudo-marca de color roja conocida.

LG : Vector de posición del centro de la última pseudo-marca de color verde conocida.

LB : Vector de posición del centro de la última pseudo-marca de color azul conocida.

NR : Vector de posición del centro de la nueva pseudo-marca de color roja.

NG : Vector de posición del centro de la nueva pseudo-marca de color verde.

NB : Vector de posición del centro de la nueva pseudo-marca de color azul.

P : Punta del bisturí en IMG_{SCLA} .

P' : Punto que señala el límite de hasta donde la hoja del bisturí puede ingresar verticalmente a la muestra en IMG_{SCLA} .

xm : Posición en el eje vertical de P' .

ym : Posición en el eje horizontal de P' .

d_1 : Posición en el eje vertical de P , proveniente de la ecuación (3.14).

d_2 : Posición en el eje vertical de P' , proveniente de la ecuación (3.14).

ref_k : Referencia de control.

pc_k : Profundidad de corte estimada.

e_k : Error de control.

se_k : Suma del error de control.

u_k : Señal de control.

e : Error de control escalado y saturado.

u : Salida del controlador PI-difuso.

ua : Señal de control de la iteración anterior escalada y saturada.

fie : Factor de escalamiento para obtener e .

fiu : Factor de escalamiento para obtener ua .

Kp : Ganancia proporcional del controlador PI-difuso.

Ki : Ganancia integral del controlador PI-difuso.

gp : Factor de escalamiento para obtener Kp .

gi : Factor de escalamiento para obtener Ki .

has : Longitud referencial para la altura de la imagen reducida.

bas : Longitud de la base de la imagen reducida.

dis : Distancia entre el centro de la marca de color verde y el punto medio entre la punta del bisturí y el límite hasta donde la hoja del bisturí puede ingresar verticalmente a la muestra.

bl : Nivel de negro.

f_1 : Factor de tolerancia de umbrales máximos.

f_2 : Factor de tolerancia de áreas.

l_2 : Distancia entre la punta del bisturí y el punto que señala el límite de hasta donde la hoja del bisturí puede ingresar verticalmente a la muestra.

f_3 : Factor de relación entre l_2 y has .

f_{31} : Factor de relación entre bas y has .

Obtener la paleta de colores : Sub-rutina descrita en el algoritmo de la figura 3.12.

Analizar la paleta de colores : Sub-rutina descrita en el algoritmo de la figura 3.16.

Obtener la pseudo-región del bisturí : Sub-rutina descrita en el algoritmo de la figura 3.37.

Obtener el punto más bajo de la pseudo-región del bisturí : Sub-rutina descrita en el algoritmo de la figura 3.39.

Controlador PI-difuso : Sub-rutina descrita en el algoritmo de la figura 3.54.

Entrada: ref_k

- 1: Inicializar; $Ts, f_1, f_2, f_3, f_{31}, bl, gp, gi, fie, fiu, se_k, ua$
- 2: Definir conjuntos difusos; $NB_{eK}, NS_{eK}, ZE_{eK}...$
- 3: $IMG :=$ tomar una fotografía
- 4: $palette, bas, has, dis, p1 \leftarrow$ obtener la paleta de colores (IMG, f_3, f_{31})
- 5: $Rp, Gp, Bp, LRp, LGp, LBp \leftarrow$ analizar la paleta de colores ($palette, bl, f_1$)
- 6: $LR := LRp + p1$
- 7: $LG := LGp + p1$
- 8: $LB := LBp + p1$
- 9: $IMG_{SCLA}, NR, NG, NB \leftarrow$ obtener la pseudo-región del bisturí ($IMG, Rp, Gp, Bp, ...$)
- 10: $LR := NR$
- 11: $LG := NG$
- 12: $LB := NB$
- 13: $xm, ym :=$ obtener el punto más bajo de la pseudo-región del bisturí (IMG_{SCLA})
- 14: $d_1 := ym$
- 15: **mientras** 1 = 1 **hacer**
- 16: $IMG :=$ tomar una fotografía
- 17: $IMG_{SCLA}, NR, NG, NB \leftarrow$ obtener la pseudo-región del bisturí ($IMG, Rp, Gp, Bp, ...$)
- 18: $LR := NR$
- 19: $LG := NG$
- 20: $LB := NB$
- 21: $xm, ym \leftarrow$ obtener el punto más bajo de la pseudo-región del bisturí (IMG_{SCLA})
- 22: $d_2 := ym$
- 23: $pc_k := d_1 - d_2$
- 24: $e_k := ref_k - pc_k$
- 25: $se_k := se_k + e_k \cdot Ts$
- 26: $se_k :=$ saturar se_k a un intervalo definido por el usuario
- 27: $e := e_k \cdot fie$
- 28: $e :=$ saturar e al intervalo de -1 a 1
- 29: $ua :=$ saturar ua al intervalo de -1 a 1
- 30: $u \leftarrow$ controlador PI-difuso ($e_k, se_k, e, ua, gp, gi, ...$)
- 31: $ua := u \cdot fiu$
- 32: $u_k :=$ saturar u a un intervalo definido por el usuario
- 33: Enviar u_k a la interface electrónica de operación del robot
- 34: **fin mientras**

Figura 3.58: Algoritmo; algoritmo de corte con penetración controlada.

Cabe resaltar que los operadores de saturación indicados en las líneas 26 y 32 del algoritmo de la figura 3.58 corresponden a los saturadores ST_e y ST_{ua} , respectivamente, abordados en la sección 3.11.3. En dicha sección se indica que en este trabajo no se establece algún criterio para calcular los límites de saturación de dichos saturadores; sin embargo, en la misma sección se sugieren consideraciones para determinar aquellos límites.

CAPÍTULO IV

ANÁLISIS DE LOS RESULTADOS DE LA INVESTIGACIÓN Y CONTRASTACIÓN DE LAS HIPÓTESIS

En esta sección se exponen y analizan los resultados obtenidos de las pruebas del algoritmo de corte con penetración controlada (algoritmo de la figura 3.58), así como también se verifica el cumplimiento o incumplimiento de las hipótesis planteadas en la sección 1.6. Para ello, se comenzará describiendo las condiciones y consideraciones bajo las que se realizaron las pruebas. Acto seguido, se mostrarán los resultados obtenidos de las mismas y, finalmente, se evaluará el cumplimiento o incumplimiento de las 7 hipótesis compuestas por una hipótesis principal y 6 hipótesis secundarias.

4.1. Configuración experimental

Cada muestra de tejido empleada para los experimentos de corte fue colocada en una plataforma y sujeta a ella mediante hilos, con la finalidad de reducir su movimiento o deformación durante los experimentos, y procurar mantener plana la superficie de la muestra. Además, se colocó debajo de la muestra una lámina de madera, envuelta en plástico, para evitar que la muestra se hunda, tal como se puede ver en la figura 4.1.

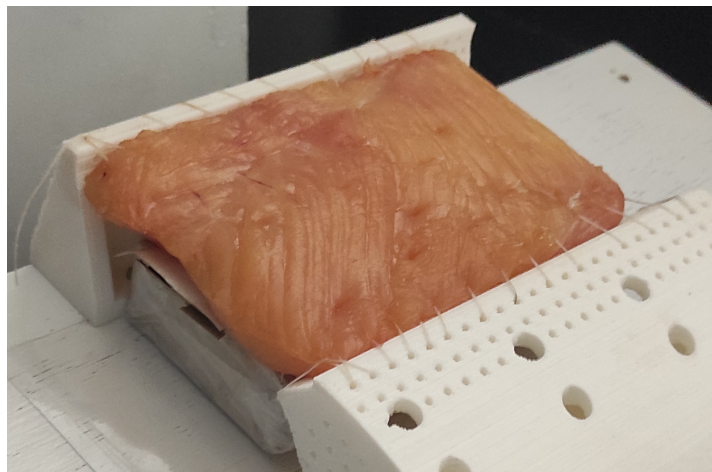


Figura 4.1: Muestra de pecho de pollo sujeta a la plataforma de muestras.

Seguidamente, se colocó una capa de papel sanitario sobre la muestra, a fin de evitar reflejos en su superficie, tal como se muestra en la figura 4.2. Se hizo esto, debido a que la

presencia de reflejos en la superficie de la muestra podría ocasionar exceso de irregularidades y con ello inducir a fallos, tal como se señala en la sección 3.9.

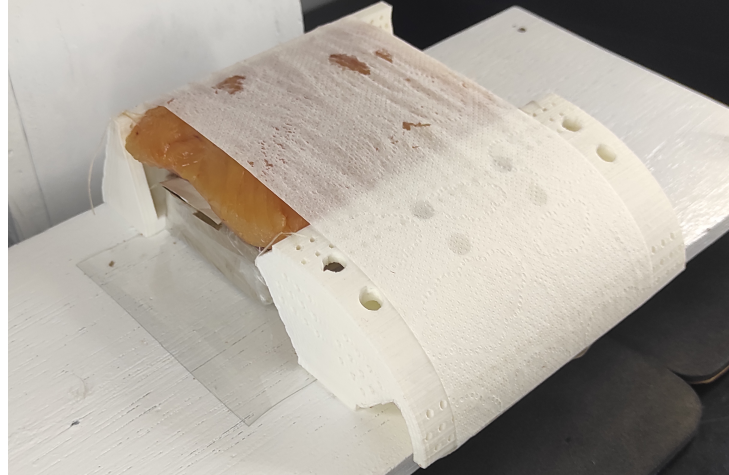


Figura 4.2: Papel sanitario colocado sobre una muestra de pecho de pollo.

Idealmente, se planteaba colocar la cámara de video con línea de vista perpendicular al plano de la hoja del bisturí. Sin embargo, durante los experimentos, se encontró que esto posibilita la indeseable visualización de reflejos sobre la hoja del bisturí. Para evitarlo, fue necesario inclinar ligeramente la cámara de video con el lado de la lente elevado. Por su parte, con la finalidad de evitar que la superficie de la muestra quede oculta a la vista de la cámara de video, se optó por inclinar ligeramente la plataforma de muestras, elevando su parte posterior, tal como se muestra en la figura 4.3.

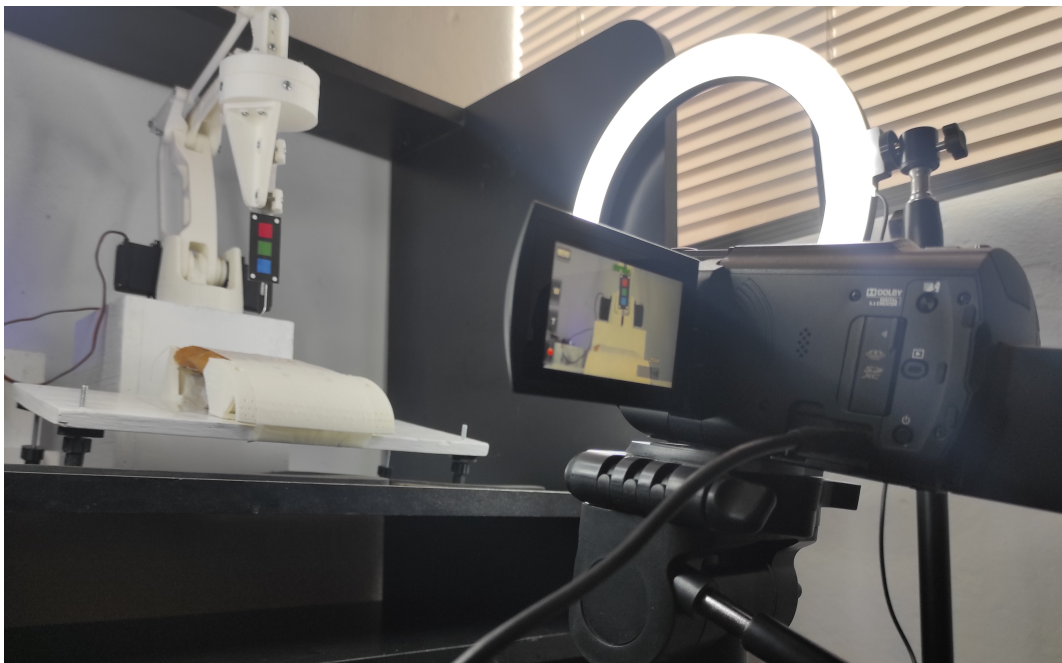


Figura 4.3: Plataforma de muestras y cámara de video ligeramente inclinadas.

Así mismo, se ubicaron dos aros de luz led por encima y a los lados de la plataforma de muestras. Esta ubicación permitió visualizar claramente la paleta de colores y el perfil del bisturí, sin mostrar reflejos de los aros de luz sobre la superficie de la hoja del bisturí. Esta disposición de materiales se muestra en la figura 4.4.



Figura 4.4: Disposición de materiales.

