

**PONTIFICIA UNIVERSIDAD
CATÓLICA DEL PERÚ**

Escuela de Posgrado



Diseño de controlador neuro-difuso para automatización de detección de fallas dentro de tuberías matrices de agua empleando un robot móvil.

Tesis para obtener el grado académico de Magíster en Ingeniería de Control y Automatización que presenta:

Jorge Luis Isaac Ramírez Chávez

Asesor:

Francisco Fabian Cuellar Cordova

Lima, 2022

RESUMEN

En los últimos años se ha evidenciado un aumento de desastres relacionados con las red de tuberías de agua en el país, ya que no se cuenta con suficientes herramientas tecnológicas que permitan evaluar su estado, especialmente en las que presentan mayor antigüedad, por lo que no es posible identificar fallas ni evitar la ocurrencia de desastres como inundaciones y desabastecimiento de agua potable.

Esta tesis propone el desarrollo de un robot móvil con la capacidad de realizar el sensado automático de fallas al inspeccionar el interior de tuberías matrices de agua empleando un robot móvil. Para ello se realiza la implementación de un Sistema de Detección de Fallas (SDF) que identifica diferentes anomalías dentro una tubería empleando redes neuronales convolucionales y un Controlador de Velocidad y Orientación del Robot (CVOR) que empleando la información recolectada por el SDF permitirá que el robot se desplace de manera autónoma. De manera que será posible detectar e identificar fallas en las tuberías de agua tales como: obstrucciones, fisuras y fugas debido a roturas o perforaciones de manera automática.

El primer capítulo describe el estado del arte de los robots empleados en inspección y la problemática. El segundo capítulo desarrolla el modelo cinemático de un robot móvil con cuatro ruedas. En el tercer capítulo se diseña el Sistema de Detección de Fallas (SDF). El cuarto capítulo trata sobre la implementación del Controlador de Velocidad y Orientación del Robot (CVOR). Los resultados obtenidos, mostrados en el quinto capítulo, evidencian la precisión en el reconocimiento de anomalías por parte del sistema bajo determinadas condiciones.

ÍNDICE

RESUMEN.....	ii
CAPÍTULO 1: INTRODUCCIÓN Y PROBLEMÁTICA.....	1
1.1 Introducción.....	1
1.2 Problemática.....	2
1.3 Métodos tradicionales de detección de fallas en tuberías.....	3
1.4 Robots móviles terrestres para inspección de tuberías.....	9
1.5 Objetivo de la tesis.....	10
1.5.1 Objetivo General.....	10
1.5.2 Objetivo específicos.....	11
CAPÍTULO 2: MODELADO DE ROBOT MÓVIL.....	12
2.1 Modelo cinemático del robot móvil.....	12
2.2 Modelo dinámico del robot móvil.....	15
2.3 Simulación de modelo dinámico de robot.....	17
2.4 Estructura de control planteada.....	23
CAPÍTULO 3: DISEÑO DE SDF.....	25
3.1 Adquisición de datos.....	25
3.1.1 Recolección de imágenes.....	25
3.1.2 Anomalías a identificar.....	26
3.2 Marco teórico de sistemas neuronales para la detección automática de características.....	27
3.2.1 Redes neuronales.....	27
3.2.2 Funciones de activación.....	28
3.2.3 Entrenamiento de las redes neuronales.....	28
3.2.4 Redes Neuronales Convolucionales.....	29
3.2.5 Aplicaciones de las CNN.....	31
3.3 Diseño del clasificador para el SDF.....	32
3.4 Implementación de CNN.....	39
3.4.1 Recolección y preprocesamiento de información de entrada.....	39
3.4.2 Acondicionamiento de computadora e instalación de paquetes.....	40
3.4.3 Entrenamiento de CNN.....	40
3.4.4 Pruebas con CNN entrenada.....	41
3.5 Acondicionamiento de datos de salida de CNN.....	42
CAPÍTULO 4: DISEÑO DE CONTROLADOR.....	43
4.1 Marco teórico de controladores difusos para el control de robots móviles.....	43
4.1.1 Teoría Clásica de conjuntos.....	43
4.1.2 Lógica difusa.....	44

4.1.3 Estructura y aplicaciones de los controladores difusos.....	45
4.2 Diseño de Controlador de Velocidad y Orientación del Robot (CVOR).....	47
4.2.1 Tabla de reglas.....	49
4.3 Implementación de Lógica Difusa.....	50
4.4 Integración de SDF y CVOR.....	52
CAPÍTULO 5: PRUEBAS Y RENDIMIENTO DE SISTEMA.....	56
5.1 Rendimiento del Sistema de Detección de Fallas.....	56
5.2 Resultados del sistema en general empleando vídeo de inspección.....	59
CONCLUSIONES.....	63
RECOMENDACIONES.....	64
BIBLIOGRAFÍA.....	65



ÍNDICE DE TABLAS

3.1. Tipos de anomalías a identificar	26
3.2. Funciones de activación.....	28
3.3: Comparación entre clasificadores.....	32
3.4 Comparación de precisión entre distintas versiones de YOLO	33
3.5. Resolución de imágenes recolectadas por las cámaras del robot de inspección.....	34
3.6. Distribución de dataset para entrenamiento, validación y pruebas de CNN.....	39
3.7.Principales especificaciones de la tarjeta Jetson Nano.....	40
3.8. Entrenamiento preliminar de red con clasificador Yolo v5 bajo diferentes configuraciones	41
3.9 Entrenamiento de red con clasificador Yolo v5.....	41
4.1. Operaciones de conjuntos difusos.....	45
4.2. Base de reglas para clasificación de fallas dentro de tuberías.....	46
4.3 Tabla de Reglas a emplear en el controlador neuro-difuso	50
5.1. Resultados obtenidos del Sistema de Detección de Fallas entrenando el clasificador bajo diferentes configuraciones	56
5.2. Rendimiento del clasificador luego del reentrenamiento de la CNN.....	58



ÍNDICE DE FIGURAS

1.1 Porcentaje de agua no facturada durante los años 2015-2019.....	3
1.2 Monitoreo de integridad de tubería usando un sistema SCADA.....	4
1.3 Detección de fuga por medio del sonido.....	4
1.4 Sistema de detección acústica experimental.....	5
1.5. Izquierda: Información obtenida del micrófono de referencia. Derecha: Información obtenida del micrófono de salida.....	5
1.6 Identificación de anomalías internas de un ducto mediante ultrasonido.....	6
1.7. Inspección visual de tuberías empleando cámaras.....	7
1.8. Cámara para inspección de tuberías Robuscam40	8
1.9 Robot para inspección de ductos DT340X.....	8
1.10. Prolaser montado en un robot de inspección.....	9
1.11 Robot móvil terrestre de inspección interna de tuberías.....	9
1.12 Robot de inspección de tuberías desarrollado en el GIT-PUCP.....	10
2.1. Izquierda: Representación en el plano de vehículo diferencial,Derecha: Configuración skid steer de 2 ruedas por lado.	12
2.2 Representación de los ICR de un robot skid steer en el plano.....	13
2.3. DLC de robot diferencial para modelamiento dinámico newtoniano.....	15
2.4 Modelo dinámico del robot en Simulink.....	17
2.5 Modelo cinemático de las ruedas del robot.....	18
2.6 Modelo de motores del robot en Simulink.....	19
2.7 Obtención de posición y orientación del robot en el plano xy.....	19
2.8 Modelo en Simulink de robot skid steer para su posterior control.....	20
2.9 Configuración de parámetros del robot.....	20
2.10 Comportamiento del robot a 18VDC, la velocidad es de 2.5 m/s.....	21
2.11 Comportamiento del robot al girar sobre su centro de masa, la velocidad angular es de 6.4 rad/s.	21
2.12 Comportamiento del robot al girar sobre su centro de masa, la velocidad angular es de 6.4 rad/s.....	22
2.13 Estructura del sistema de detección autónoma de fallas planteado.....	24
3.1 Izquierda: Cámara trasera del robot. Derecha: Cámara PTZ y de profundidad ubicados en la parte delantera del robot.....	25
3.2 Imágenes de interior de tubería de HDPE captadas empleando robots.....	26
3.3 Estructura base de red neuronal artificial.....	27
3.4 Curva que define la red neuronal empleando: a) entrenamiento online, b) entrenamiento batch	29
3.5 Estructura de CNN para reconocimiento de números.....	30
3.6 Estructura de CNN para reconocimiento del nivel de corrosión en tuberías de metal.....	31
3.7 Estructura del clasificador YOLO v5.....	33
3.8 Estructura para la extracción de características de las imágenes.....	34
3.9 Convolución aplicada a imagen en escala de grises empleando un kernel de 3 x 3.....	35
3.10 Ejemplo de zero padding aplicado al borde de la matriz de salida.....	36
3.11 Gráfica de función de activación ReLu.....	36
3.12 Obtención de mapa de detección de características.....	36
3.13 Recolección de características y empleo de SPP.....	37
3.14 Submuestreo empleando Max-Pooling con un filtro de 2 x 2.....	38
3.15 Obtención de bounding box de imagen en base al vector de predicción.....	38
3.16 Izquierda: Imagen original. Derecha: Imagen con anomalía detectada por la CNN.....	39
3.17 Reconocimiento automático de anomalías por parte de la CNN.....	42
3.18. Cálculo del centro de un bounding box.....	42
4.1. Representación clásica de dos subconjuntos con elementos en común.....	43
4.2 Funciones de pertenencia difusas más comunes: a) triangular, b) trapezoidal, c) gaussiana y d) sigmoial.....	44
4.3 Conjunto difuso que describe la velocidad.....	45
4.4 Estructura de controlador difuso.....	46
4.5 Diagrama de flujo para detección de fallas en tuberías empleando cámaras CCTV.....	47
4.6 Extracción de características de interior de tubería empleando un sistema difuso	47
4.7 Bloque del controlador a implementar.....	48

4.8 Funciones de pertenencia del error de orientación en grados (°).....	48
4.9 Funciones de pertenencia del error de velocidad en m/s.....	49
4.10 Funciones de pertenencia para el voltaje de los motores del robot.	49
4.11. Implementación de lógica difusa en Matlab.....	51
4.12 Gráficos de superficie para cada salida en base al valor de las entradas. Izquierda: Voltaje suministrado a motores izquierdos. Derecha: Voltaje suministrado a motores derechos.....	51
4.13 Pruebas de validación de controlador diseñado.....	52
4.14 Acondicionamiento de salidas de la CNN para obtener la velocidad y orientación deseada del robot.....	52
4.15 Integración de Sistema de Detección de Fallas y Controlador de Velocidad y Orientación del robot.....	53
4.16 Pruebas de integración del SDF con el CVOR.	53
4.17 Diagrama empleado para la simulación del comportamiento autónomo del robot empleando el controlador diseñado.....	55
5.1. Evaluación de la exactitud de las predicciones de los clasificadores entrenados.....	57
5.2 Comparación de consumo de GPU en cada clasificador entrenado.....	57
5.3 Evaluación de la mejora de la red Train_2 al entrenarla por un tiempo mayor.....	58
5.4 Evaluación del consumo de recursos del GPU durante el entrenamiento.....	58
5.5 Resultados obtenidos por el SDF.	59
5.6 Secuencia de reconocimiento automático de objeto extraño (tubería de PVC).	60
5.7 Secuencia de reconocimiento de agujero dentro de tubería.....	60
5.8 Valores de probabilidad y posición de anomalías reconocidas a lo largo del video.....	61
5.9 Variación de las salidas del controlador durante identificación de fallas.....	62



CAPÍTULO 1: INTRODUCCIÓN Y PROBLEMÁTICA

1.1 Introducción

Uno de los grandes desafíos a nivel global es la adecuada distribución del servicio de agua potable, alcantarillado y saneamiento a las familias pertenecientes a una población, debido a factores demográficos como el rápido crecimiento de las poblaciones, y factores técnicos como falta de monitoreo y mantenimiento de las tuberías, ineficiencias en los sistemas de distribución de agua, etc. Nuestro país no es ajeno a este problema, por el contrario, este problema se ve agravado por el hurto de agua potable causado por las conexiones clandestinas. En el año 2017 el Servicio de Agua Potable y Alcantarillado de Lima (Sedapal) perdió 28% de su facturación anual en Lima y Callao debido a tuberías rotas y conexiones clandestinas, lo cual equivaldría a casi 500 millones de soles.