4.2. Medición de la profundidad de corte

El principio utilizado para medir la profundidad de corte consistió en seccionar cada muestra perpendicularmente al plano que definió la hoja del bisturí durante la incisión y luego medir la profundidad de la abertura dejada por la hoja del bisturí. En dicha sección de muestra, la abertura tiene el aspecto visual de una línea que parte de la superficie de la muestra y termina dentro de ella. En adelante, aquellas líneas serán denominadas líneas de abertura. Este procedimiento de medición se basó el principio empleado en [10]. En la figura 4.5, se puede ver una muestra de pecho de pollo seccionada perpendicularmente al plano de la hoja del bisturí.



Figura 4.5: Sección de muestra de pecho de pollo con dos líneas de abertura.

Tal como se puede ver en la figura 4.5, resulta difícil distinguir alguna línea de la abertura y, en consecuencia, se dificulta medir la profundidad de corte. Sin embargo, se encontró que aquellas líneas pueden ser resaltadas visualmente, colocando salsa de soya sobre la cara seccionada, tal como se muestra en la figura 4.6.

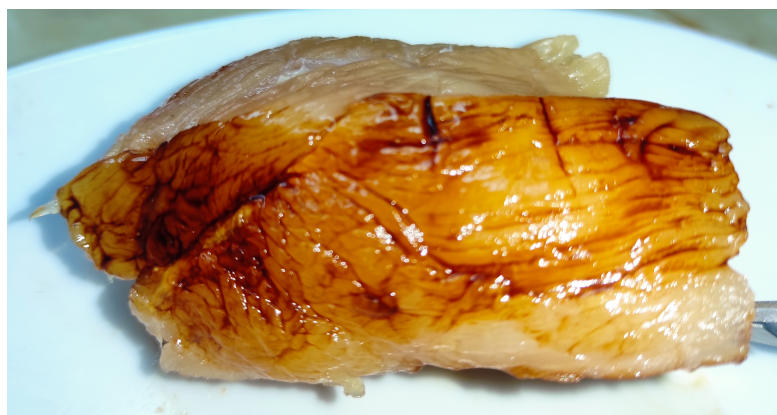


Figura 4.6: Sección de muestra de pecho de pollo con dos líneas de abertura resaltadas.

Se midieron las aberturas empelando imágenes capturadas con un microscopio digital. En ellas se puede ver una regla de calibración de microscopio, la misma que fue colocada

sobre las caras seccionadas, a fin ser utilizada como referencia de longitud, tal como se puede ver en la figura 4.7.

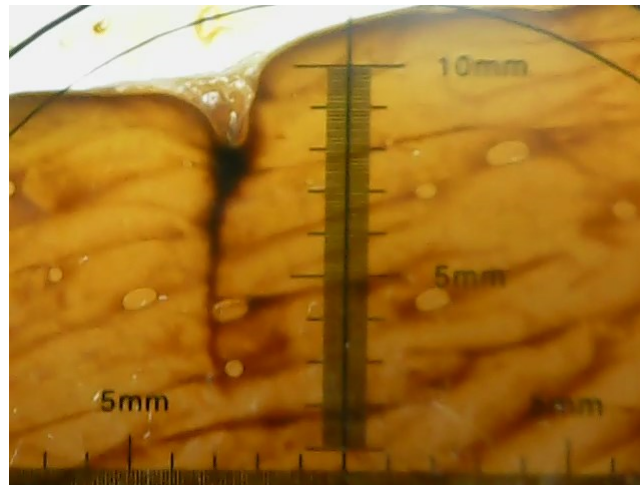


Figura 4.7: Cara seccionada vista con un microscopio digital.

En cada imagen de una cara seccionada, se midió la profundidad de corte como la longitud del segmento de recta de la abertura, que es el segmento de recta que parte desde el segmento de recta que pasa por la superficie de la muestra, se dirige aproximadamente a través de la línea de abertura y llega hasta el punto más bajo de aquella línea. Posteriormente, aquella distancia es comparada con una longitud conocida extraída de la regla de calibración. En la figura 4.8, se muestra un caso de medición de la profundidad de corte. En ella se representa al segmento de recta que pasa por la superficie de color rojo, al segmento de recta de la abertura de color verde y al segmento de recta de longitud conocida de color azul. Cabe resaltar que en el anexo C se muestra este proceso de medición aplicado a las 30 pruebas de corte numeradas en las tablas 4.1, 4.2, y 4.3.

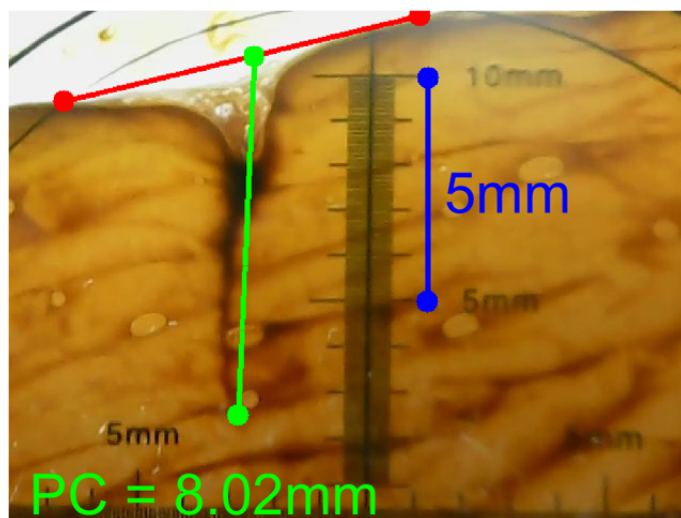


Figura 4.8: Medición de la profundidad de corte.

4.3. Resultados de la investigación

El algoritmo de corte con penetración controlada fue implementado en el programa de computadora que es detallado en [32], el que a su vez fue redactado en el lenguaje de programación C++ y ejecutado en el entorno de desarrollo Microsoft Visual Studio 2019 [25]. Las pruebas de corte fueron realizadas utilizando muestras de pecho de pollo, empleando una hoja de bisturí número 10. La clase de muestra fue elegida debido a su uso, como tejido blando, en [10], además de su fácil acceso en el mercado local. Cabe resaltar que existen otras clases de muestras utilizables, tales como hígado de cerdo (utilizada en utilizada en [7], [6], [38]), o riñón de cerdo (utilizada en [31] [38]), entre otras.

Se encontró que para cortes controlados con referencias de profundidad de corte inferiores a 5mm, no siempre se lograba penetrar la muestra, dando lugar a que en ocasiones el bisturí únicamente se hunda en la muestra sin abrir una grieta o abriendo una muy pequeña, pero dando la errada la percepción visual de que ha penetrado exitosamente, para que luego de retirar el bisturí la superficie de la muestra, esta regrese visiblemente a su forma original. Así mismo, las dimensiones de la hoja del bisturí y los espesores de las muestras dificultaron realizar cortes con referencias de profundidad superiores a 10mm. Es por estas razones que se eligió realizar cortes con penetración controlada en el rango de 5mm a 10mm.

Se ejecutaron 10 pruebas de corte por cada una de las referencias de 5mm, 7.5mm y 10mm, haciendo un total de 30 pruebas de cortes con penetración controlada.

En las tablas 4.1, 4.2, y 4.3, se muestran los resultados obtenidos de las pruebas de corte realizadas con referencias de 5mm, 7.5mm, y 10mm, respectivamente. En estas tablas y en la figura 4.9, se emplean las siguientes variables.

ref : Referencia de control de profundidad de corte (mm).

pcm : Profundidad de corte medida (mm).

er : Error de control de profundidad de corte (mm). Se calcula con la ecuación (4.1).

err : Error relativo porcentual de control de profundidad de corte (%). Se calcula con la ecuación (4.2).

$$er = ref - pcm \quad (4.1)$$

$$err = 100 \cdot \frac{er}{ref} \quad (4.2)$$

Prueba #	<i>ref(mm)</i>	<i>pcm(mm)</i>	<i>er(mm)</i>	<i>err(%)</i>
1	5	5,01	-0,01	-0,20
2	5	4,66	0,34	6,80
3	5	4,90	0,10	2,00
4	5	3,77	1,23	24,60
5	5	5,02	-0,02	-0,40
6	5	6,14	-1,14	-22,80
7	5	4,95	0,05	1,00
8	5	3,46	1,54	30,80
9	5	4,94	0,06	1,20
10	5	3,80	1,20	24,00

Tabla 4.1: Profundidades de corte obtenidas con referencias de 5mm.

Prueba #	<i>ref(mm)</i>	<i>pcm(mm)</i>	<i>er(mm)</i>	<i>err(%)</i>
11	7,5	8,75	-1,25	-16,66
12	7,5	5,59	1,91	25,46
13	7,5	7,78	-0,28	-3,73
14	7,5	8,63	-1,13	-15,06
15	7,5	8,78	-1,28	-17,06
16	7,5	8,41	-0,91	-12,13
17	7,5	8,02	-0,52	-6,93
18	7,5	8,03	-0,53	-7,06
19	7,5	8,19	-0,69	-9,2
20	7,5	9,38	-1,88	-25,06

Tabla 4.2: Profundidades de corte obtenidas con referencias de 7.5mm.

Prueba #	<i>ref(mm)</i>	<i>pcm(mm)</i>	<i>er(mm)</i>	<i>err(%)</i>
21	10	13,30	-3,30	-33,00
22	10	9,81	0,19	1,90
23	10	10,74	-0,74	-7,40
24	10	8,31	1,69	16,90
25	10	9,59	0,41	4,10
26	10	8,67	1,33	13,30
27	10	9,28	0,72	7,20
28	10	11,97	-1,97	-19,70
29	10	10,78	-0,78	-7,80
30	10	9,21	0,79	7,90

Tabla 4.3: Profundidades de corte obtenidas con referencias de 10mm.

Los valores de las profundidades de corte medias en las 30 pruebas fueron correlacionados con sus respectivas referencias de control de profundidad de corte, obteniendo un coeficiente de correlación de Pearson r de 0.89. Este valor de r al ser cercano a 1, indica una fuerte correlación entre ambas variables, tal como se señala en la sección 2.4.

En la figura 4.9, se representan gráficamente las profundidades de corte medidas, con



Figura 4.10: Efecto final acoplado al sensor FT300, sosteniendo un bisturí. Esta fotografía fue tomada en el Laboratorio de Mecatrónica de la Universidad de Ingeniería y Tecnología UTEC.

En consecuencia, se verifica que fue posible diseñar un efecto final para un manipulador robótico que le provea la capacidad de sostener un bisturí.

■ **Contrastación de la segunda hipótesis secundaria**

Durante el desarrollo del presente trabajo de tesis, la disponibilidad de materiales permitió realizar el trabajo, directamente en base a información experimental obtenida de la realización de cortes quirúrgicos con bisturí en muestras de pecho de pollo, a diferentes profundidades. Por tal motivo, no fue necesario crear una base de datos de imágenes que simulen cortes quirúrgicos.

En consecuencia, no se verificó que fuera posible crear una base de datos de imágenes para el desarrollo de algoritmos, en las que se simulen cortes quirúrgicos con bisturí a diferentes profundidades.

■ **Contrastación de la tercera hipótesis secundaria**

En la sección 3.3, se expone el análisis de color en los espacios de color RGB y HSV como parte fundamental del criterio general de detección de píxeles de color, repre-

sentado por el algoritmo de la figura 3.17.

En consecuencia, se verifica que el análisis de imágenes en los espacios de color RGB y HSV permitió desarrollar un algoritmo de detección de píxeles de color.

■ **Contrastación de la cuarta hipótesis secundaria**

En la sección 3.3.8, se plantea un criterio general de detección de píxeles de color, representado por el algoritmo de la figura 3.17. El mismo forma parte del algoritmo de corte con penetración controlada.

En consecuencia, se verifica que el empleo del algoritmo de detección de píxeles de color permitió desarrollar un algoritmo de estimación de la profundidad de corte con bisturí.

■ **Contrastación de la quinta hipótesis secundaria**

En la sección 3.11, se muestra cómo fueron evaluadas tres técnicas de control, que fueron: un controlador PID, un controlador difuso, y un controlador PI-difuso, de las que se eligió una para formar parte del algoritmo de corte con penetración controlada.

En consecuencia, se verifica que la revisión bibliográfica permitió seleccionar una técnica de control de profundidad de corte, priorizando el empleo de controladores PI, PID, difuso.

■ **Contrastación de la sexta hipótesis secundaria**

En la sección 3.12, se puede ver que el algoritmo de corte con penetración controlada utiliza como sub-rutinas al algoritmo de la figura 3.39 para la estimación de la profundidad de corte y al algoritmo de la figura 3.54 para el control de la profundidad de corte.

En consecuencia, se verifica que el uso combinado del algoritmo de estimación de la profundidad de corte, con la técnica de control de profundidad de corte, permitió desarrollar un algoritmo de corte con penetración controlada.