En el país no se cuenta con suficientes herramientas tecnológicas que permitan evaluar el estado de la red de tuberías de agua y desagüe, especialmente en las que presentan mayor antigüedad, por lo que no es posible identificar fallas ni evitar la ocurrencia de desastres como inundaciones y desabastecimiento de agua potable. A nivel mundial se han desarrollado robots para realizar este tipo de tareas, los cuales cuentan con diseños complejos como los robot tipo oruga (Yamashita, A. y otros, 2011), (Streich, H. & Adria O., 2004) o tipo serpiente (Brunete, A. y otros, 2007) que por su forma y estructura flexible son capaces de desplazarse en tuberías de diámetro pequeño y formas variables. Se ha diseñado un robot que emplea actuadores inflables (Adams, W. y otros, 2018) permitiendo adaptar el tamaño del robot dependiendo del tamaño de la tubería a inspeccionar. Por otro lado se tienen los robots con diseños más convencionales que consisten en plataformas robóticas móviles con ruedas en las que se montan dispositivos como sensores de presión (Dvajasvie, G. y otros, 2018), cámaras (Hu, Y., Song Z. & Zhu, J., 2012), herramientas de limpieza (Saenz, J. y otros, 2010) o sensores láser (Le, D. V. K. y otros, 2020).

Los tipos de controladores empleados para el desplazamiento de este tipo de robots son los controladores PID, control LQR, controladores difusos (Mehrandezh, M. y otros, 2008), etc cuyo correcto funcionamiento depende de los sensores acoplados al robot. Para la automatización del desplazamiento del robot dentro de tuberías

aparte de los controladores mencionados, se usan métodos variados para estimar su posición empleando procesamiento de nube de puntos, fusión de datos (Murtra, A. C., & Tur, J. M. M., 2013) o procesamiento de imágenes. Existen casos donde se han empleado métodos de aprendizaje como Q-learning (Pinto, D. S. S., & da Silva, K. R. G., 2016) o redes neuronales (Jayakumar, D., Jaganath, R., & Selvarasu, R., 2016) para mejorar la identificación de defectos dentro de tubería, para ello se suministran datos de entrenamiento que permite que el controlador ajuste sus parámetros de reconocimiento para una mayor precisión al momento de identificar anomalías.

Se propone el desarrollo de un controlador neuro-difuso para realizar el sensado autónomo de fallas al inspeccionar el interior de tuberías matrices de agua empleando un robot móvil. El controlador a diseñar permitirá reconocer anomalías dentro de las tuberías empleando redes neuro-difusas. De manera que será posible detectar e identificar fallas en las tuberías de agua tales como: obstrucciones, fisuras y fugas debido a roturas o perforaciones.

1.2 Problemática

Evaluar el estado de la red de tuberías de agua y desagüe es una tarea esencial para identificar fallas y evitar desastres con el paso del tiempo. Debido a que no se cuentan con herramientas tecnológicas de manera local que permitan realizar monitoreos de manera frecuente y con mayor precisión que los métodos tradicionales, los trabajos de mantenimiento de tuberías no se realizan de manera efectiva. Esta situación se ve reflejada en la inundaciones y desabastecimientos causados por roturas de tuberías matrices de agua que a su vez amenazan la salud e integridad física de la población.

Según la Superintendencia Nacional de Servicios de Saneamiento, la pérdida de agua en los prestadores de servicios de saneamiento reflejados en porcentaje de Agua No Facturada (ANF) es mayor del 28% (SUNASS, 2020, págs. 25-28), es decir, al menos casi una tercera parte del agua suministrada a lo largo del país se pierde. En la Figura 1.1 se muestra el porcentaje de agua no facturada por año entre los años 2015-2019 en pequeñas, medianas y grandes empresas prestadoras de servicios de saneamiento. Se evidencia que el problema de pérdida de agua no presenta disminución con el paso del tiempo. Es importante mencionar que las causas de la existencia de agua no facturada son las fugas, filtraciones y conexiones

clandestinas. Estas situaciones podrían ser diagnosticadas efectivamente mediante monitoreos frecuentes.

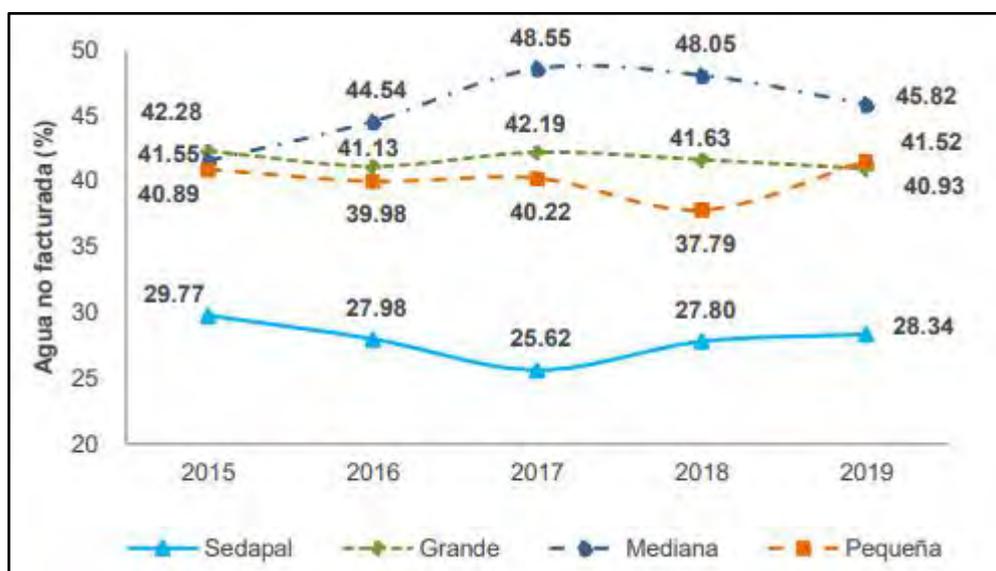


Figura 1. 1 Porcentaje de agua no facturada durante los años 2015-2019.
(SUNASS, 2020)

1.3 Métodos tradicionales de detección de fallas en tuberías

A partir de sistemas de Supervisión, Control y Adquisición de Datos (SCADA) y sensores físicos y virtuales ubicados en un cierto número de puntos a lo largo de la tubería es posible registrar presiones y flujos (Torres, L. y otros, 2014), como se muestra en la Figura 1.2. En base a la información obtenida y el correcto diseño de los sensores virtuales a partir de algoritmos de estimación e identificación es posible localizar fugas y obstrucciones en el cuerpo del ducto. Esto se logra a partir de un análisis armónico de la presión transitoria del fluido (agua) medida por los sensores. Además, con la respuesta frecuencial obtenida y si se tienen modelos analíticos precisos del fluido es posible obtener información de fallas en sistemas complejos usando herramientas de control automático (González, O., Verde, C., & Torres, L., 2017).

La detección de pequeñas fallas dependerá de la sensibilidad de los sensores físicos implementados, además la inspección se dificulta si es que las tuberías se encuentran bajo tierra o agua. A pesar de ser un sistema de detección robusto capaz de detectar, aislar y reconstruir diferentes tipos de fugas, el costo de implementación

aumenta considerablemente dependiendo de la longitud de las tuberías a monitorear y a la sensibilidad de los sensores a emplear.

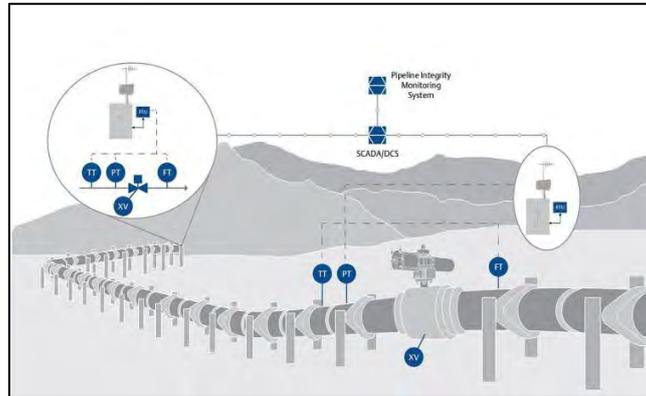


Figura 1.2 Monitoreo de integridad de tubería usando un sistema SCADA.
(Emesent, 2020)

Detección acústica:

Es uno de los métodos más usados para la detección de fugas en tuberías de agua debido a su bajo costo y rapidez de inspección. Generalmente se emplean geófonos o micrófonos de tierra, los cuales detectan y amplifican el ruido creado por fugas dentro de las tuberías. Este procedimiento consiste en desplazar el sensor por encima de donde se encuentra enterrada la tubería e ir analizando la intensidad del sonido. La fuga se encuentra donde el sonido sea mayor como se muestra en la Figura 1.3.

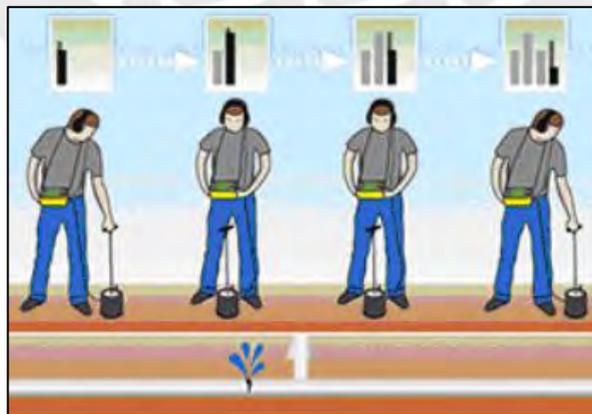


Figura 1.3 Detección de fuga por medio del sonido.
(Oxon Leak Detection, 2019)

El resultado dependerá de la sensibilidad y frecuencia de operación del sensor así como la experiencia del operador. Existen ciertos factores que pueden disminuir la efectividad de este método como la longitud de la tubería, la interferencia de ruido externo a la tubería, la presión del agua, el material y diámetro de la tubería, la profundidad a la que está enterrada, etc.

Entre los métodos de detección acústica en investigación se tiene un sistema que consiste en un parlante, un amplificador de audio y dos micrófonos para la adquisición de datos (Khan, M. S., 2017) como se muestra en la Figura 1.4. El micrófono de referencia mide la intensidad del sonido que sale del parlante mientras que el micrófono de salida evalúa la intensidad del sonido luego de desplazarse una distancia z . Comparando la información obtenida por ambos parlantes se puede hallar la atenuación del sonido, el cual es mayor si es que existe una grieta ya que el sonido generado por el micrófono escapa por ese medio. En la Figura 1.5 se muestran los resultados obtenidos por el micrófono de referencia y el de salida dentro de una tubería rota de PVC, se puede observar la atenuación del sonido.

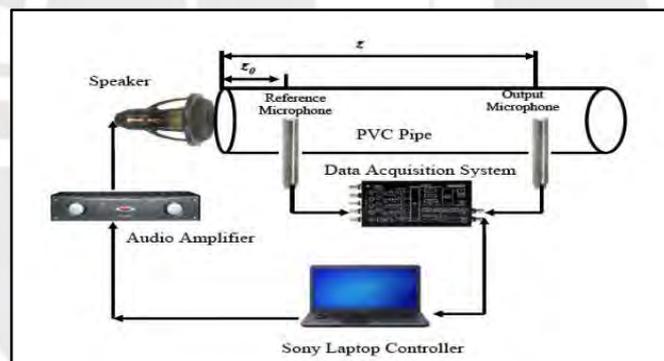


Figura 1.4 Sistema de detección acústica experimental.

(Khan, M. S., 2017)

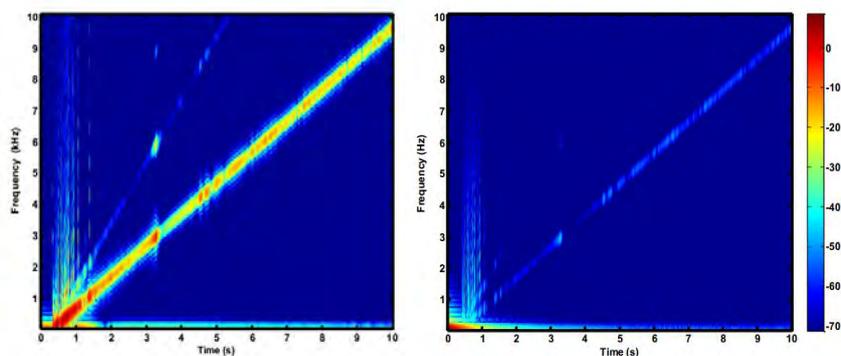


Figura 1.5. Izquierda: Información obtenida del micrófono de referencia. Derecha: Información obtenida del micrófono de salida. (Khan, M. S., 2017)

Existen métodos donde se localizan anomalías dentro del ducto mediante la emisión y detección de ultrasonido (0.3 a 1.3 MHz). En la Figura 1.6 se muestra el resultado de una inspección donde se estima el espesor de las paredes internas de un ducto de 36 pulgadas de diámetro y mediante la variación del sonido se estiman roturas y corrosión (Norli, P. y otros, 2018).