CONCLUSIONES

1. Fue posible diseñar un algoritmo de corte con penetración controlada con bisturí, empleando un manipulador robótico, para la realización de cortes en muestras de tejido blando de superficie plana. Se llegó a esta conclusión a partir de realizar 30 pruebas de corte a muestras de pecho de pollo. Los resultados fueron expuestos en las tablas 4.1, 4.2 y 4.3. En ellas se muestra que fue posible la realización de cortes con penetración controlada con errores relativos porcentuales de entre -33.00 % y 30.80 %, cifras que son similares a los márgenes de error de entre 7.33 % y 41.34 % obtenidos en [31]. Así mismo, en base a la información de aquellas tablas, se obtuvo un coeficiente de correlación de Pearson de 0.89 (véase la sección 2.4), que al ser cercano a 1 muestra que existe una fuerte correlación entre las profundidades de corte medidas y sus respectivas referencias de control de profundidad de corte. Por estas razones, en el presente trabajo de tesis se considera que los resultados obtenidos de las pruebas de corte, que implementa el algoritmo de corte con penetración controlada, fueron satisfactorios y, en consecuencia, se cumplió con el objetivo general de la tesis.
2. Ha sido posible diseñar e implementar un efector final que le otorgó a un manipulador robótico la capacidad de sostener un bisturí. Para llegar a esta conclusión, se fabricó el efector final desarrollado en la sección 3.2 y se lo acopló a un manipulador robótico, tal como se muestra en la figura 4.10. En dicha figura, el manipulador robótico utilizado fue el robot UR5 [36].
3. No se verificó que fuera posible crear una base de datos de imágenes para el desarrollo de algoritmos, en la que las imágenes simulen cortes quirúrgicos con bisturí a diferentes profundidades. Esto se debió a que no fue necesaria la creación de dicha base de datos, dado que se contó con las facilidades para realizar directamente los cortes quirúrgicos a diferentes profundidades, tal como se aborda a lo largo del presente trabajo de tesis. Sin embargo, se presume que es posible elaborar una base de datos que, en lugar de simular, muestre la realización de cortes con penetración controlada. Dicha base de datos sería elaborada recolectando múltiples imágenes tomadas durante las pruebas del algoritmo de corte con penetración controlada. Asimismo, se considera que aquella base de datos podría resultar útil para la realización de futuros trabajos que aborden la realización de cortes quirúrgicos controlados.

4. Fue posible desarrollar e implementar, en un programa de computadora, un algoritmo de detección de píxeles de color (algoritmo de la figura 3.17), en base al análisis de imágenes en los espacios de color RGB y HSV. Se llegó a esta conclusión debido a que el programa de computadora mostrado en [32] que fue utilizado en el presente trabajo de tesis para la realización de cortes con penetración controlada, contiene como sub-rutina al algoritmo de la figura 3.17, el mismo que es utilizado para detectar píxeles de color en base a información en el espacio de color HSV. Del mismo modo, en la sección 3.3 se expone el análisis de color en los espacios de color RGB y HSV como parte fundamental del desarrollo del algoritmo de corte con penetración controlada.
5. Fue posible desarrollar e implementar, en un programa de computadora, un algoritmo de estimación de la profundidad de corte con bisturí en base al algoritmo de detección de píxeles de color. Se llegó a esta conclusión, debido a que en el programa de computadora mostrado en [32], que fue utilizado en el presente trabajo de tesis para la realización de cortes con penetración controlada, se puede ver que se logra la estimación de la profundidad de corte, utilizando previamente una serie de sub-rutinas dentro de los que se encuentra el algoritmo de la figura 3.17, el mismo que es utilizado para detectar píxeles de color.
6. Fue posible la realización de cortes con penetración controlada, utilizando 2 técnicas de control diferentes, tal como se puede ver en la sección 3.11. En aquella sección se evaluó el uso de 3 técnicas de control que fueron: un controlador PID, un controlador difuso, y un controlador proporcional integral difuso. De estas, se eligió utilizar la técnica PI-difuso para formar parte del algoritmo de corte con penetración controlada, debido a la mayor versatilidad de implementación que ofrece. Esto sugiere que aquella técnica puede resultar ampliamente aplicable en robots con diferentes características dinámicas, siempre que ejecuten la tarea de realizar cortes con penetración controlada con bisturí en condiciones similares a las que se emplean en el presente trabajo de tesis (véase la sección 3.1).
7. Fue posible desarrollar e implementar, en un programa de computadora, un algoritmo de corte con penetración controlada en base al uso combinado del algoritmo de estimación de la profundidad de corte, con la técnica de control de profundidad de corte. Se llegó a esta conclusión, debido a que el programa de computadora mostrado en [32], que implementa el algoritmo de corte con penetración controlada, utiliza como sub-rutina al algoritmo de la figura 3.39 para estimar la profundidad de corte y, posteriormente, utiliza como sub-rutina al algoritmo de la figura 3.56 para controlar la profundidad de corte.

RECOMENDACIONES

1. Se debe mantener las condiciones de luz controladas, tanto en su intensidad como en su ángulo de incidencia sobre la muestra. La presencia de alguna fuente de luz no controlada podría generar reflejos indeseados que podrían perjudicar la obtención de la pseudo-región del bisturí y con ello también se perjudicaría la estimación de la profundidad de corte. Así mismo, se recomienda utilizar luz blanca, a fin de no alterar la percepción de color de los objetos contenidos en las imágenes tomadas.
2. Es necesario ubicar las fuentes de luz en posiciones que no permitan su reflejo en la hoja del bisturí hacia la lente de la cámara de video, a fin de evitar la generación de sub-regiones dentro de la hoja del bisturí que dificulten la obtención exitosa de la pseudo-región del bisturí.
3. Es necesario que la superficie de cada muestra sea plana. La presencia de alguna protuberancia delante de la zona a cortar podría ocultar parte del bisturí generando así la errada percepción visual de que el bisturí ha penetrado en la muestra.
4. De ser necesario, se debe inclinar ligeramente la cámara de video y/o la plataforma de muestras, a fin de evitar que otros objetos como la muestra o la cámara de video se vean reflejados en sobre la superficie de la hoja del bisturí, en las imágenes tomadas.
5. Es importante procurar que la cámara de video capture y entregue imágenes con colores visiblemente bien definidos. Esta recomendación se fundamenta en que si los colores no están bien definidos, podría resultar imposible obtener las pseudo-marcas de color y en consecuencia no se podría obtener exitosamente la imagen reducida, ni estimar satisfactoriamente la profundidad de corte.

BIBLIOGRAFÍA

- [1] **Acemoglu, A., Fichera, L., Kepiro, I., Caldwell, D., y Mattos, L.** (2017). *Laser Incision Depth Control in Robot-Assisted Soft Tissue Microsurgery*. *Journal of Medical Robotics Research*. NO 3. VOL 2. PP 1740006. DOI: 10.1142/S2424905X17400062.
- [2] **Alldatasheet** (2021). *MG996R High Torque Metal Gear Dual Ball Bearing Servo*. Alldatasheet [en línea]. [Fecha de acceso 15 de Septiembre del 2021]. URL disponible en: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1131873/ETC2/MG996R.html>.
- [3] **Aw, K., Fu, L., y Mcdaid, A.** (2013). *An IPMC actuated robotic surgery end effector with force sensing*. *International Journal of Smart and Nano Materials*. NO 4. VOL 4. PP 246–256. DOI: 10.1080/19475411.2013.862579.
- [4] **Canny, J.** (1986). *A Computational Approach to Edge Detection*. En *IEEE Transactions on Pattern Analysis and Machine Intelligence*. NO 6. VOL PAMI-8. PP 679-698. DOI: 10.1109/T-PAMI.1986.4767851.
- [5] **CCS** (2021). *Pcw ide compiler for microchip pic10/12/16 devices*. Custom Computer Services Inc. [en línea]. [Fecha de acceso 15 de Septiembre del 2021]. URL disponible en: http://www.ccsinfo.com/product_info.php?products_id=PCW_full.
- [6] **Chanthasopeephan, T., Desai, J. P., y Lau, A. C. W.** (2003). *Measuring Forces in Liver Cutting for Reality-based Haptic Display*. En *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)* (Cat. No.03CH37453). VOL 3. PP 3083–3088. DOI: 10.1109/IROS.2003.1249630.
- [7] **Chanthasopeephan, T., Desai, J. P., y Lau, A. C.W.** (2006). *Determining Fracture Characteristics in Scalpel Cutting of Soft Tissue*. En *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics 2006*. BioRob 2006. PP 899–904. DOI: 10.1109/BIOROB.2006.1639205.
- [8] **Deshpande, N., Ortiz, J., Caldwell, D. G., y Mattos, L. S.** (2014). *Enhanced Computer-*

Assisted Laser Microsurgeries with a “Virtual Microscope” based surgical system. En 2014 IEEE *International Conference on Robotics and Automation (ICRA)*. Hong Kong, China. PP 4194–4199. DOI: 10.1109/ICRA.2014.6907469.

[9] **EEZYrobots** (2018). EEZYbotARM MK2. EEZYrobots [en línea]. [Fecha de acceso 15 de Julio del 2021]. URL disponible en: http://www.eezyrobots.it/eba_mk2.html.

[10] **Fichera, L., Pardo, D., Illiano, P., Caldwell, D. G., y Mattos, L. S.** (2015). *Feed Forward Incisión Control for Laser Microsurgery of Soft Tissue*. En 2015 IEEE *International Conference on Robotics and Automation (ICRA)*. Seattle, Washington, USA. PP 1235–1240. DOI: 10.1109/ICRA.2015.7139349.

[11] **Ford, A. y Roberts, A.** (1998). *Colour Space Conversions*. Poynton.ca [en línea]. [Fecha de acceso 24 de Noviembre del 2020]. URL disponible en: <https://poynton.ca/PDFs/coloureq.pdf>.

[12] **FTDI** (2020). FT232R USB UART IC *Datasheet*. *Future Technology Devices International Ltd.* [en línea]. [Fecha de acceso 18 de Noviembre del 2021]. URL disponible en: https://ftdichip.com/wp-content/uploads/2020/08/DS_FT232R.pdf.

[13] **Heath, J.** (2017). PWM: *Pulse Width Modulation: What is it and how does it work?*. Analog IC Tips [en línea]. [Fecha de acceso 15 de Julio del 2021]. URL disponible en: <https://www.analogictips.com/pulse-width-modulation-pwm/>.

[14] **Ibraheem, N., Hasan, M., Khan, R. Z., y Mishra, P.** (2012). *Understanding Color Models: A Review*. *ARNP Journal of Science and Technology*. NO 3. VOL 2. PP 265–275. ISSN 2225-7217.

[15] **Intuitive** (2020). *Da Vinci Surgical Systems*. Intuitive [en línea]. [Fecha de acceso 1 de Noviembre del 2020]. URL disponible en: <https://www.intuitive.com/en-us/products-and-services/da-vinci/systems>.

[16] **ISO** (2004). *International Standard ISO 9409-1. The International Organization for Standardization*. 3ra. Edición. ICS : 25.040.30.

[17] **Kanagaraj, N., Sivashanmugam, P., y Paramasivam, S.** (2008). *Fuzzy Coordinated*

PI Controller: Application to the Real-Time Pressure Control Process. Advances in Fuzzy Systems. VOL 2008. PP 1-9. DOI: 10.1155/2008/691808.

[18] **Khuntia, S.R., Mohanty, K.B., Panda S., y Ardil, C.** (2011). *Comparative Study of P-I, I-P, Fuzzy and Neuro-Fuzzy Controllers for Speed Control of DC Motor Drive.* *World Academy of Science, Engineering and Technology International Journal of Electrical and Computer Engineering.* NO 5. VOL 5. PP 714-718. ISSN: 1307-6892.

[19] **Kirch, W.** (2008). *Pearson's Correlation Coefficient. Encyclopedia of Public Health.* Springer. Dordrecht, Países bajos. PP 1090–1091. ISBN 978-1-4020-5614-7. DOI: 10.1007/978-1-4020-5614-7_2569.

[20] **Lee, K. H.** (2005). *First Course on Fuzzy Theory and Applications.* 1era. Edición. Heidelberg, Berlín, Alemania . Springer. ISSN: 1615-3871.

[21] **Long, P., Khalil, W., y Martinet, P.** (2014). *Force/vision control for robotic cutting of soft materials.* En *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems.* Chicago, Illinois, USA. PP 4716–4721. DOI: 10.1109/IROS.2014.6943233.

[22] **Mallick, S.** (2015). *Filling holes in an image using OpenCV (python / C++).* LearnOpenCV [en línea]. [Fecha de acceso 7 de Noviembre del 2021]. URL disponible en: <https://learnopencv.com/filling-holes-in-an-image-using-opencv-python-c/>.

[23] **Mattos, L. S., Dellepiane, M., y Caldwell, D. G.** (2011). *Next-Generation Micromanipulator for Computer-Assisted Laser Phonomicrosurgery.* En *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* Boston, Massachusetts, USA. PP 4555–4559. DOI: 10.1109/IEMBS.2011.6091128.

[24] **Microchip** (2006). *PIC18f2455/2550/4455/4550 Data Sheet.* Microchip [en línea]. [Fecha de acceso 16 de Noviembre del 2021]. URL disponible en: <https://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf>.

[25] **Microsoft** (2021). *Codifique más rápido.* Microsoft [en línea]. [Fecha de acceso 15 de Septiembre del 2021]. URL disponible en: <https://visualstudio.microsoft.com/es/vs/>.