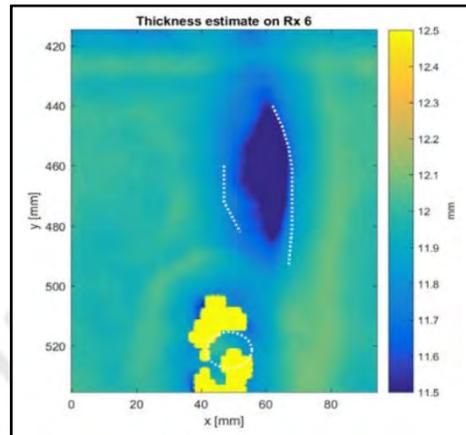


Figura 1.6 Identificación de anomalías internas de un ducto mediante ultrasonido.
(Norli, P. y otros, 2018).

Inspección interna de tuberías:

A diferencia de los métodos anteriores, la inspección interna permite encontrar un mayor rango de anomalías como deformación de las paredes internas de la tubería, corrosión, grietas superficiales, etc. Existen diferentes maneras de realizar inspecciones internas en tuberías de agua, las cuales se diferencian por el sensor empleado y el resultado obtenido. A continuación se detallan los métodos más usados:

Inspección visual interna: Este método consiste en introducir cámaras conectadas por cable a una computadora o pantalla HMI. El sistema también incluye luces LED para iluminar el entorno dentro de la tubería. La longitud del cable determinará la distancia máxima que puede inspeccionar la cámara, además para que la cámara se desplace dentro de la tubería es necesario empujar el cable a medida que se va introduciendo en la tubería. En caso de existir curvas, algunos mecanismos permiten realizar giros de 45°, como se muestra en la Figura 1.7, y en algunos casos de hasta 90°.



Figura 1.7. Inspección visual de tuberías empleando cámaras. (LightFoot Mechanical, 2019)

Este sistema es usado en áreas reducidas, además el tamaño compacto de las cámaras permite realizar inspecciones en tuberías delgadas. Mediante algoritmos de procesamiento de imágenes es posible identificar anomalías y calcular su tamaño y ubicación. En la Figura 1.8 se muestra el modelo de cámara Robuscam40 de la compañía Pro-loc para inspecciones en tuberías de hasta 23 mm de diámetro.



Figura 1.8. Cámara para inspección de tuberías Robuscam40 (Pro-loc, 2016)

Inspección interna mediante magnetismo:

Entre los métodos más usados se encuentra el empleo de sensores PEC (Pulsos de Corrientes Parásitas), los cuales utilizan corriente eléctrica alterna sinusoidal de una frecuencia particular para excitar una parte de la tubería. Esta técnica utiliza un voltaje de función escalonada la cual se realiza a diferentes frecuencias permitiendo obtener una respuesta electromagnética a diferente frecuencias que se pueden medir a la vez. Si las mediciones de la intensidad de la respuesta se realizan en el dominio del tiempo, las señales producidas por anomalías cerca de la bobina aparecerán primero. Se implementó un Proceso de Gauss basado en técnicas de machine learning para interpretar las respuesta electromagnéticas luego de usar pulsos de corrientes parásitas en una tubería ferromagnética (Ulapane, N. y otros, 2014). Es posible usar el principio de corrientes parásitas para complementar las imágenes obtenidas luego de una inspección interna de tuberías (Vidal-Calleja, T. y otros, 2014). Las características encontradas por el sensor PEC se asocian a las imágenes obtenidas mediante un algoritmo de odometría y las distancias dentro de

las tuberías en las que se tomaron los datos por parte de ambos sensores. De esta manera se reduce el número de falsos positivos permitiendo realizar un diagnóstico más preciso. La principal desventaja de este tipo de inspección es que solo se pueden realizar en tuberías de material ferromagnético.

Inspección interna empleando robots:

Es el método de inspección más completo y el que tiene mayor número de investigaciones comparándolo a los mencionados anteriormente debido a que, dependiendo de la complejidad del robot y de los sensores y actuadores acoplados, es posible encontrar casi cualquier anomalía dentro del ducto logrando un diagnóstico más detallado. Su capacidad de configurar el tamaño de sus ruedas y la posición de la cámara le permite realizar inspecciones de tuberías desde a 8" a 36". Los robots de inspección de tuberías generalmente usan una cámara PTZ y un sistema de iluminación basado en luces LED. Su diseño modular permite añadirles sensores especializados como: cámaras de profundidad, escáneres láser 2D y 3D, perfiladores láser, sondas acústicas, etc.

Empleando algoritmos de procesamiento de imágenes en los datos obtenidos por la cámara es posible localizar e identificar anomalías del interior de las tuberías. Uno de los modelos de robots más completos del mercado es el DT340X de la compañía canadiense Deep Trekker el cual se muestra en la Figura 1.9. Para complementar la inspección realizada por el robot se emplea un cabrestante de 400 metros de cable con contador de distancias integrado. Además de un controlador con pantalla donde se muestra el video del interior de la tubería en tiempo real y la distancia recorrida.



Figura 1.9 Robot para inspección de ductos DT340X. (Deep Trekker, 2018)

En caso se requiera conocer si existen deformaciones en las paredes internas de la tubería se debe emplear alguno de los sensores láser ya mencionados. En la Figura 1.10 se presenta el perfilador láser ProLaser de la compañía británica Mini-Cam que

permite medir con una resolución de hasta 0.5 mm. Con esta información es posible hacer una reconstrucción en 2D y 3D del interior de la tubería y calcular el tamaño de las deformaciones y del espesor de la tubería para conocer su nivel de desgaste. Se justifica entonces el uso de un robot para la inspección de tuberías de agua.

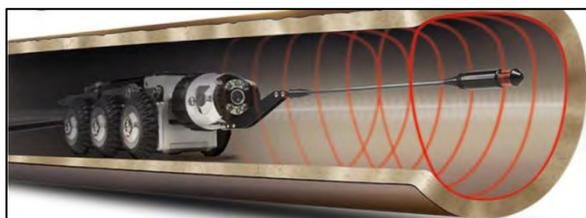


Figura 1.10. Prolaser montado en un robot de inspección. (Minicam, 2019)

1.4 Robots móviles terrestres para inspección de tuberías.

Los robots móviles se desarrollaron debido a la necesidad de extender el alcance de operación de la robótica en el medio terrestre además de incrementar la autonomía limitando en todo lo posible la intervención humana (Baturone, A. O., 2005). De acuerdo al grado de autonomía, los robots en general pueden clasificarse en tres tipos: teleoperados, semi autónomos, que incluyen a los de funcionamiento repetitivo, y autónomos. En la Figura 1.11 se muestra un robot móvil comercial de la compañía Clean-co Systems que realiza inspecciones del estado interno de tuberías. Posee una estructura compacta y alargada que le permite entrar a tuberías de diámetro reducido. Además este tipo de robots cuenta con un alto grado de protección IP ante polvo y agua.



Figura 1.11 Robot móvil terrestre de inspección interna de tuberías. (Clean-co Systems, 2019)

Los robots de inspección de tuberías se caracterizan porque se comunican con el operador ubicado en el exterior empleando un cable umbilical el cual sirve también como medio de alimentación eléctrica. La longitud de este cable define la distancia

máxima de trabajo del robot además de la potencia que deben suministrar los motores del mismo.

Para el desarrollo de la presente tesis se emplea un robot de investigación implementado por el Grupo de Innovación Tecnológica (GIT-PUCP) que se muestra en la Figura 1.12. Este robot presenta un configuración skid steer de 2 ruedas fijas por lado, tiene la capacidad de desplazarse 400 metros dentro de tuberías de un diámetro interno no menor a 60 cm. Su sistema de sensado está compuesto por una cámara PTZ (pan-tilt-zoom) y una cámara de profundidad ubicados en la parte frontal del robot mientras que una cámara web y un RPLiDAR se montan en la parte trasera. Los datos a procesar recolectados por los sensores mencionados serán de tipo imagen y nube de puntos, de manera adicional se cuenta con la información de cuanto se ha desplazado el robot mediante un encoder que mide cuánto cable ha sido desenrollado. Tanto el desplazamiento del robot como la identificación de anomalías dentro de la tubería se realizan de manera teleoperada por lo que se busca emplear los sensores que posee el robot para automatizar el proceso de inspección.



Figura 1.12 Robot de inspección de tuberías desarrollado en el GIT-PUCP.

1.5 Objetivo de la tesis

1.5.1 Objetivo General

El objetivo general de la tesis es diseñar un controlador inteligente basado en redes neuro-difusas que permita el reconocimiento automático de anomalías encontradas dentro de tuberías matrices de agua empleando un robot móvil.

1.5.2 Objetivo específicos

- Estudiar el estado del arte en metodologías de diagnóstico de fallas utilizando robots móviles para la recolección de datos.
- Definir el modelo matemático del robot a emplear según el tipo de configuración que posee.
- Diseñar un controlador basado en redes neuro-difusas para la identificación autónoma de las anomalías presentes en las paredes internas de tuberías matrices de agua.
- Realizar el entrenamiento de la red neuro-difusa diseñada buscando el máximo índice de éxito en el reconocimiento de anomalías. Para ello se emplean base de datos en línea como ImageNet o en su defecto, se recolectarán datos manualmente.
- Simular el desempeño del sistema propuesto empleando python y Matlab.
- Implementar el Sistema de Detección de Fallas (SDF) y el Controlador de Velocidad y Orientación del Robot (CVOR), los cuales trabajan en conjunto para el reconocimiento automático de fallas.

CAPÍTULO 2: MODELADO DE ROBOT MÓVIL

2.1 Modelo cinemático del robot móvil

El robot móvil a emplear para evaluar el funcionamiento del controlador neuro-difuso a implementar tendrá una configuración skid steer de 4 ruedas fijas. Para simplificar la obtención de los modelos cinemáticos de robots se adoptan las siguientes consideraciones iniciales:

- El robot se mueve sobre una superficie plana a velocidades bajas.
- El robot no tiene partes flexibles, se comporta como un sólido rígido.
- La trayectoria del vehículo para un periodo suficientemente pequeño de tiempo con dirección invariable será el de un arco de circunferencia.

En la configuración Skid Steer se disponen de múltiples ruedas en cada lado del robot móvil que actúan de manera simultánea, es decir todas las ruedas de cada lado giran a la misma vez y con la misma velocidad. Su movimiento es similar a la configuración diferencial, aunque la gran diferencia radica en la presencia de deslizamiento cuando el robot gira. Por lo tanto es necesario emplear sensores inerciales para estimar los deslizamientos. Dentro de las ventajas se encuentran la gran maniobrabilidad y controlabilidad que presentan estos tipos de robots.

El espacio de trabajo en el caso de robots móviles que navegan en el plano sería un espacio de tres dimensiones para especificar la posición y el ángulo de orientación. En la Figura 2.1 se muestra la representación en el plano de un robot skid steer y su similitud con un robot diferencial de dos ruedas.

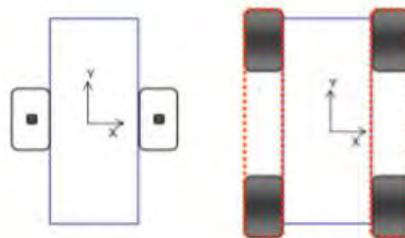


Figura 2.1. Izquierda: Representación en el plano de vehículo diferencial, Derecha: Configuración skid steer de 2 ruedas por lado.

Para el análisis cinemático de este tipo de configuración en específico se consideran las asunciones adicionales:

- El centro de masa y el centro geométrico del robot se encuentran localizados en el mismo punto.
- Todas las ruedas de cada lado poseen la misma velocidad.
- Todas las ruedas del robot siempre están en contacto con el suelo.

Los vehículos skid steer funcionan en base a dos entradas de control: V_l y V_r las cuales representan la velocidad lineal de las ruedas del lado izquierdo y derecho del robot. Entonces se busca una transformación fd donde obtiene la cinemática directa en el plano que cumpla lo siguiente:

$$\begin{pmatrix} v_x \\ v_y \\ \omega_z \end{pmatrix} = fd \begin{pmatrix} V_l \\ V_r \end{pmatrix} \dots (\text{Ecuación 1})$$

Donde:

$V = (V_l, V_r)$ es la velocidad de la rueda izquierda y derecha del robot respectivamente.

$v = (v_x, v_y)$ es la velocidad lineal del vehículo,

ω_z es la velocidad angular del robot.

En la Figura 2.2 se muestra el Centro de Rotación Inmediato (ICR_v) del robot (Mandow, A., Martinez, J. L. y otros, 2007), el cual se encuentra en la parte izquierda de la Figura indicando que en ese instante de tiempo el robot está girando hacia la izquierda con una velocidad angular ω_z . El ICR_v puede ser descompuesto en el Centro de Rotación Inmediato izquierdo (ICR_l) y derecho (ICR_r) cuyo componente y en el plano es el mismo además de la velocidad angular ω_z .

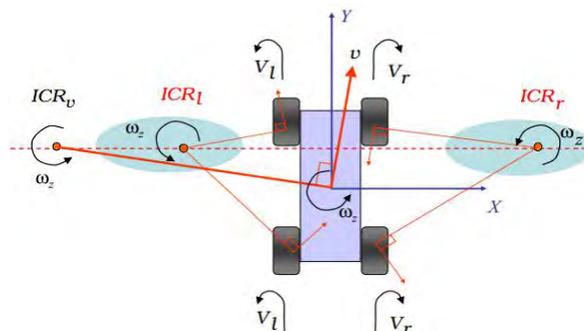


Figura 2.2. Representación de los ICR de un robot skid steer en el plano. (Mandow, A., Martinez, J. L. y otros, 2007)

En base a las relaciones geométricas de lo descrito se cumple:

$$x_{ICRv} = \frac{-u_y}{\omega_z} \dots \text{(Ecuación 2)}$$

$$x_{ICRl} = \frac{\alpha_l \cdot V_l - u_y}{\omega_z} \dots \text{(Ecuación 3)}$$

$$x_{ICRr} = \frac{\alpha_r \cdot V_r - u_y}{\omega_z} \dots \text{(Ecuación 4)}$$

$$y_{ICRv} = y_{ICRl} = y_{ICRr} = \frac{u_x}{\omega_z} \dots \text{(Ecuación 5)}$$

Donde:

$ICR_v = (X_{ICRv}, Y_{ICRv})$ es el centro de rotación del vehículo.

$ICR_l = (X_{ICRl}, Y_{ICRl})$ es el centro de rotación de las ruedas izquierdas,

$ICR_r = (X_{ICRr}, Y_{ICRr})$ es el centro de rotación de las ruedas derechas,

α_l = factor de corrección de ruedas de lado izquierdo,

α_r = factor de corrección de ruedas de lado derecho,

Dentro de los factores de corrección se consideran factores mecánicos de las ruedas como que tan infladas o desgastadas se encuentran.

De las ecuaciones 2 a 5, se obtiene la relación cinemática mostrada en la siguiente ecuación:

$$\begin{pmatrix} v_x \\ v_y \\ \omega_z \end{pmatrix} = A \cdot \begin{pmatrix} V_l \\ V_r \end{pmatrix} \dots \text{(Ecuación 6)}$$

Donde los elementos de la matriz A son:

$$A = \frac{1}{x_{ICRr} - x_{ICRl}} \cdot \begin{bmatrix} -y_{ICRv} \cdot \alpha_l & y_{ICRv} \cdot \alpha_r \\ x_{ICRr} \cdot \alpha_l & -x_{ICRl} \cdot \alpha_r \\ -\alpha_l & \alpha_r \end{bmatrix} \dots \text{(Ecuación 7)}$$

Asumiendo una simetría ideal del modelo cinemático del robot y de todas sus ruedas se cumple:

$$\alpha = \alpha_l = \alpha_r \dots \text{(Ecuación 8)}$$

$$x_{ICRv} = -x_{ICRl} = x_{ICRr} \dots \text{(Ecuación 9)}$$

Empleando las igualdades de las ecuaciones 8 y 9, las ecuación 7 quedaría de la siguiente manera:

$$A = \frac{\alpha}{2x_{ICR}} \cdot \begin{bmatrix} 0 & 0 \\ x_{ICR} & x_{ICR} \\ -1 & 1 \end{bmatrix} \dots(\text{Ecuación 10})$$

2.2 Modelo dinámico del robot móvil

Para obtener el modelo dinámico del robot móvil, se emplea el método de Newton-Euler para su cálculo. Se grafica el Diagrama de Cuerpo Libre (DLC) del sistema y se analizan las fuerzas que actúan sobre él como se muestra en la Figura 2.3. Se debe tener en cuenta la similitud descrita anteriormente entre la configuración diferencial y skid steer.

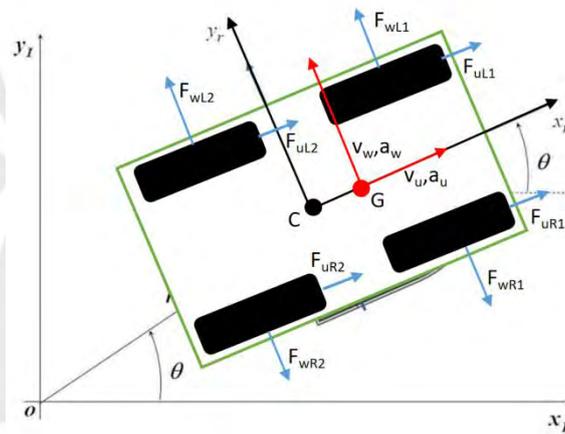


Figura 2.3. DLC de robot diferencial para modelamiento dinámico newtoniano.

Las ruedas y motores empleados en el robot son los mismos, cada rueda se conecta a un motor por transmisión directa, por lo tanto se cumple que :

$$F_{uL1} = F_{uL2} \dots (\text{Ecuación 11})$$

$$F_{uR1} = F_{uR2} \dots (\text{Ecuación 12})$$

$$F_{wL1} = F_{wL2} \dots (\text{Ecuación 13})$$

$$F_{wR1} = F_{wR2} \dots (\text{Ecuación 14})$$

$$F_{uL} = F_{uL1} + F_{uL2} \dots (\text{Ecuación 15})$$

$$F_{uR} = F_{uR1} + F_{uR2} \dots (\text{Ecuación 16})$$

Considerando la segunda ley de Newton para encontrar las relaciones entre las fuerzas, torques y aceleraciones se obtienen las siguientes ecuaciones dinámicas:

$$Ma_u = F_{uL} + F_{uR} \dots (\text{Ecuación 17})$$

$$Ma_w = F_{wL} - F_{wR} \dots \text{ (Ecuación 18)}$$

$$J\ddot{\theta} = (F_{uR} + F_{uL})L + (F_{wR} - F_{wL})d \dots \text{ (Ecuación 19)}$$

Donde:

M = masa del robot,

(a_u, a_w) = aceleración del centro de masa del robot,

(F_{uR}, F_{uL}) = fuerzas longitudinales ejercidas en las ruedas derecha e izquierda,

(F_{wR}, F_{wL}) = fuerzas laterales ejercidas en las ruedas derecha e izquierda,

J = momento de inercia de robot,

$\ddot{\theta}$ = aceleración angular de robot,

L = distancia entre ruedas,

d = distancia del centro de masa del robot con respecto al centro geométrico.

Sustituyendo los valores de aceleración en las ecuaciones 17, 18 y 19 se obtiene:

$$\dot{v}_u = v_w \dot{\theta} + \frac{F_{uL} + F_{uR}}{M} \dots \text{ (Ecuación 20)}$$

$$\dot{v}_w = -v_u \dot{\theta} + \frac{F_{wL} - F_{wR}}{M} \dots \text{ (Ecuación 21)}$$

$$\ddot{\theta} = \frac{L}{J} (F_{uR} - F_{uL}) + \frac{d}{J} (F_{wR} - F_{wL}) \dots \text{ (Ecuación 22)}$$

Teniendo en cuenta las restricciones de robots no-holonómicos se asume que no existe deslizamiento lateral por lo tanto:

$$v_w = d\dot{\theta} \dots \text{ (Ecuación 23)}$$

Sustituyendo la ecuación 23 en 20 y 21, y combinando 22 se obtiene:

$$\dot{v}_u = d\dot{\theta}^2 + \frac{(F_{uL} + F_{uR})}{M} \dots \text{ (Ecuación 24)}$$

$$\ddot{\theta} = \frac{L}{Md^2 + J} (F_{uR} - F_{uL}) - \frac{Mdv_u}{Md^2 + J} \dot{\theta} \dots \text{ (Ecuación 25)}$$

Se descomponen los valores de las fuerzas longitudinales en torques aplicados a cada rueda y las ecuaciones quedarían de la siguiente manera:

$$\dot{v}_u = d\dot{\theta}^2 + \frac{(\tau_L + \tau_R)}{MR} \dots \text{ (Ecuación 26)}$$

$$\ddot{\theta} = \frac{L}{R(Md^2 + J)} (\tau_R - \tau_L) - \frac{Mdv_u}{Md^2 + J} \dot{\theta} \dots \text{ (Ecuación 27)}$$

Donde:

R = radio de las ruedas del robot.

$(\tau_R; \tau_L)$ = torque aplicado a las ruedas derecha e izquierda respectivamente.

Las ecuaciones 26 y 27 servirán para modelar el comportamiento dinámico del robot.

2.3 Simulación de modelo dinámico de robot

Empleando las ecuaciones 26 y 27 se modela la dinámica del robot, donde la entrada será el torque aplicado a las dos ruedas del lado derecho y a las dos ruedas del lado izquierdo del robot. Al emplear motores y ruedas idénticas, el torque total por lado será la multiplicación del torque individual por el número de ruedas por lado. En la Figura 2.4 se muestra la simulación del donde las entradas son los torques aplicados a las ruedas del lado derecho (τ_R) y a las del lado izquierdo (τ_L). Las salidas son la velocidad angular y la velocidad lineal o de traslación del robot.

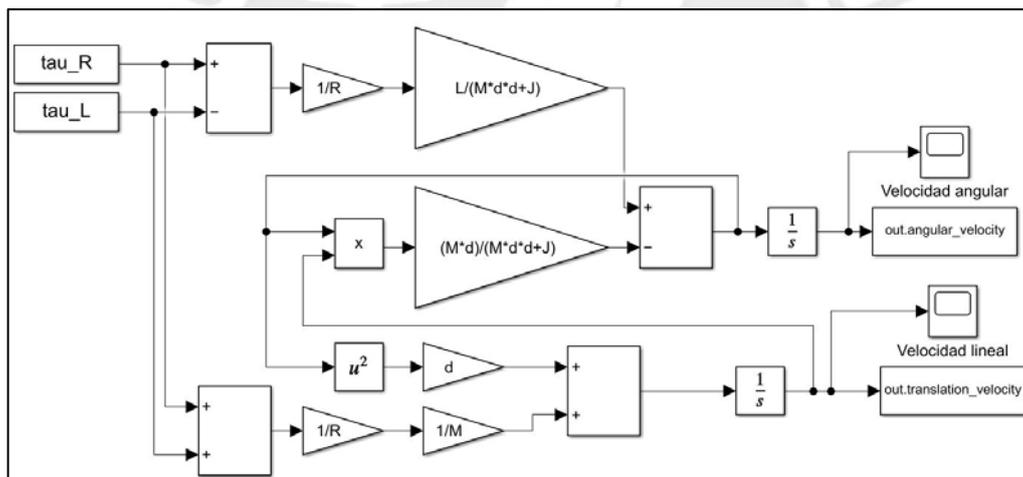


Figura 2.4 Modelo dinámico del robot en Simulink

Para modelar los motores que se emplean en el robot es necesario conocer en todo momento las velocidades angulares de las ruedas del lado derecho y lado izquierdo. Para ello es necesario agregar la cinemática de las ruedas en el modelo de Simulink como se muestra en la Figura 2.5 en base a las siguiente ecuaciones:

$$\dot{\phi}_R = \frac{v_u + L\dot{\theta}}{R} \dots \text{(Ecuación 28)}$$

$$\dot{\phi}_L = \frac{v_u - L\dot{\theta}}{R} \dots \text{(Ecuación 29)}$$

Donde:

$\dot{\phi}_R$ = Velocidad angular de la rueda derecha,

$\dot{\phi}_L$ = Velocidad angular de la rueda izquierda.