[26] **Ogata, K.** (2010). *Ingeniería de control moderna.* Pearson Education, S.A., 5ta. Edición.

Madrid, España. ISBN: 978-84-8322-660-5.

[27] **Plataniotis, K. y Venetsanopoulos, A.** (2000). *Color Image Processing and Applications*. 1era. Edición. New York, USA. Springer. DOI: 10.1007/978-3-662-04186-4.

[28] **Robotiq** (2018). *Robotiq FT300 Force Torque Sensor - Instruction Manual*. Robotiq Inc. [en línea]. [Fecha de acceso 16 de Noviembre del 2021]. URL disponible en: https://assets.robotiq.com/website-assets/support_documents/document/FT_Sensor_Instruction_Manual_PDF_20181218.pdf.

[29] **Shi, P., Cheng, B., Zong, Y., Gong, Y., Yan, H., Hu, Y., Zhang, H., Kuang, S., Liu, W., Wang, L., y Su, B.** (2018). *A Soft Tissue Scalpel Cutting Robotic System with Sucker Fixation*. En 2018 IEEE 14th International Conference on Control and Automation (ICCA). Anchorage, Alaska, USA. PP 1162–1167. DOI: 10.1109/ICCA.2018.8444361.

[30] **So, W.-C., Tse, C., y Lee, Y.-S.** (1996). *Development of a Fuzzy Logic Controller for DC/DC Converters: Design, Computer Simulation, and Experimental Evaluation*. IEEE Transactions on Power Electronics. NO 1. VOL 11. PP 24–32. DOI: 10.1109/63.484413.

[31] **Su, B., Wang, X., y Liao, H.** (2015). *Soft Tissue Cutting Control Method with Surgical Scalpel*. En *The 11th Asian Conference on Computer Aided Surgery (ACCAS 2015)*. DOI 10.13140/RG.2.1.3662.6086.

[32] **Suárez-Quispe, J. C.** (2022). Programa de corte con penetración controlada. Blog personal de investigación [en línea]. [Fecha de acceso 18 de Marzo del 2022]. URL disponible en: <https://juansuarezq.blogspot.com/2022/03/programa-de-corte-con-penetracion.html>.

[33] **Suárez-Quispe, J. C.** (2022). Programa de la interface electrónica del robot MK2 Modificado. Blog personal de investigación [en línea]. [Fecha de acceso 18 de Marzo del 2022]. URL disponible en: <https://juansuarezq.blogspot.com/2022/03/programa-de-la-interface-electronica.html>.

[34] **Suarez-Quispe, J. C. y Ramos, O. E.** (2020). *Scalpel Region Detection based on the Location of Color Marks and Edge Detection*. En 2020 IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON). PP 1–4. DOI: 10.1109/INTERCON50315.2020.9220207.

[35] **Thakur, M.** (2022). Pearson correlation coefficient. WallStreetMojo [en línea]. [Fecha de acceso 23 de Febrero del 2022]. URL disponible en: <https://www.wallstreetmojo.com/pearson-correlation-coefficient/>.

[36] **Universal-Robots** (2020). Universal robot UR5e. *Universal Robots* [en línea]. [Fecha de acceso 1 de Noviembre del 2020]. URL disponible en: <https://www.universal-robots.com/products/ur5-robot/>.

[37] **UTEC** (2021). Universidad de Ingeniería y Tecnología. UTEC [en línea]. [Fecha de acceso 15 de Julio del 2021]. URL disponible en: <https://www.utec.edu.pe/>.

[38] **Valdastri, P., Tognarelli, S., Menciassi, A., y Dario, P.** (2009). *A scalable platform for biomechanical studies of tissue cutting forces*. Meas. Sci. Technol. VOL 20. PP 45801–45801. DOI: 10.1088/0957-0233/20/4/045801.

[39] **Valdastri, P., Houston, K., Menciassi, A., Dario, P., Sieber, A., Yanagihara, M., y Fujie, M.** (2007). *Miniaturized Cutting Tool With Triaxial Force Sensing Capabilities for Minimally Invasive Surgery*. *Journal of Medical Devices*. NO 3. VOL 1. PP 206–211. DOI: 10.1115/1.2778700.

[40] **Yang, B., Wang, H., Chen, W., y Wang, Z.** (2016). *Vision-Based Cutting Control of Deformable Objects*. En 2016 IEEE *International Conference on Real-time Computing and Robotics (RCAR)*. Angkor Wat, Cambodia. PP 650–655. DOI: 10.1109/RCAR.2016.7784106.

ANEXO A

SUB-RUTINAS DE LÓGICA DIFUSA

A.1. La sub-rutina *rule*

La sub-rutina *rule* representa una implicación difusa de Mamdani (véase la sección 2.3.3). Siendo así, esta sub-rutina tiene dos entradas que son evaluadas con respecto a un conjunto difuso, cada una, para calcular sus grados de pertenencia con respecto a dichos conjuntos. Estos grados de pertenencia son a_1 y a_2 . La salida de esta sub-rutina es un vector que representa el perfil que se obtiene al truncar el conjunto difuso de salida. La sub-rutina *rule*, es descrita en el algoritmo de la figura A.1, y tiene las siguientes entradas, salidas y sub-rutinas:

- e : Primera entrada.
- ce : Segunda entrada.
- V_1 : Vector que define el conjunto difuso en el que se evalúa la primera entrada.
- V_2 : Vector que define el conjunto difuso en el que se evalúa la segunda entrada.
- V_3 : Vector que define el conjunto difuso en el que se evalúa la salida.
- V_0 : Perfil truncado del conjunto difuso de salida definido por V_3 .
- $trian_v$: Trunca el perfil del conjunto difuso de salida definido por V_3 .
- $trian_d$: Calcula el grado de pertenencia de e o ce con respecto al conjunto difuso definido por V_1 o V_2 , respectivamente.

Entrada: V_1, V_2, V_3, e, ce

Salida: V_0

- 1: $a := 0$
 - 2: $a_1 \leftarrow trian_d(V_1, e)$
 - 3: $a_2 \leftarrow trian_d(V_2, ce)$
 - 4: **si** $a_1 < a_2$ **entonces**
 - 5: $a := a_1$
 - 6: **si no**
 - 7: $a := a_2$
 - 8: **fin si**
 - 9: $V_0 := trian_v(V_3, a)$
-

Figura A.1: Algoritmo; *rule*.

La sub-rutina *rule* utiliza la sub-rutina *trian_v*. Esta sub-rutina (*trian_v*) permite calcular el perfil V de un conjunto difuso P truncado hasta un valor máximo igual al valor numérico y . La sub-rutina *trian_v* es descrita en el algoritmo de la figura A.2, y tiene las siguientes entradas, salidas y sub-rutinas:

P : Vector de 3 elementos que define el conjunto difuso cuyo perfil se busca truncar.

y : Valor máximo para truncar P .

V : Perfil truncado del conjunto difuso definido por P .

trian_d : Calcula el grado de pertenencia de t con respecto al conjunto difuso definido por P .

Entrada: P, y

Salida: V

1: $t := -1$

2: $V := \text{vector}$

3: $nv := \text{obtener la cantidad de elementos de } V$

4: $Ts := 2/(nv)$

5: **para** $i \leftarrow 1, nv$ **hacer**

6: $V(i) := \text{trian}_d(P, t)$

7: **si** $V(i) > y$ **entonces**

8: $V(i) := y$

9: **fin si**

10: $t := t + Ts$

11: **fin para**

Figura A.2: Algoritmo; *trian_v*.

Por su parte las sub-rutinas *rule* y *trian_v* utiliza la sub-rutina *trian_d*. Esta sub-rutina permite calcular el grado de pertenencia y de un valor numérico x con respecto a un conjunto difuso triangular definido por el vector de 3 elementos P . La sub-rutina *trian_d*, es descrita en el algoritmo de la figura A.3, y tiene las siguientes entradas y salidas:

x : Entrada.

P : Vector de 3 elementos que define el conjunto difuso en el que se evalúa la entrada.

y : Grado de pertenencia de x con respecto al conjunto difuso definido por P .

Entrada: P, x
Salida: y

- 1: $a := P(1)$
- 2: $b := P(2)$
- 3: $c := P(3)$
- 4: **si** $x \leq a$ **entonces**
- 5: $y := 0$
- 6: **fin si**
- 7: **si** $x > a \wedge x < b$ **entonces**
- 8: $y := (x - a)/(b - a)$
- 9: **fin si**
- 10: **si** $x = b$ **entonces**
- 11: $y := 1$
- 12: **fin si**
- 13: **si** $x > b \wedge x < c$ **entonces**
- 14: $y := (c - x)/(c - b)$
- 15: **fin si**
- 16: **si** $x \geq c$ **entonces**
- 17: $y := 0$
- 18: **fin si**

Figura A.3: Algoritmo; $trian_d$.

A.2. La sub-rutina $rule_K$

La sub-rutina $rule_K$ representa una implicación difusa de Mamdani (véase la sección 2.3.3). Siendo así, tiene dos entradas que son evaluadas con respecto a un conjunto difuso cada una para calcular sus grados de pertenencia con respecto a dichos conjuntos. Estos grados son a_1 y a_2 . La salida de esta sub-rutina es un vector que representa el perfil que se obtiene al trincar el conjunto difuso de salida. La sub-rutina $rule_K$ se diferencia de la sub-rutina $rule$ en las formas de los conjuntos difusos que utiliza. La sub-rutina es descrita en el algoritmo de la figura A.4, y tiene las siguientes entradas, salidas y sub-rutinas:

e : Primera entrada.

ce : Segunda entrada.

V_1 : Vector que define el conjunto difuso en el que se evalúa la primera entrada.

c_1 : Selector de la forma de conjunto difuso V_1 .

V_2 : Vector que define el conjunto difuso en el que se evalúa la segunda entrada.

c_2 : Selector de la forma de conjunto difuso V_2 .

V_3 : Vector que define el conjunto difuso en el que se evalúan la salida.

V_0 : Perfil truncado del conjunto difuso de salida definido por V_3 .

$trian_d$: Calcula el grado de pertenencia de e con respecto al conjunto difuso definido por

V_1 .

$trape_d$: Calcula el grado de pertenencia de ce con respecto al conjunto difuso definido por

V_2 .

$trian_{vK}$: Trunca el perfil del conjunto difuso de salida definido por V_3 .

Entrada: $V_1, c_1, V_2, c_2, V_3, e, ce$

Salida: V_0

```
1: si  $c_1 = 1$  entonces
2:    $a_1 \leftarrow trian_d(V_1, e)$ 
3: fin si
4: si  $c_1 = 2$  entonces
5:    $a_1 \leftarrow trape_d(V_1, e)$ 
6: fin si
7: si  $c_2 = 1$  entonces
8:    $a_2 \leftarrow trian_d(V_2, ce)$ 
9: fin si
10: si  $c_2 = 2$  entonces
11:    $a_2 \leftarrow trape_d(V_2, ce)$ 
12: fin si
13: si  $a_1 < a_2$  entonces
14:    $a := a_1$ 
15: si no
16:    $a := a_2$ 
17: fin si
18:  $V_0 := trian_{vK}(V_3, a)$ 
```

Figura A.4: Algoritmo; $rule_K$.

La sub-rutina $rule_K$ utiliza la sub-rutina $trian_{vK}$. Esta sub-rutina permite calcular el perfil V de un conjunto difuso P truncado hasta un máximo igual al valor numérico y . La sub-rutina $trian_{vK}$ se diferencia de la sub-rutina $trian_v$ en el rango de valores de la salida. La sub-rutina $trian_{vK}$ es descrita en el algoritmo de la figura A.5, y tiene las siguientes entradas, salidas y sub-rutinas:

P : Vector de 3 elementos que define el conjunto difuso cuyo perfil se busca truncar.

x : Valor máximo para truncar P .

V : Perfil truncado del conjunto difuso definido por P .

$trian_d$: Calcula el grado de pertenencia de t con respecto al conjunto difuso definido por P .

Entrada: P, y

Salida: V

1: $t := 0$

2: $V := \text{vector}$

3: $nv := \text{obtener la cantidad de elementos de } V$

4: $Ts := 1/(nv)$

5: **para** $i \leftarrow 1, nv$ **hacer**

6: $V(i) := \text{trian}_d(P, t)$

7: **si** $V(i) > y$ **entonces**

8: $V(i) := y$

9: **fin si**

10: $t := t + Ts$

11: **fin para**

Figura A.5: Algoritmo; trian_{vK} .

Por su parte la sub-rutina rule_K también la sub-rutina trape_d . Esta sub-rutina permite calcular el grado de pertenencia y de un valor numérico x con respecto a un conjunto difuso trapezoidal definido por el vector de 4 elementos P . La sub-rutina trape_d , es descrita en el algoritmo de la figura A.6, y tiene las siguientes entradas y salidas:

x : Entrada.

P : Vector de 4 elementos que define el conjunto difuso en el que se evalúa la entrada.

y : Grado de pertenencia de x con respecto al conjunto difuso definido por P .