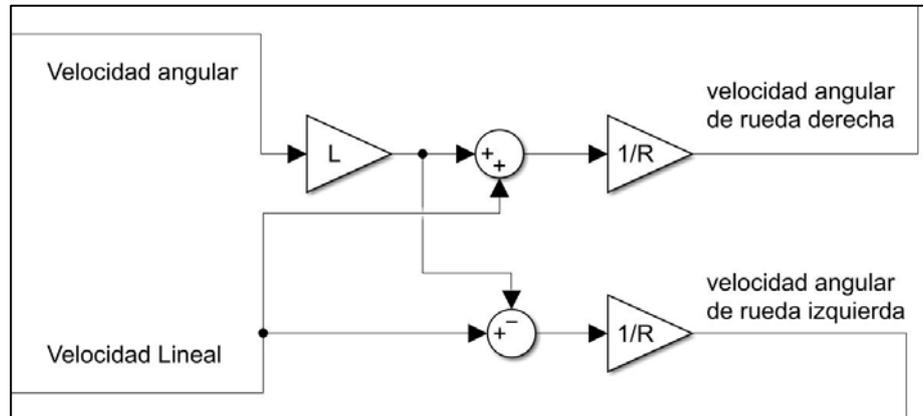


Figura 2.5 Modelo cinemático de las ruedas del robot.

Es importante resaltar que en este robot, se cumple que las velocidades angulares de las ruedas son iguales a las velocidades angulares de los motores ya que se encuentran conectados de manera directa ($\omega_m = \dot{\phi}$).

Para modelar los motores que forman parte del robot se emplean las siguientes ecuaciones:

$$i = (V - K_b * \omega_m) \frac{1}{R_a + L_a s} \dots \text{(Ecuación 30)}$$

$$\tau = N K_t i \dots \text{(Ecuación 31)}$$

Donde:

i = corriente del motor,

V = voltaje de entrada del motor,

K_b = constante electromotriz de motor,

K_t = constante de torque de motor,

ω_m = velocidad angular de giro del eje del motor.

R_a = resistencia de motor,

L_a = inductancia de motor,

N = reducción de motorreductor,

τ = torque generado por motor.

Como señales de entrada los motores reciben un valor de voltaje y la velocidad angular de la rueda a la cual se encuentran directamente conectada. Y como salida se obtiene el torque generado. En la Figura 2.6 se muestra el modelamiento en Simulink de los motores que emplea el robot. Se agrupan los motores del lado izquierdo y derecho del robot por lo que al torque de salida se le multiplica por el número de motores que componen al robot, en este caso se tienen dos motores por lado.

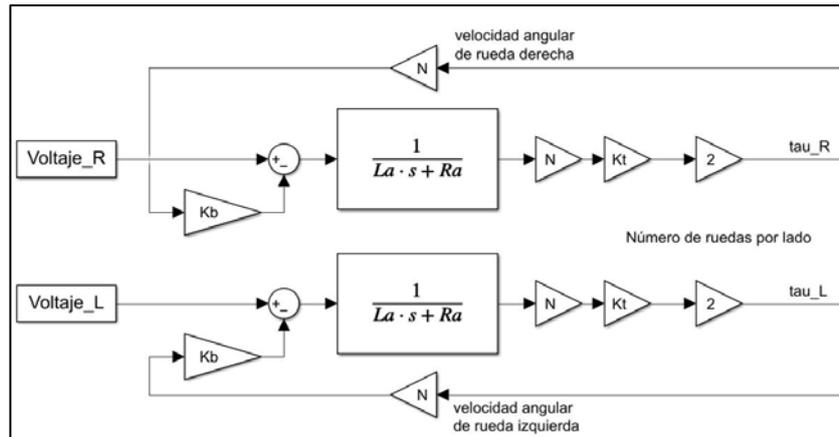


Figura 2.6 Modelo de motores del robot en Simulink.

Para analizar la respuesta del robot en cuanto a su desplazamiento en el plano x-y y su orientación con respecto al eje x se realizan las transformaciones mostradas en la Figura 2.7.

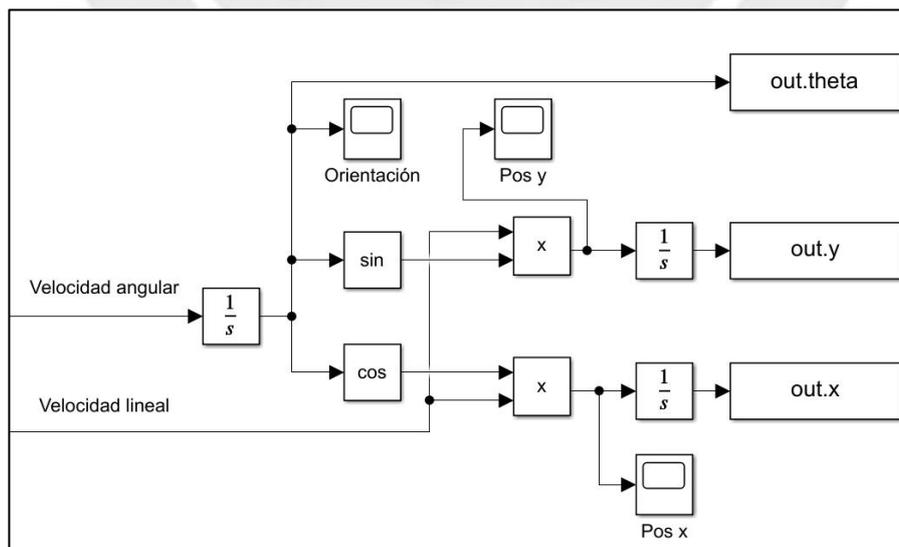


Figura 2.7 Obtención de posición y orientación del robot en el plano xy.

La Figura 2.8 muestra los sistemas mencionados agrupados, de manera que la entrada que permite el movimiento del robot es el voltaje aplicado a los motores del lado derecho y los del lado izquierdo. Las salidas serán la posición, velocidad y orientación del robot.

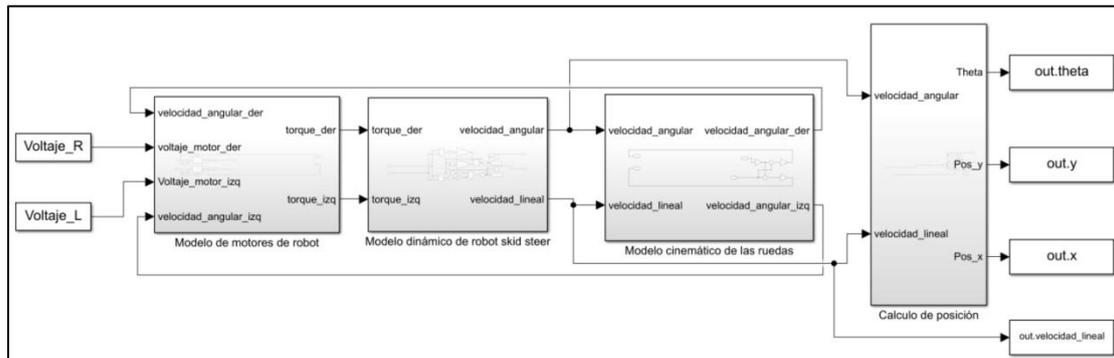


Figura 2.8 Modelo en Simulink de robot skid steer para su posterior control.

Empleando un script en Matlab se define el valor de los parámetros empleados en el modelo en base a las características reales del robot:

```

%% Parametros mecánicos de robot

%Parámetros de los motores
La=3; %H, inductancia
Ra=6; %Ohm, resistencia

Kb=0.0625; %V/rad/s constante electromotriz
N=40; %relación de transmisión de motorreductor
Kt=0.062; %Nm/Amp constante de torque

M = 82; % Masa del robot en Kg
J = (1/12)*M*((0.3^2)+(0.6^2)); % Momento de inercia del robot en kg*m^2
d = 0.1;% Distancia entre centro de masa y centro geométrico del robot en m.
R = 0.381; % Radio de la rueda del robot en m.
L = 0.4; %Distancia entre ruedas en m.

```

Figura 2.9 Configuración de parámetros del robot.

Se realizan pruebas ingresando el valor de voltaje para el motor de manera manual, se debe tener en cuenta que el motor opera hasta con 24 VDC.

En la Figura 2.10 se muestran los valores de desplazamiento en el plano x-y, la velocidad angular y velocidad lineal del robot para una entrada de 18VDC en cada uno de sus motores.

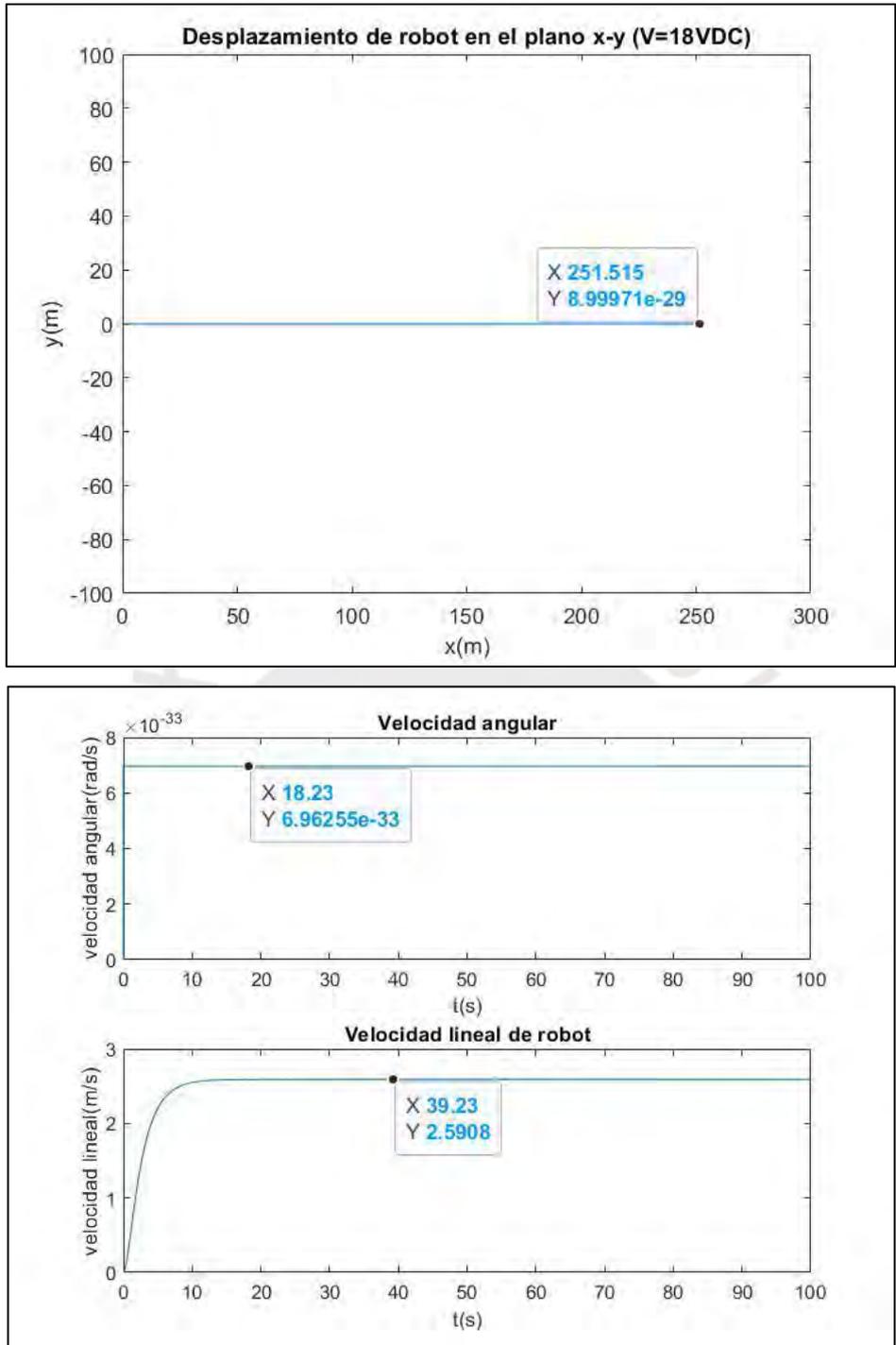


Figura 2.10 Comportamiento del robot a 18VDC, la velocidad es de 2.5 m/s.

Se realizan dos pruebas adicionales, la primera evalúa el comportamiento del robot al enviar valores de voltaje invertidos a los motores de la derecha e izquierda como se muestra en la Figura 2.11. En la última prueba se envía un voltaje de 17VDC a los motores de la derecha y 15 VDC a los motores de la izquierda, el resultado se muestra en la Figura 2.12.

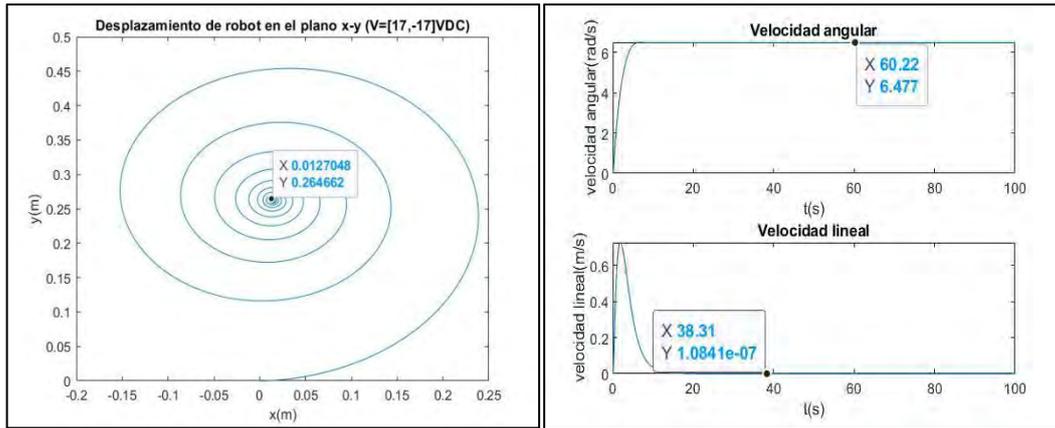


Figura 2.11 Comportamiento del robot al girar sobre su centro de masa, la velocidad angular es de 6.4 rad/s.

En la Figura 2.11 se demuestra el recorrido que seguirá el robot al girar sobre su centroide. Debido a que el centro de masa y el centro geométrico no se encuentran en el mismo punto, al intentar girar sobre su propio eje el robot describe un movimiento en espiral convergiendo en el punto (0.01,0.26). En cuanto a su velocidad angular, se mantiene en 6.48 rad/s y su velocidad lineal es de prácticamente 0 al encontrarse girando.

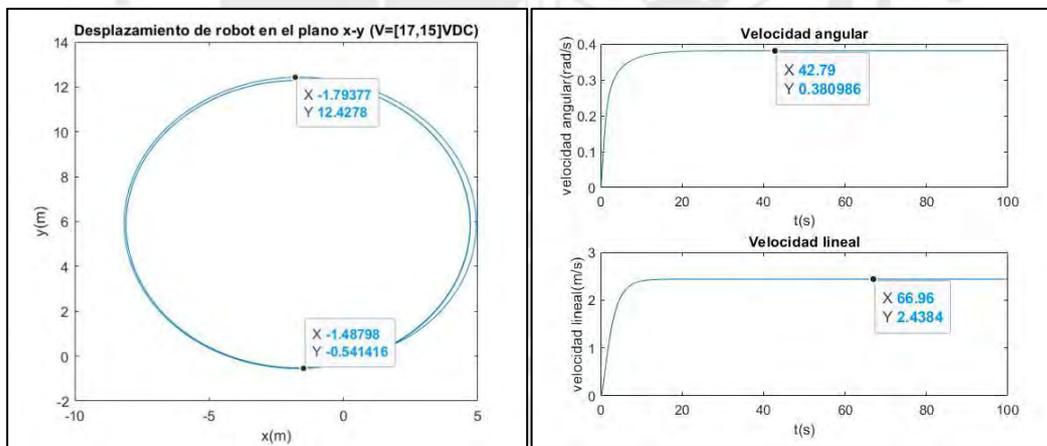


Figura 2.12 Comportamiento del robot al girar hacia la izquierda.

Al alimentar con un voltaje mayor los motores de la derecha del robot, estos giran con mayor rapidez permitiendo al robot girar con un radio de curvatura de casi 13 m, a una velocidad lineal de 2.4 m/s y angular de 0.38 rad/s. Si la diferencia de voltaje de los motores es mayor, el radio de curvatura disminuye al aumentar la velocidad angular del robot. Se concluye que el modelamiento realizado en Simulink va de acorde al comportamiento real esperado del robot.

2.4 Estructura de control planteada

En la Figura 2.13 se muestra la estructura de control planteada en base al estado del arte revisado, el modelo cinemático del robot y los sensores incluidos en el sistema. Las partes principales son el Sistema de Detección de Fallas (SDF) y el Controlador de Velocidad y Orientación del Robot (CVOR). De manera adicional se cuenta con un sistema de visión y un sistema de sensores.

El SDF consiste en una Red Neuronal Convolutiva (CNN) entrenada para la identificación de anomalías de manera automática mediante imágenes captadas por las múltiples cámaras del robot. El SDF detecta la presencia de tres tipos de anomalías: objetos extraños, grietas y perforaciones y sus posiciones dentro de la tubería. Se envía la información acerca de la probabilidad de existencia de anomalías (e) y su posición (p).

El CVOR se encarga de controlar la velocidad (v) y orientación (θ) mediante el voltaje que se le suministra a los motores del lado izquierdo y derecho del robot móvil dependiendo del valor de la probabilidad de existencia de una anomalía. En caso se identifique una anomalía se enviará la instrucción de disminuir la velocidad y modificar la orientación del robot variando el voltaje suministrado a los diversos motores dependiendo del tipo de anomalía, de manera que es posible analizarla con más detalle. El controlador será de tipo difuso y su funcionamiento será comparado con un controlador PID para evaluar su desempeño.

El sistema de visión está compuesto por tres tipos de cámaras montadas en la parte frontal y trasera del robot, todas las imágenes captadas son enviadas al SDF. El sistema de sensores está compuesto por una Unidad de Medición Inercial (IMU) para medir la orientación y velocidad del robot, un escáner láser tipo RPLiDAR que permite medir distancias respecto a las paredes de las tuberías y encoders de motor para el cálculo del desplazamiento del robot.

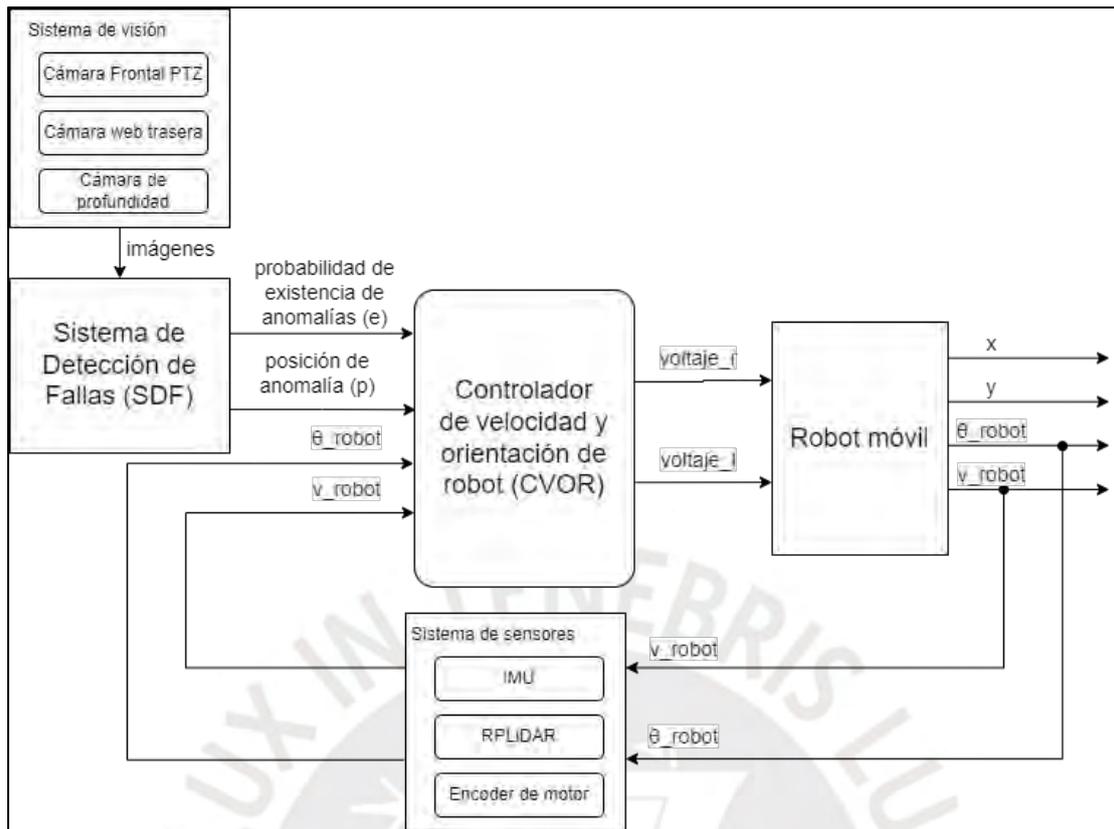


Figura 2.13. Estructura del sistema de detección autónoma de fallas planteado.

CAPÍTULO 3: DISEÑO DE SDF

3.1 Adquisición de datos

En el presente trabajo se empleará un robot móvil con configuración skid steer para la adquisición de los datos de entrada. Para ello se usará una cámara PTZ colocada en parte frontal del robot cuyo rango de visión permitirá observar una mayor área de la tubería, una cámara de profundidad ubicada también en la zona frontal del robot con potencial para reconstrucción 3D del interior del ducto, para el desarrollo del trabajo propuesto sólo se analizan las imágenes rgb obtenidas por este sensor. Por último, se emplea una cámara convencional en la parte trasera del robot para complementar la recolección de imágenes de las anomalías encontradas.

3.1.1 Recolección de imágenes

Las cámaras del robot móvil recolectan video en tiempo real del interior de la tubería inspeccionada por el robot, esta información es enviada a través de 600 m de cable hacia una estación base con un tiempo de retardo estimado de 1 segundo. La distribución de las cámaras se muestran en la Figura 3.1 mientras que en la Figura 3.2 se presenta un ejemplo de imágenes obtenidas con robots de similares características bajo las condiciones esperadas de trabajo.

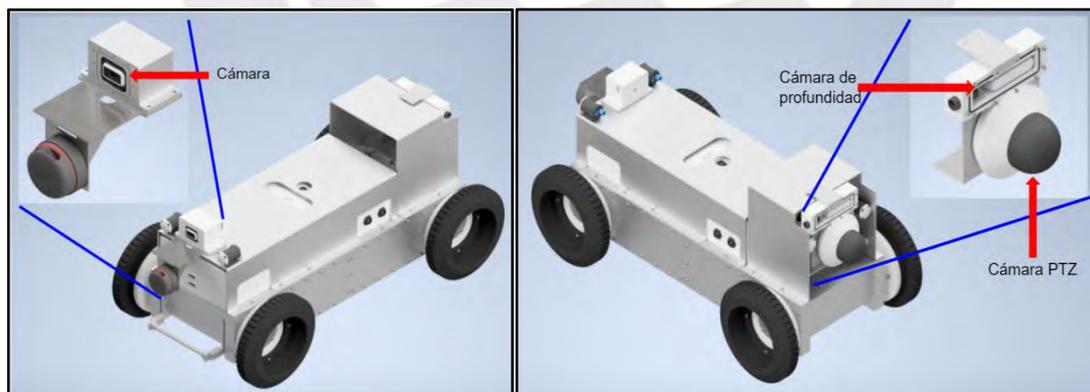


Figura 3.1. Izquierda: Cámara trasera del robot. Derecha: Cámara PTZ y de profundidad ubicados en la parte delantera del robot.



Figura 3.2. Imágenes de interior de tubería de HDPE captadas empleando robots.

3.1.2 Anomalías a identificar

El controlador a implementar será capaz de identificar 3 tipos de anomalías diferentes y su posición relativa en cada imagen procesada por la CNN. Las anomalías a evaluar son : i) agujeros, los cuales son producto de conexiones clandestinas o perforaciones erróneas causadas a las tuberías, ii) grietas ocasionadas producto de golpes externos o composición deficiente de la tubería, iii) objetos extraños ubicados dentro de las tuberías que terminan afectando el flujo de líquidos.