Entrada: P, x

Salida: y

1: $a := P(1)$

2: $b := P(2)$

3: $c := P(3)$

4: $d := P(4)$

5: **si** $x \leq a$ **entonces**

6: $y := 0$

7: **fin si**

8: **si** $x > a \wedge x < b$ **entonces**

9: $y := (x - a)/(b - a)$

10: **fin si**

11: **si** $x \geq b \wedge x \leq c$ **entonces**

12: $y := 1$

13: **fin si**

14: **si** $x > c \wedge x < d$ **entonces**

15: $y := (d - x)/(d - c)$

16: **fin si**

17: **si** $x \geq d$ **entonces**

18: $y := 0$

19: **fin si**

Figura A.6: Algoritmo; $trape_d$.

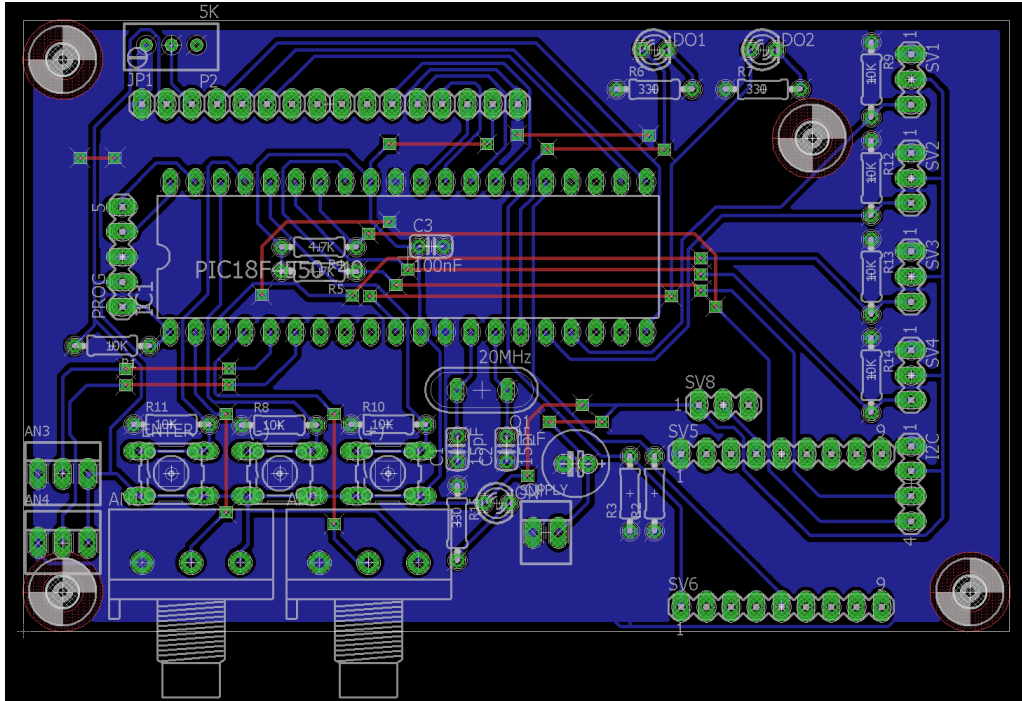


Figura B.2: Diseño de la tarjeta electrónica impresa de la interface electrónica de operación del robot MK2 modificado.

En la figura B.3 se muestra la interface electrónica de operación del robot MK2 modificado fabricada, con un módulo conversor USB a TTL acoplado a ella. El módulo conversor está basado en el circuito integrado FT232RL [12].

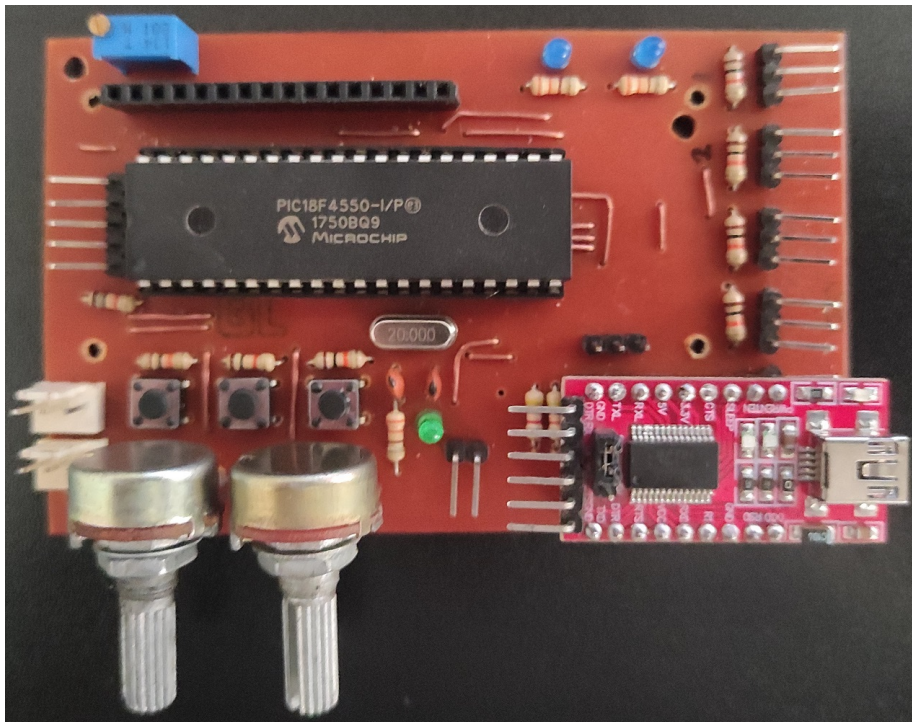


Figura B.3: Interface electrónica de operación del robot MK2 modificado.

ANEXO C
MEDICIÓN DE LA PROFUNDIDAD DE CORTE

En las siguientes figuras se muestran las imágenes de las caras de sección de las 30 pruebas de cortes controlados, cuyos resultados fueron expuestos en las tablas 4.1, 4.2, y 4.3. Así mismo, se muestran también las profundidades de corte obtenidas en cada caso.

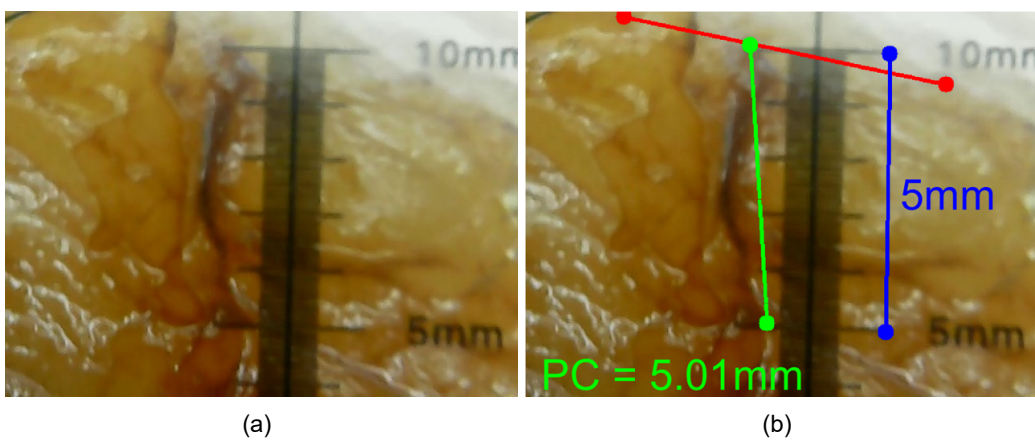


Figura C.1: Prueba #1; cara seccionada (a), medición de la profundidad de corte (b).

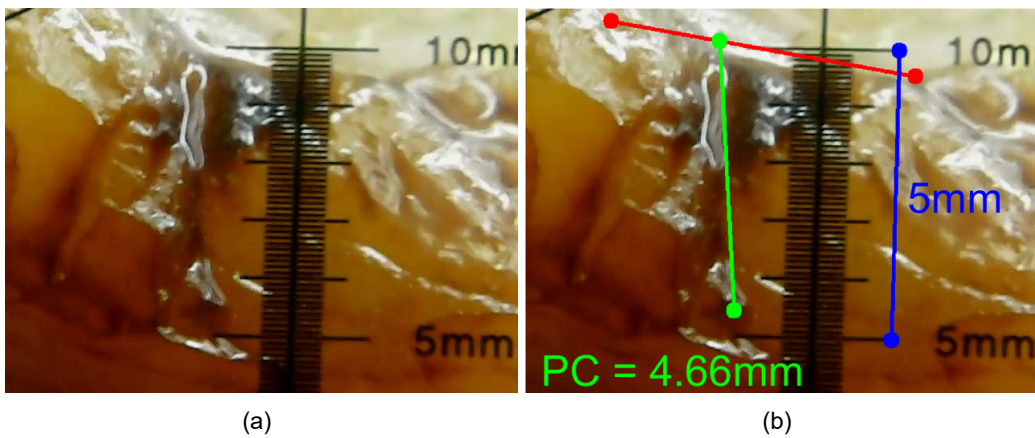


Figura C.2: Prueba #2; cara seccionada (a), medición de la profundidad de corte (b).

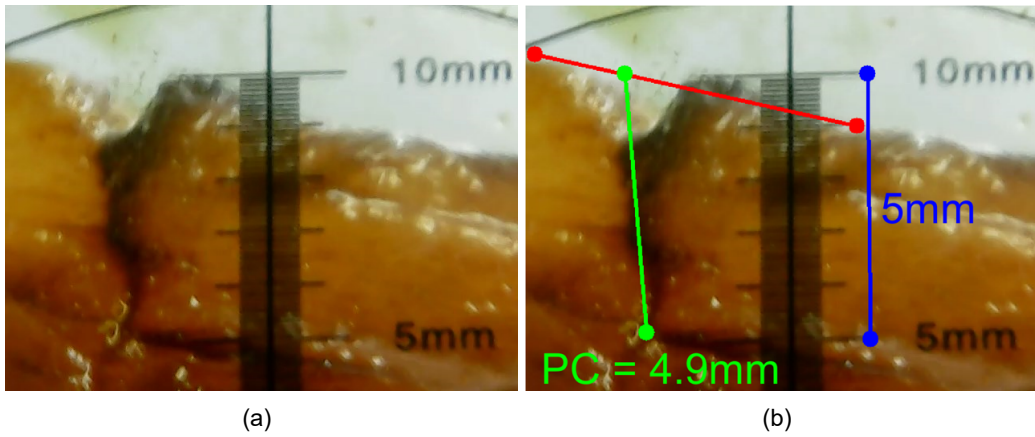


Figura C.3: Prueba #3; cara seccionada (a), medición de la profundidad de corte (b).

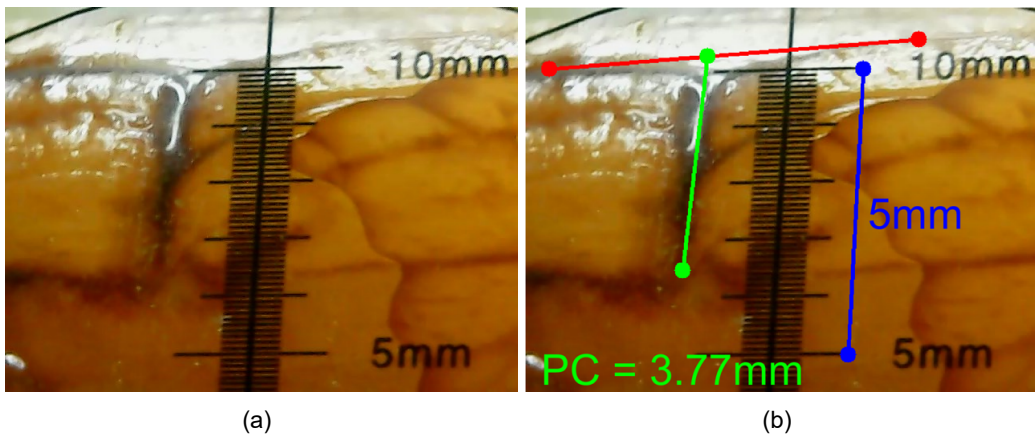


Figura C.4: Prueba #4; cara seccionada (a), medición de la profundidad de corte (b).

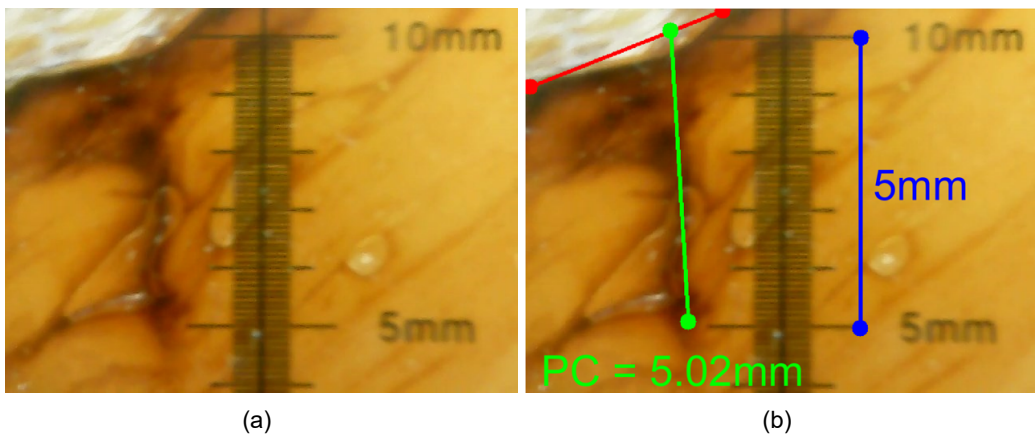


Figura C.5: Prueba #5; cara seccionada (a), medición de la profundidad de corte (b).