En la Tabla 3.1 se muestran los tipos de fallas a identificar y su respectiva etiqueta.

Tabla 3.1. Tipos de anomalías a identificar.

Tipo de anomalía	Posición	Etiqueta
Agujero	Centro	0
Agujero	Izquierda	1
Agujero	Derecha	2
Grieta	Centro	3
Grieta	Izquierda	4
Grieta	Derecha	5
Objeto extraño	Centro	6
Objeto extraño	Izquierda	7
Objeto extraño	Derecha	8
Sin anomalías	No aplica	9

La presencia o en su defecto, la ausencia de anomalías determinará la velocidad con la cual el robot se desplaza dentro de la tubería. Mientras que la posición de la anomalía servirá para controlar la orientación del robot mientras se desplaza. Si es

que las dimensiones de la tubería no permiten realizar giros, se tiene como alternativa el control de la posición de la cámara frontal PTZ del robot.

3.2 Marco teórico de sistemas neuronales para la detección automática de características

3.2.1 Redes neuronales

Los sistemas basados en redes neuronales artificiales son capaces de aprender diferentes patrones presentes en la información de entrada empleando métodos matemáticos algebraicos. La estructura de una red neuronal artificial se compone de: una capa de entrada, una o más capas ocultas y una capa de salida. Cada capa puede poseer múltiples neuronas con pesos asociados que determinarán el valor de la salida al procesarlos empleando una función de activación. En la Figura 3.3 se muestra la estructura básica de una red neuronal artificial.

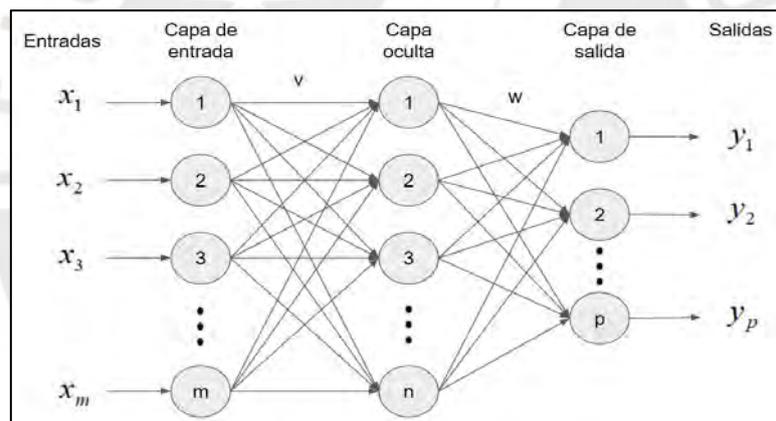


Figura 3.3. Estructura base de red neuronal artificial.

El número de neuronas en la capa de entrada depende de la cantidad de datos que ingresan a la red neuronal, además se cumple que el número de neuronas en la capa de salida no puede ser mayor al número de entradas. Por otro lado, la cantidad de neuronas en la capa oculta depende del tipo de operación que realiza la red. Los valores v y w representan el valor de los coeficientes de conexión entre neuronas cuyo valor se actualiza durante el periodo de entrenamiento de la red neuronal.

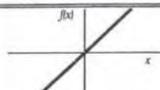
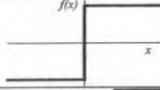
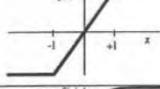
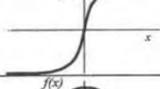
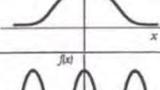
Todas las neuronas de la red contienen una función de activación, generalmente las neuronas de la capa de entrada y la de salida aplican una función de activación lineal a sus respectivas entradas. Las neuronas pertenecientes a las capas ocultas

poseen en su mayoría funciones no-lineales las cuales se eligen dependiendo del comportamiento de los datos de entrada.

3.2.2 Funciones de activación

Las neuronas que son parte de las capas ocultas poseen su propia función de activación, éstas reciben como entrada la sumatoria del producto entre el valor de las neuronas de entrada y su peso correspondiente. Y su salida estará basada en la función de activación la cual generalmente es de tipo sigmoideal, gaussiana, lineal, escalón, etc. En la Tabla 3.2 se presentan las características de las funciones de activación mencionadas.

Tabla 3.2. Funciones de activación.

	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq +l \\ +1, & \text{si } x > +l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A \text{sen}(\omega x + \varphi)$	$[-1, +1]$	

3.2.3 Entrenamiento de las redes neuronales

Para elegir el método de entrenamiento a emplear en la red neuronal artificial es necesario considerar sus parámetros, su función de optimización y la invarianza frente a transformaciones que le puedan ocurrir a los datos de entrada. El objetivo del proceso de entrenamiento es ajustar los valores de los coeficientes de conexión entre neuronas empleando alguna función de costo predeterminada.

La frecuencia con la que se actualizan estos valores define el modo de entrenamiento el cual puede ser online o batch. En el modo de entrenamiento online los valores de los coeficientes se ajustan después cada ejemplo de entrenamiento mientras que en el modo de entrenamiento batch la actualización ocurre después de

trabajar con todo el conjunto de datos o dataset. En la Figura 3.4 se muestra la curva que define la red neuronal empleando los modos de entrenamiento descritos.

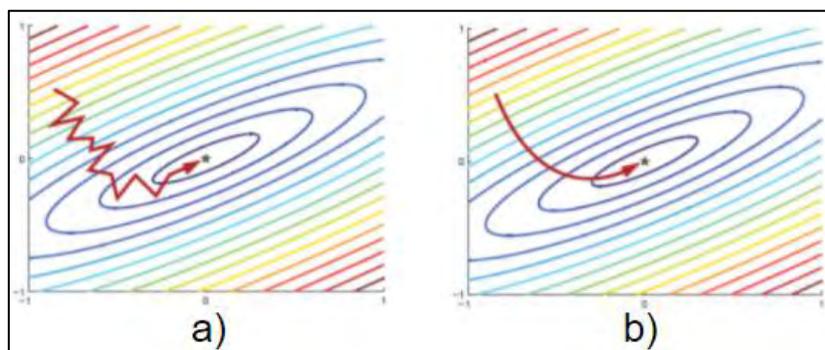


Figura 3.4. Curva que define la red neuronal empleando: a) entrenamiento online, b) entrenamiento batch

El método de entrenamiento online es más rápido, suele obtener mejores soluciones y facilita la adaptación al cambio. Por otro lado, el entrenamiento batch permite definir con claridad las condiciones de convergencia y análisis de dinámica del sistema. Cabe destacar que muchas técnicas de optimización funcionan exclusivamente con el entrenamiento batch.

3.2.4 Redes Neuronales Convolucionales

De los múltiples tipos de redes neuronales artificiales existentes, las Redes Neuronales Convolucionales (CNN) fueron diseñadas especialmente para el reconocimiento de características empleando imágenes. Las CNN imitan al sistema visual humano, asociando patrones en imágenes con objetos diferentes. Dependiendo de la complejidad de la CNN se pueden lograr aplicaciones como la identificación del estado de ánimo de personas, detección de células cancerígenas o el control autónomo de robots.

La información de entrada de las CNN son imágenes las cuales son representadas en matrices de píxeles. Esta información es procesada para hallar la Región De Interés de cada imagen (ROI), la cual es analizada y clasificada. El hallar la ROI antes de la clasificación es clave para reducir enormemente los costos computacionales ya que la red identifica qué porción de la imagen es la que debe analizar. La estructura general de las CNN está compuesta por la capa convolucional, la capa para las funciones de activación, la capa de agrupamiento (pooling) y la capa de salida (Mohareri, O., Dhaouadi, R., & Rad, A. B., 2012). En la

Figura 3.5 se muestra la estructura básica de una CNN para el reconocimiento de números.

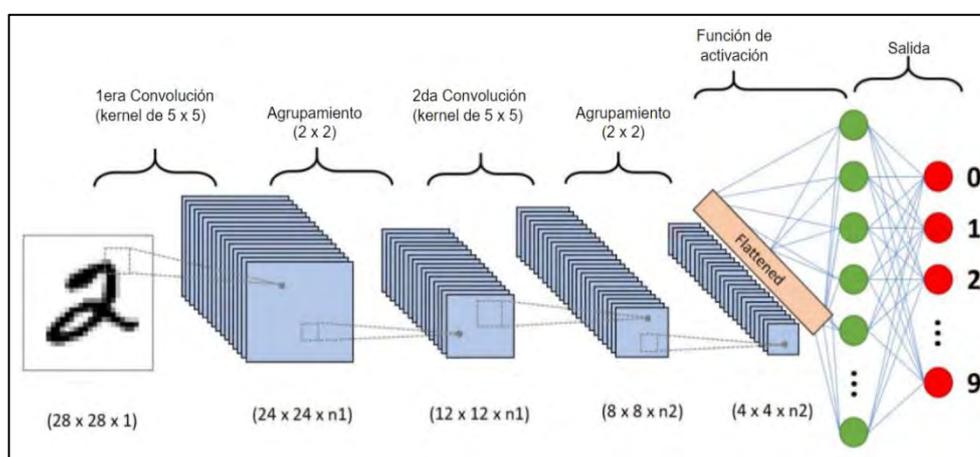


Figura 3.5 Estructura de CNN para reconocimiento de números.

Las capas convolucionales son el componente principal de las CNN ya que permiten extraer las características más importantes de las imágenes. El número de capas convolucionales dentro de una CNN representa el número de características totales a extraer, las cuales se obtienen mediante la operación de convolución de cada pixel de la imagen con un kernel. Dependiendo del tamaño del kernel empleado se reduce ligeramente la información de entrada como se observa en la Figura 3.5, en este caso se emplean dos kernels con dimensiones de 5 x 5.

Las capas de agrupamiento (pooling) tienen como objetivo reducir el costo computacional reduciendo el tamaño de la imagen a procesar en cada operación de convolución. En la capa de la función de activación se realiza la clasificación de la información, generalmente se emplea la función Unidad Lineal Rectificada (ReLU) donde los valores negativos se igualan a cero.

Para realizar el entrenamiento de las CNN y obtener un mayor índice de éxito se necesitan grandes cantidades de imágenes similares a las que se esperan encontrar en el entorno de trabajo. Debido a esta necesidad, se han implementado en línea diversas bases de datos que recopilan imágenes de las aplicaciones más utilizadas en machine learning. Una de las bases de datos más empleadas por investigadores es ImageNet, la cual contiene más de 14 millones de imágenes clasificadas en al menos 20 mil categorías. Otra de las ventajas de usar ImageNet es que permite

realizar el procesamiento en la nube sin la necesidad de descargar las imágenes del dataset a analizar.

3.2.5 Aplicaciones de las CNN

Las CNN permiten automatizar operaciones tan complejas como el reconocimiento de rostros (Qu, D., Huang, Z., Gao, Z., Zhao, Y., & Song, G., 2018), reconocimiento de múltiples objetos en el entorno (Zhang, H., Lan, X., Zhou, X., Tian, Z., Zhang, Y., & Zheng, N., 2018), y la clasificación de objetos muy pequeños (Nguyen, T. V., Liu, L., & Nguyen, K., 2016) donde se logra clasificar dispositivos de montaje superficial (SMD) soldados en tarjetas de circuitos impresas (PCB). Esto permite que los robots sean capaces de entender el entorno donde se encuentra con mayor rapidez que otros métodos autónomos.

En cuanto a las CNN desarrolladas específicamente para el reconocimiento de anomalías en tuberías se presenta el desarrollado para realizar el reconocimiento del estado de tuberías de metal respecto a su nivel de corrosión (Vriesman, D., Junior, A. B., Zimmer, A., & Koerich, A. L., 2019). Se emplea una red con dos capas de convolución obteniendo una precisión mínima de 96%. Para este caso se emplea un modelo pre-entrenado que también fue empleado para la detección de anomalías en tuberías. En la Figura 3.6 se muestra la estructura de la CNN empleada. En otra investigación (Chen, K., Hu, H., Chen, C., Chen, L., & He, C., 2018) se realiza una detección con mayor detalle de anomalías dentro de tuberías empleando un CNN, que a comparación de otros algoritmos empleados para la detección autónoma de anomalías presenta mayor precisión y adaptabilidad al entorno.