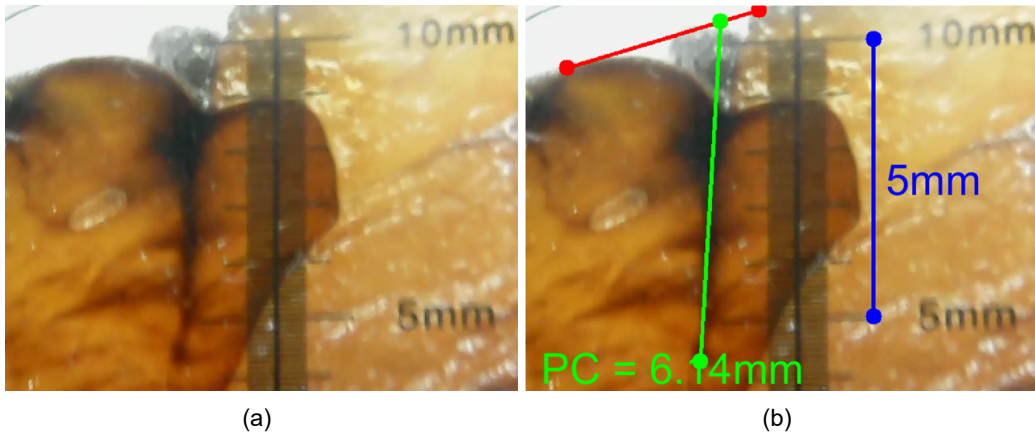


Figura C.6: Prueba #6; cara seccionada (a), medición de la profundidad de corte (b).

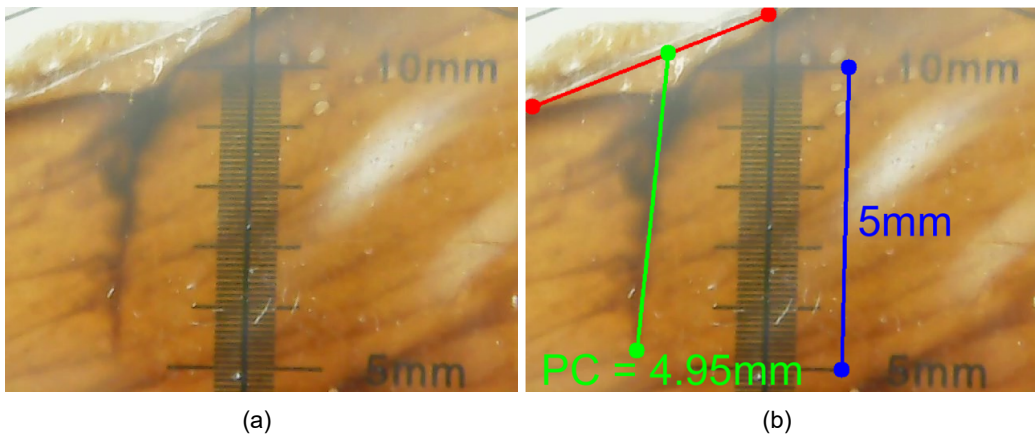


Figura C.7: Prueba #7; cara seccionada (a), medición de la profundidad de corte (b).

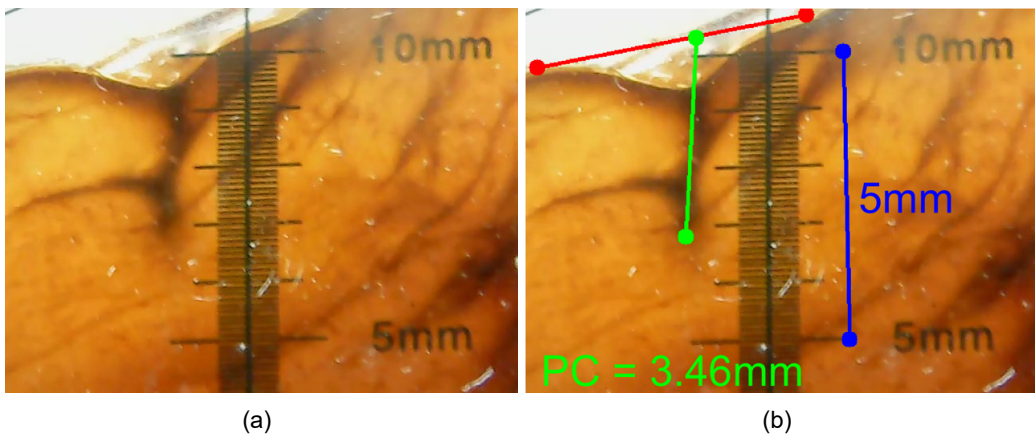


Figura C.8: Prueba #8; cara seccionada (a), medición de la profundidad de corte (b).

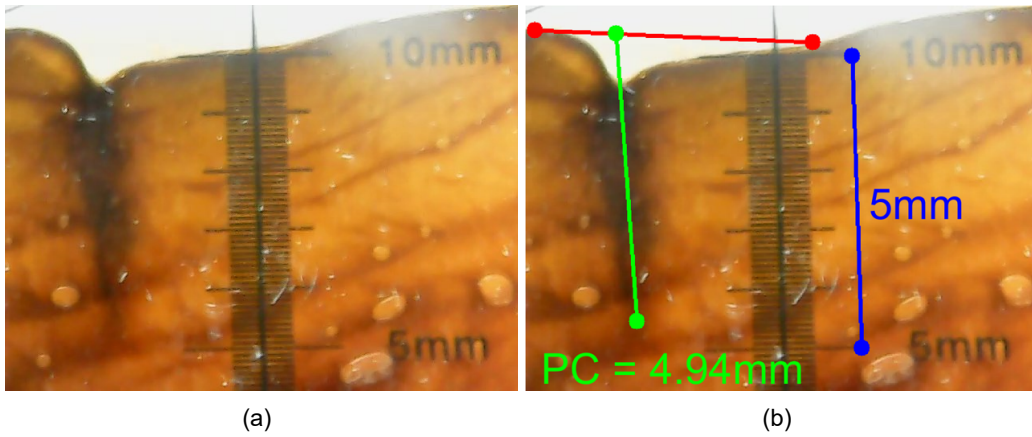


Figura C.9: Prueba #9; cara seccionada (a), medición de la profundidad de corte (b).

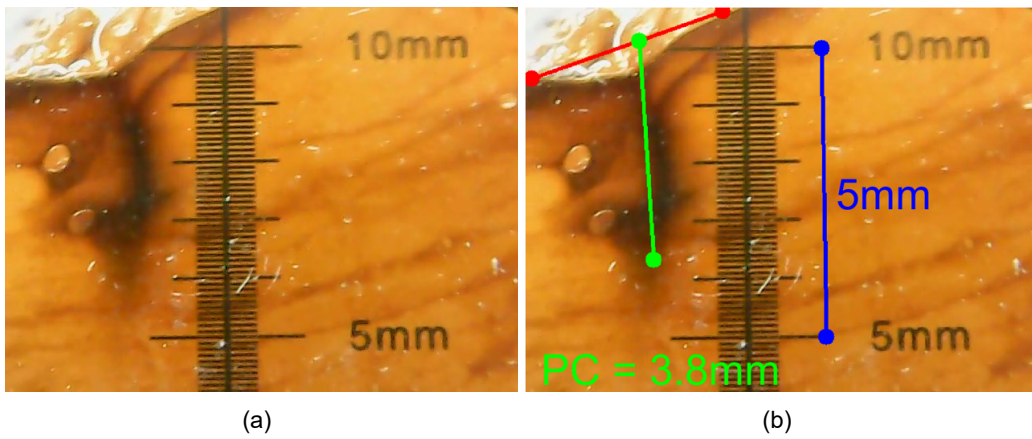


Figura C.10: Prueba #10; cara seccionada (a), medición de la profundidad de corte (b).

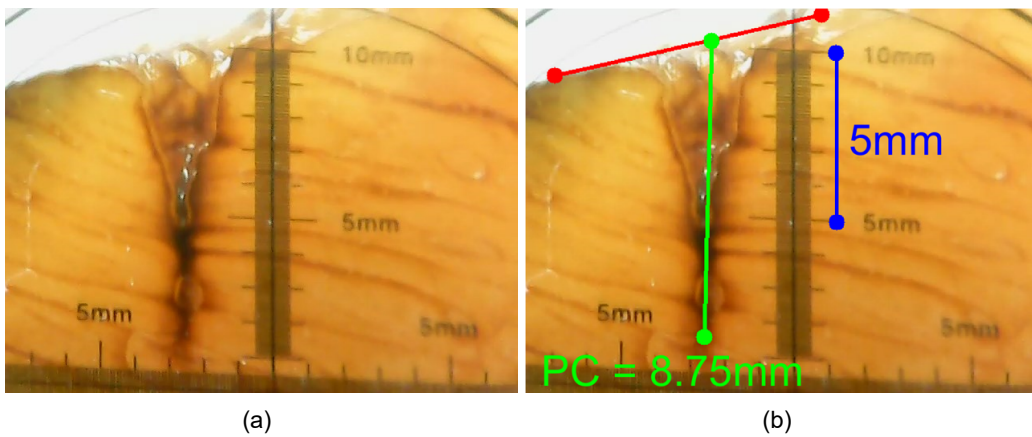


Figura C.11: Prueba #11; cara seccionada (a), medición de la profundidad de corte (b).

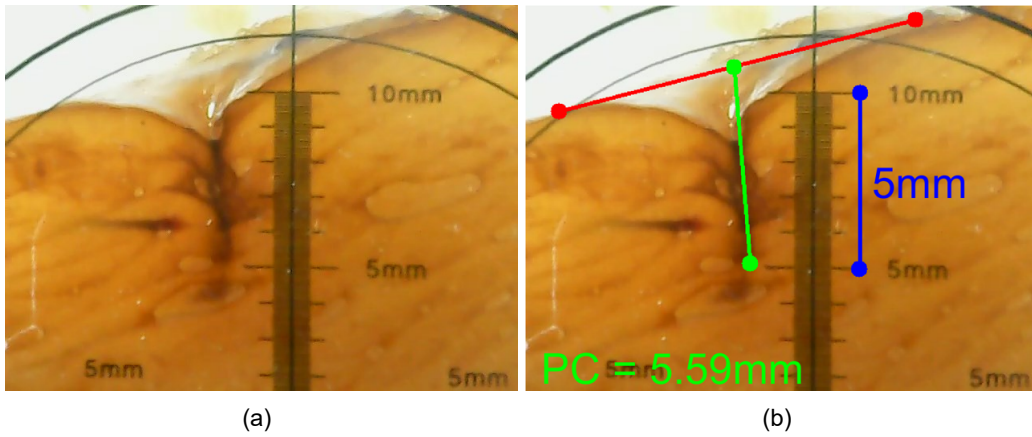


Figura C.12: Prueba #12; cara seccionada (a), medición de la profundidad de corte (b).

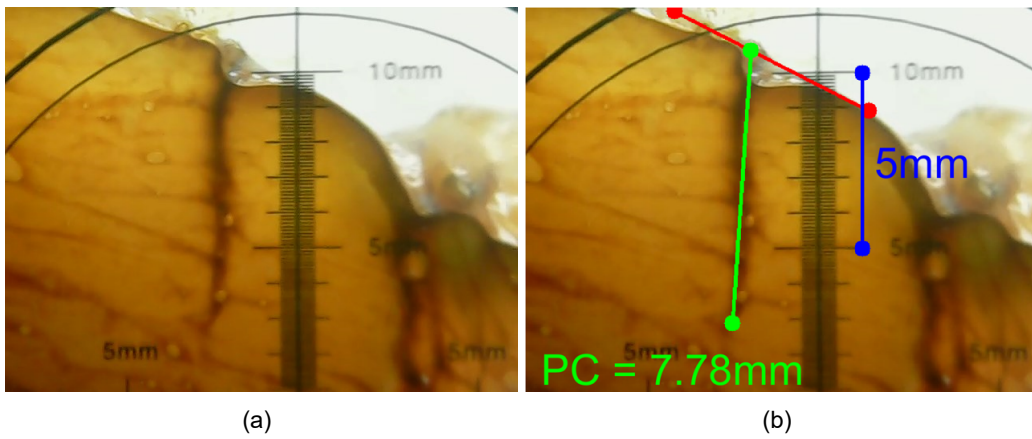


Figura C.13: Prueba #13; cara seccionada (a), medición de la profundidad de corte (b).

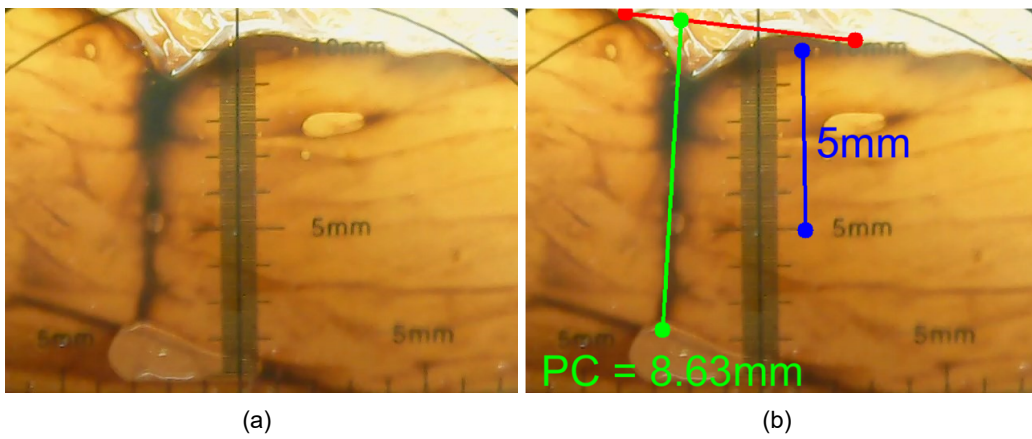


Figura C.14: Prueba #14; cara seccionada (a), medición de la profundidad de corte (b).

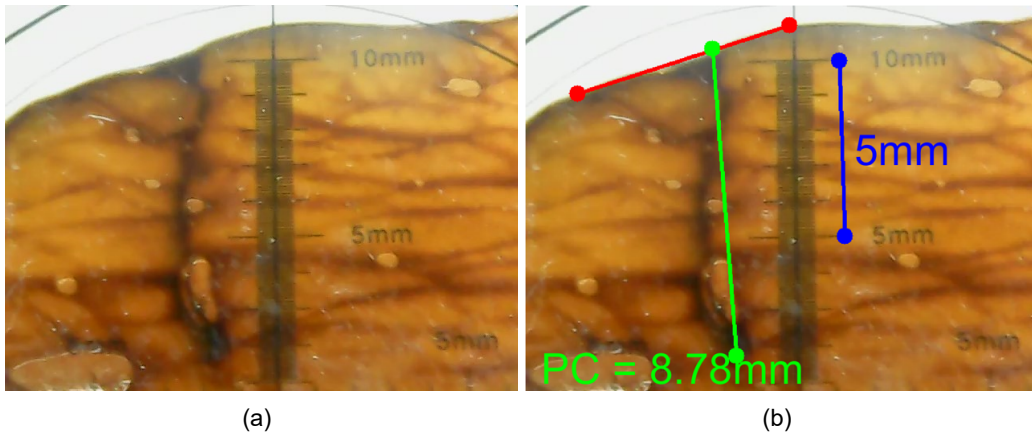


Figura C.15: Prueba #15; cara seccionada (a), medición de la profundidad de corte (b).

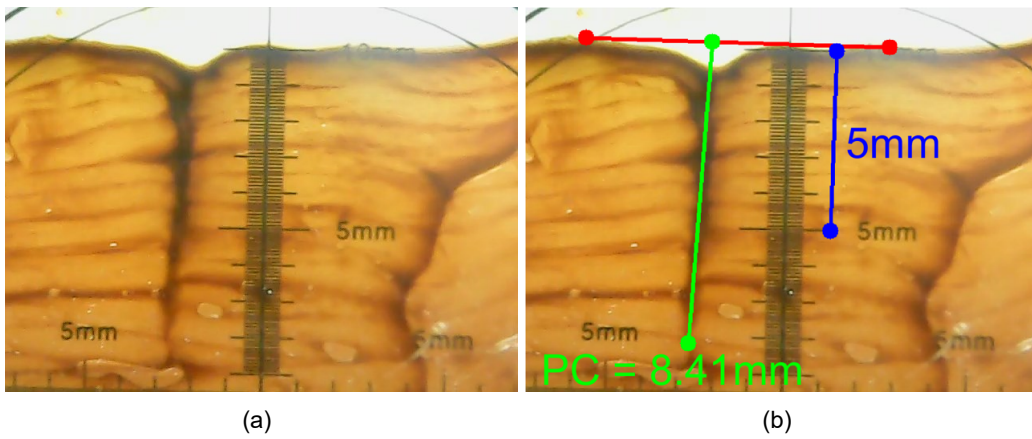


Figura C.16: Prueba #16; cara seccionada (a), medición de la profundidad de corte (b).

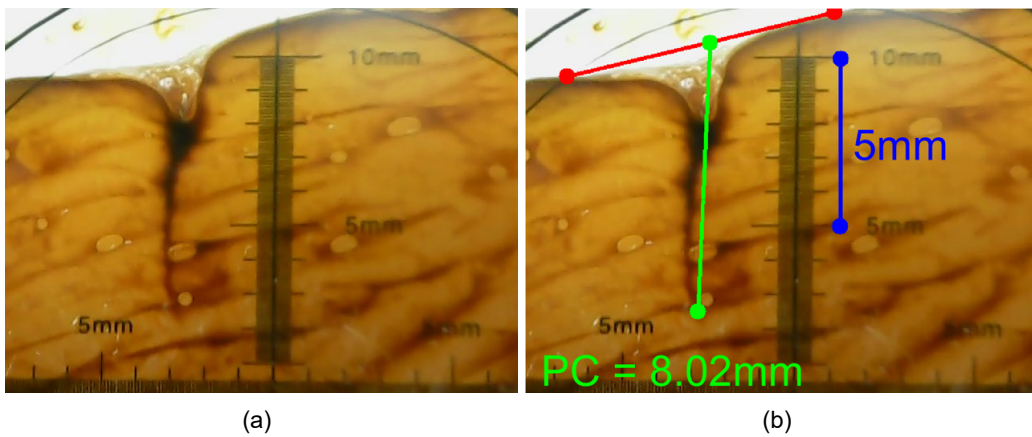


Figura C.17: Prueba #17; cara seccionada (a), medición de la profundidad de corte (b).

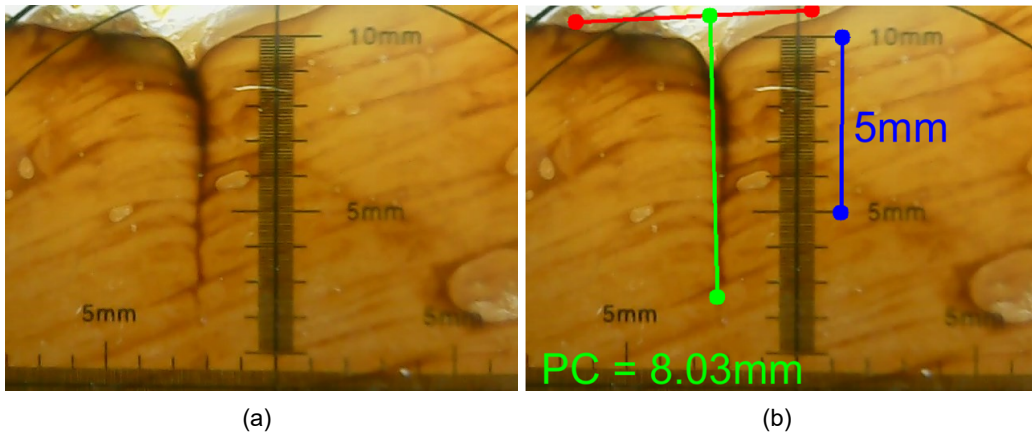


Figura C.18: Prueba #18; cara seccionada (a), medición de la profundidad de corte (b).

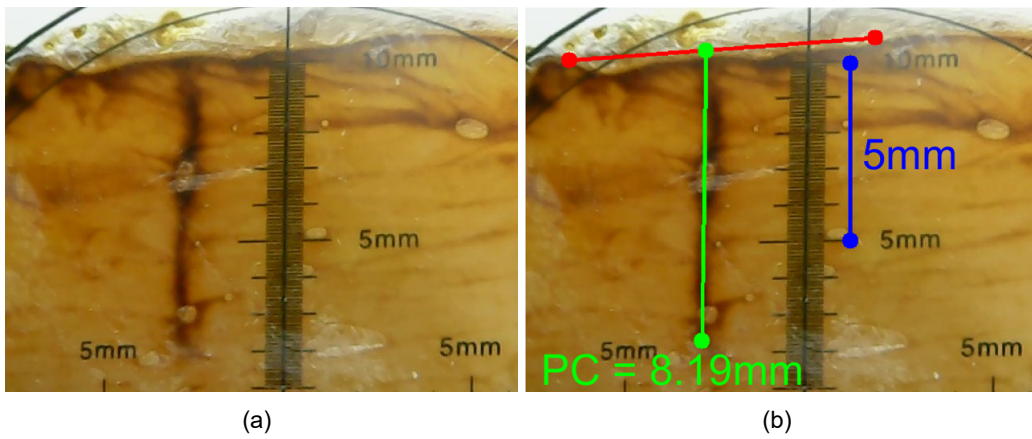


Figura C.19: Prueba #19; cara seccionada (a), medición de la profundidad de corte (b).

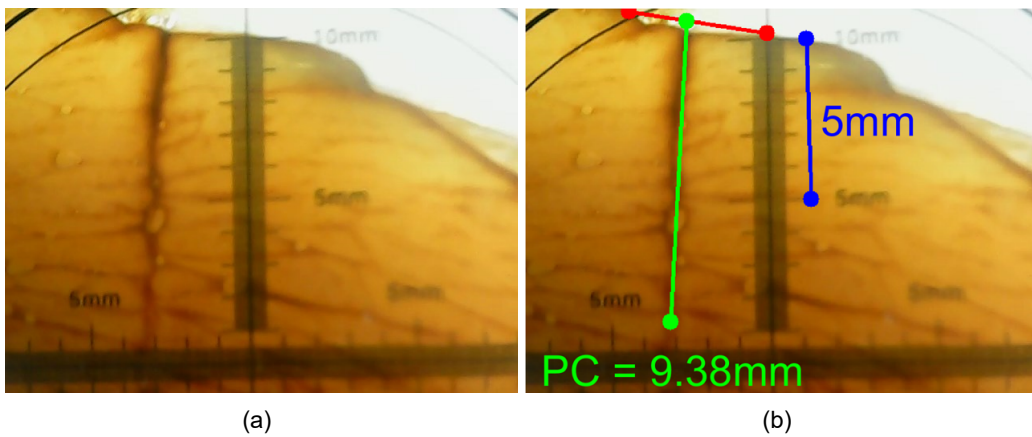


Figura C.20: Prueba #20; cara seccionada (a), medición de la profundidad de corte (b).

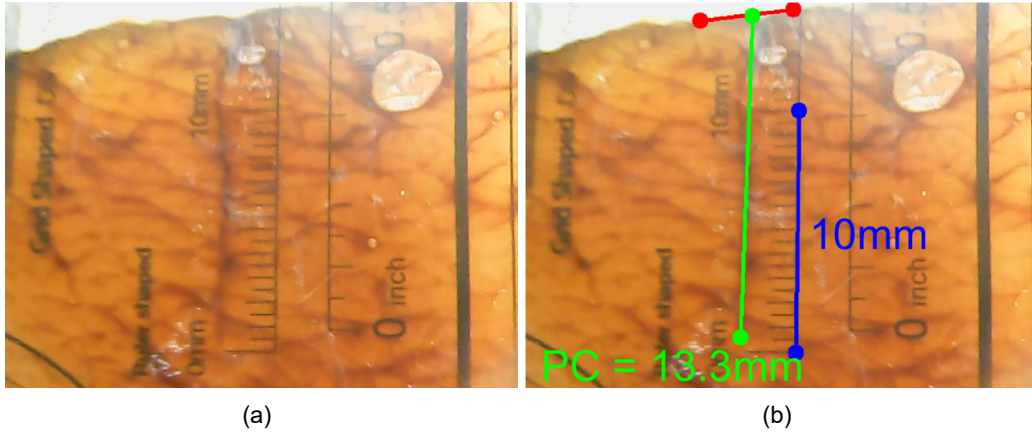


Figura C.21: Prueba #21; cara seccionada (a), medición de la profundidad de corte (b).

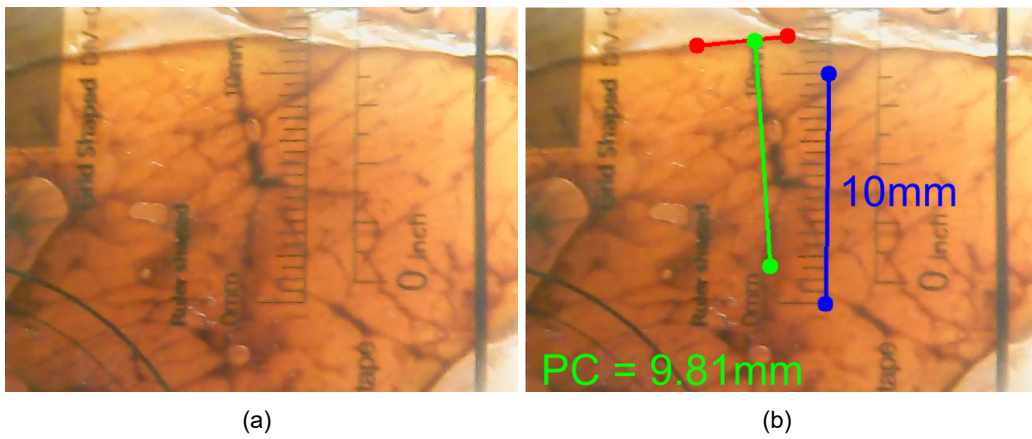


Figura C.22: Prueba #22; cara seccionada (a), medición de la profundidad de corte (b).

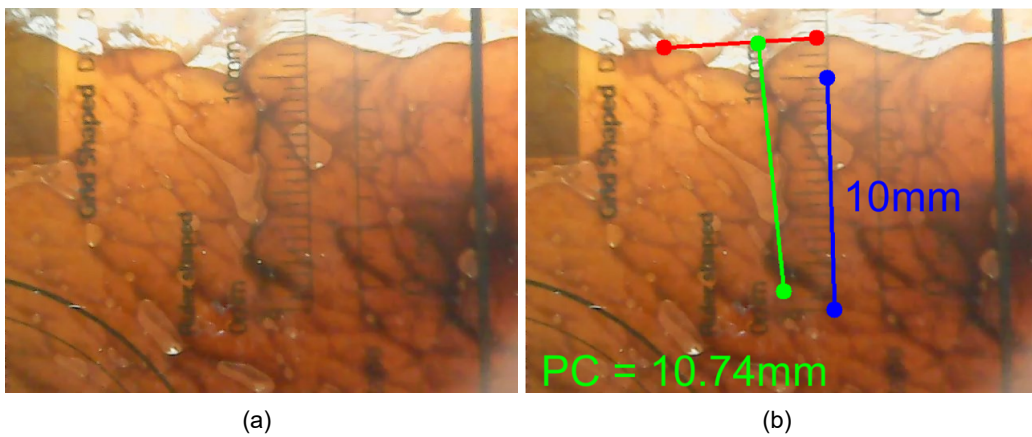


Figura C.23: Prueba #23; cara seccionada (a), medición de la profundidad de corte (b).

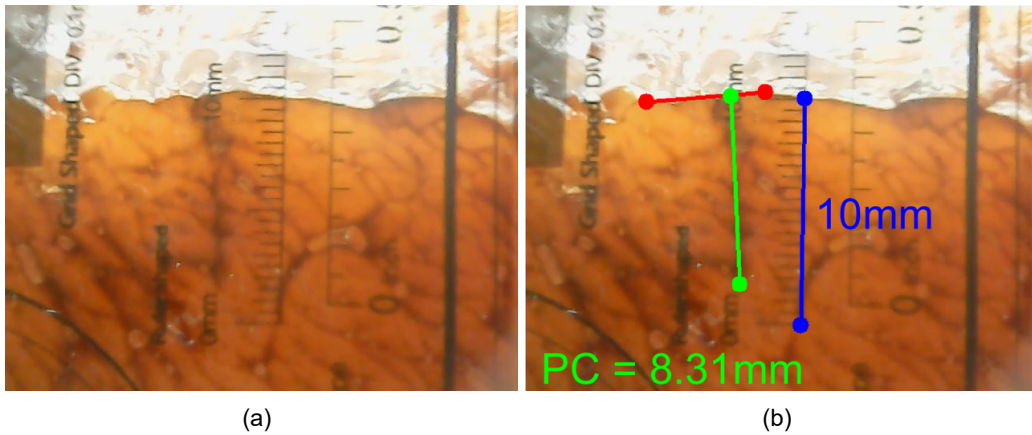


Figura C.24: Prueba #24; cara seccionada (a), medición de la profundidad de corte (b).

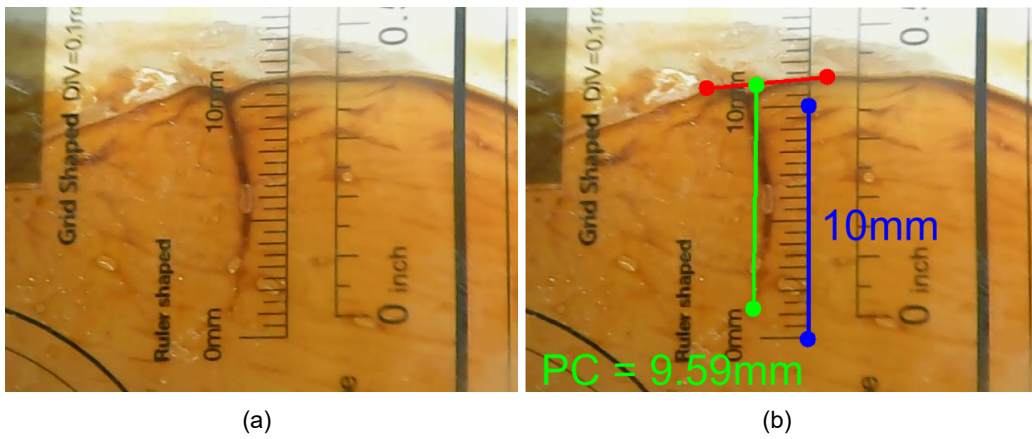


Figura C.25: Prueba #25; cara seccionada (a), medición de la profundidad de corte (b).

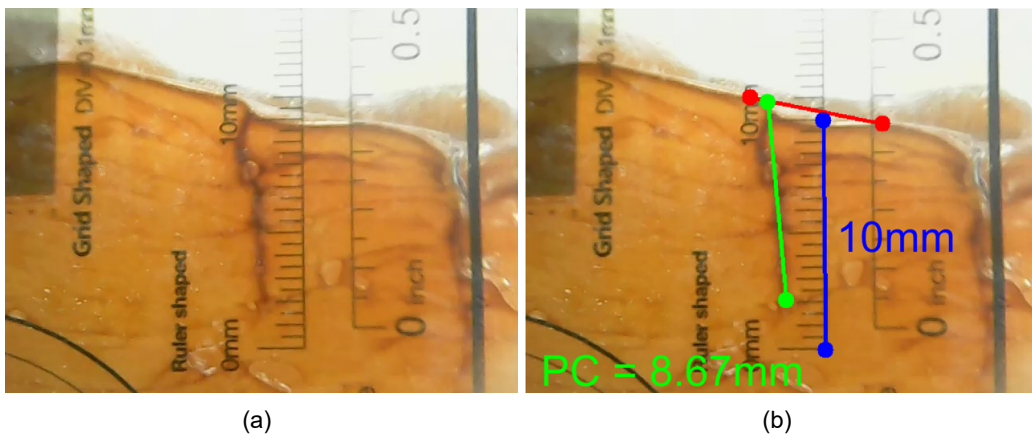
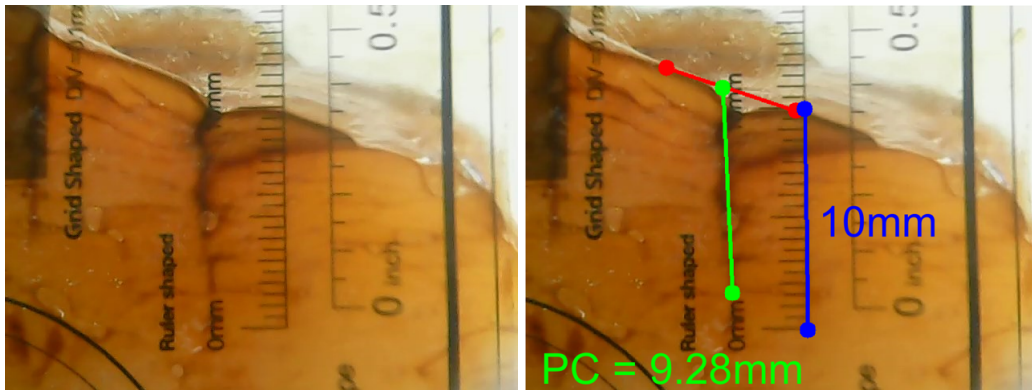


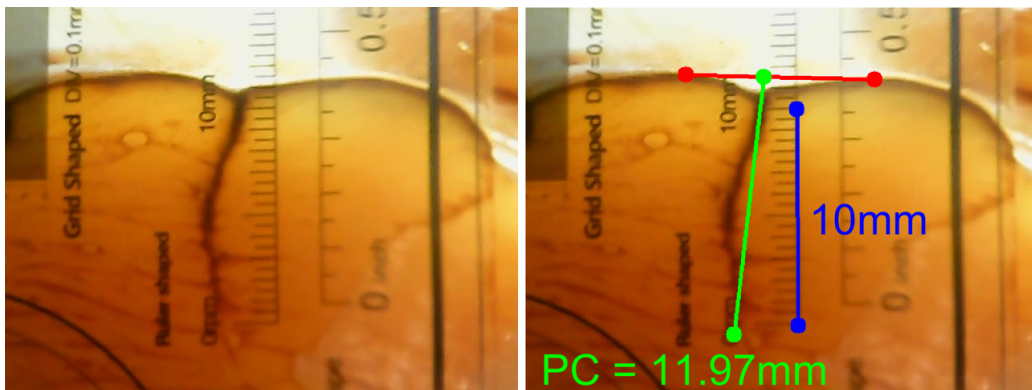
Figura C.26: Prueba #26; cara seccionada (a), medición de la profundidad de corte (b).



(a)

(b)

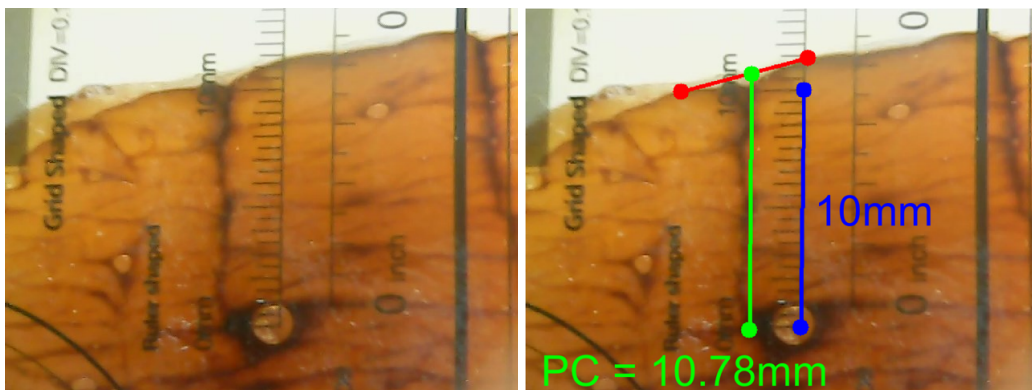
Figura C.27: Prueba #27; cara seccionada (a), medición de la profundidad de corte (b).



(a)

(b)

Figura C.28: Prueba #28; cara seccionada (a), medición de la profundidad de corte (b).



(a)

(b)

Figura C.29: Prueba #29; cara seccionada (a), medición de la profundidad de corte (b).

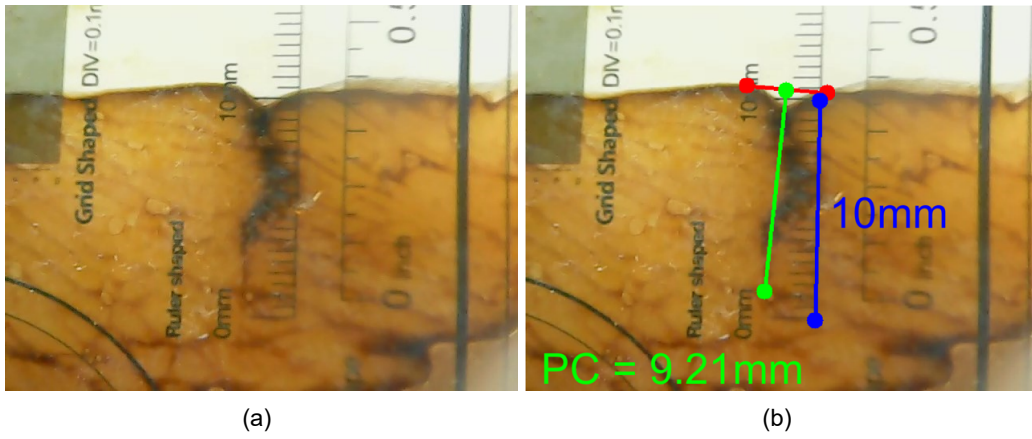


Figura C.30: Prueba #30; cara seccionada (a), medición de la profundidad de corte (b).