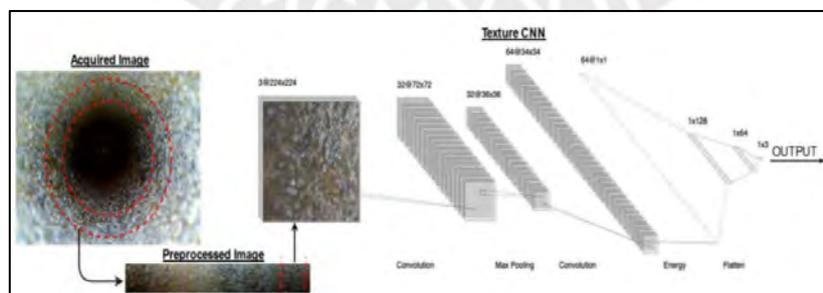


Figura 3.6. Estructura de CNN para reconocimiento del nivel de corrosión en tuberías de metal. (Vriesman, D., Junior, A. B., Zimmer, A., & Koerich, A. L., 2019)

3.3 Diseño del clasificador para el SDF

Una CNN necesita de una gran cantidad de información de entrada para realizar un entrenamiento adecuado y así poder reconocer una gran variedad de objetos de manera automática. Existen bibliotecas virtuales que poseen grandes cantidades de imágenes agrupadas convenientemente para entrenar CNNs dependiendo de su finalidad. Al realizar el entrenamiento, es necesario emplear imágenes que posean cierta similitud con las imágenes que obtendrá el robot durante su trabajo de inspección ya que esto afecta directamente la eficacia de la red.

La detección y aprendizaje de características por parte de las CNN se realizan durante los procesos de convolución, una misma red puede tener múltiples operaciones sucesivas de convolución. A mayor número de capas de convoluciones, será posible reconocer formas más complejas dentro de las imágenes pero el tiempo de ejecución de la red será mayor debido a que existe un mayor costo computacional. El máximo número de convoluciones a aplicar también dependen del tamaño en píxeles de la imagen de entrada y los procesos de subsampling aplicados a la salida de la operación de convolución.

Para seleccionar la estructura de CNN a emplear se realiza una comparación entre diferentes clasificadores de objetos como se muestra en la Tabla 3.3. Los clasificadores que más resaltan de esta comparación son los que se basan en YOLO.

Tabla 3.3: Comparación entre clasificadores. (Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M., 2020)

Method	Backbone	Size	FPS	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
YOLOv4: Optimal Speed and Accuracy of Object Detection									
YOLOv4	CSPDarknet-53	416	38 (M)	41.2%	62.8%	44.3%	20.4%	44.4%	56.0%
YOLOv4	CSPDarknet-53	512	31 (M)	43.0%	64.9%	46.5%	24.3%	46.1%	55.2%
YOLOv4	CSPDarknet-53	608	23 (M)	43.5%	65.7%	47.3%	26.7%	46.7%	53.3%
Learning Rich Features at High-Speed for Single-Shot Object Detection [84]									
LRF	VGG-16	300	76.9 (M)	32.0%	51.5%	33.8%	12.6%	34.9%	47.0%
LRF	ResNet-101	300	52.6 (M)	34.3%	54.1%	36.6%	13.2%	38.2%	50.7%
LRF	VGG-16	512	38.5 (M)	36.2%	56.6%	38.7%	19.0%	39.9%	48.8%
LRF	ResNet-101	512	31.3 (M)	37.3%	58.5%	39.7%	19.7%	42.8%	50.1%
Receptive Field Block Net for Accurate and Fast Object Detection [47]									
RFBNet	VGG-16	300	66.7 (M)	30.3%	49.3%	31.8%	11.8%	31.9%	45.9%
RFBNet	VGG-16	512	33.3 (M)	33.8%	54.2%	35.9%	16.2%	37.1%	47.4%
RFBNet-E	VGG-16	512	30.3 (M)	34.4%	55.7%	36.4%	17.6%	37.0%	47.6%
YOLOv3: An incremental improvement [63]									
YOLOv3	Darknet-53	320	45 (M)	28.2%	51.5%	29.7%	11.9%	30.6%	43.4%
YOLOv3	Darknet-53	416	35 (M)	31.0%	55.3%	32.3%	15.2%	33.2%	42.8%
YOLOv3	Darknet-53	608	20 (M)	33.0%	57.9%	34.4%	18.3%	35.4%	41.9%
YOLOv3-SPP	Darknet-53	608	20 (M)	36.2%	60.6%	38.2%	20.6%	37.4%	46.1%
SSD: Single shot multibox detector [50]									
SSD	VGG-16	300	43 (M)	25.1%	43.1%	25.8%	6.6%	25.9%	41.4%
SSD	VGG-16	512	22 (M)	28.8%	48.5%	30.3%	10.9%	31.8%	43.5%

En la Tabla 3.4 se comparan las distintas versiones de YOLO empleando un mismo dataset para el reconocimiento de distintos objetos. Se aprecia que el clasificador YOLO v5 (Zhu, X., Lyu, S., Wang, X., & Zhao, Q., 2021) presenta un mejor rendimiento en cuanto a su precisión media.

Tabla 3.4 Comparación de precisión entre distintas versiones de YOLO. (Zhu, X., Lyu, S., Wang, X., & Zhao, Q., 2021)

Label	YOLOv3 Average Precision	YOLOv4 Average Precision	YOLOv5l Average Precision
Small-Vehicle	29.25	39.62	44.8
Large-Vehicle	55.84	73.43	70.1
Plane	83.06	90.39	91.3
Storage-tank	44.69	61.52	63
Ship	71.19	82.67	78.6
Harbor	67.94	80.35	82.7
Ground-track-field	36.12	67.32	65.7
Soccer-ballfield	36.82	54.24	59.8
Tennis-court	87.30	92.57	92.7
Swimming-pool	39.76	57.57	65.4
baseball	61.35	76.62	75.8
roundabout	44.14	55.98	55.9
Basketball-court	37.79	63.04	64.5
bridge	26.65	42.41	50.1
helicopter	15.84	34.54	48.2

En base a estas comparaciones de rendimiento se concluye que el clasificador del robot para el reconocimiento de anomalías dentro de la tubería será el YOLO v5.

3.3.1 Estructura de YOLO v5.

En la Figura 3.7 se muestra la estructura, dividida en 4 etapas, que se empleará para la detección de anomalías dentro de las tuberías.

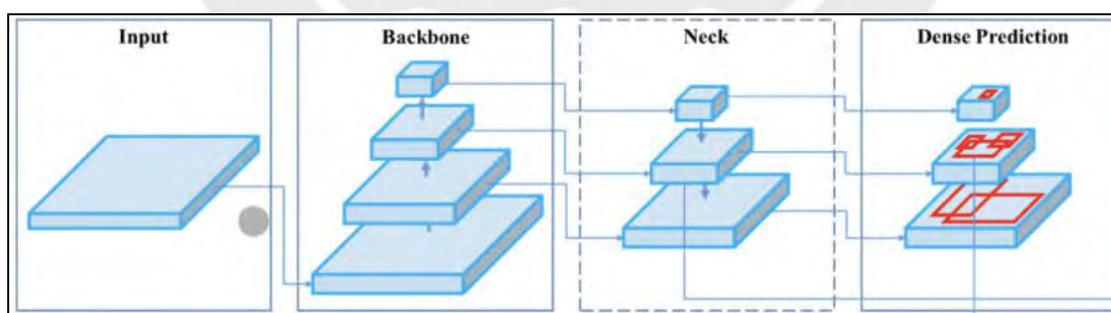


Figura 3.7 Estructura del clasificador YOLO v5. (Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M., 2020)

Etapas 1: Input - Imagen de entrada

El número de neuronas de la capa de entrada están definidas por el número de píxeles de las imágenes de entrada y del tipo de imagen. En la siguiente Tabla se

detalla la resolución de las imágenes recolectadas por el sistema de cámaras del robot móvil.

Tabla 3.5. Resolución de imágenes recolectadas por las cámaras del robot de inspección.

Modelo	Tipo de cámara	Resolución (píxeles)
RealSense D435i	Cámara de profundidad	1280 x 720
DS-2DE2A404IW-DE3	Cámara PTZ	2560 x 1440
Logitech C920 HD	Cámara convencional	1920 x 1080

Se escoge entonces trabajar con imágenes con una resolución de 1280 x 720 píxeles para que la CNN pueda ser empleada por todas las cámaras de la Tabla 3.5. Es importante mencionar que todas las cámaras transmiten imágenes a color por lo tanto el número de neuronas que conforman la capa de entrada sería:

$$1280 \times 720 \times 3 = 2\,764\,800 \text{ píxeles}$$

Etapa 2: Backbone - Extracción de características

Para la 2da etapa se realizan operaciones de convolución a la imagen de entrada empleando filtros. Cada operación de convolución permite extraer características de las imágenes y a mayor número de convoluciones realizadas se obtienen características más complejas pero también se va reduciendo la cantidad de píxeles con las que se trabaja. Para la red a implementar se van a realizar más de 50 operaciones de convolución. En las primeras operaciones de convolución se extraen características como líneas o curvas y en las últimas se extraen formas más complejas. Todas estas características extraídas son enviadas a la siguiente etapa de procesamiento. En la Figura 3.8 se muestra el funcionamiento de esta etapa de la red.

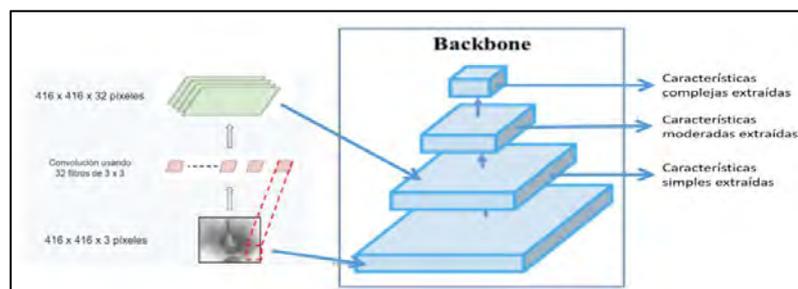


Figura 3.8 Estructura para la extracción de características de las imágenes.

La convolución es una operación producto escalar entre grupos de píxeles cercanos de la imagen de entrada y una matriz de dimensión reducida o kernel. Esta operación se aplica a todas las neuronas de entradas, es decir a cada pixel de la imagen. Por esta razón es esencial reducir el número de neuronas en la capa de entrada evitando incrementar el costo computacional de la CNN. Además se aplican múltiples kernels, que en conjunto son llamados filtros, a cada imagen obteniendo una matriz de salida por cada kernel aplicado.

La primera operación de convolución servirá para extraer características simples como contornos. Se emplearán 32 filtros de dimensiones 3 x 3 por lo tanto se obtienen 32 matrices de dimensiones 416 x 416. Los valores iniciales de cada kernel serán aleatorios e irán variando mediante el método de backpropagation. De esta manera se ajustarán en cada iteración. En la Figura 3.9 se muestra un ejemplo de la operación de convolución empleando un kernel de 3 x 3.

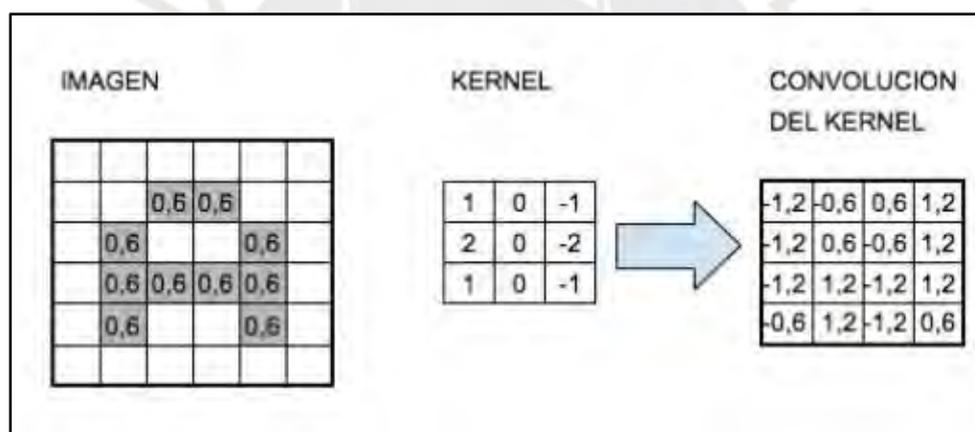


Figura 3.9. Convolución aplicada a imagen en escala de grises empleando un kernel de 3 x 3. (He, K., Zhang, X., Ren, S., & Sun, J., 2015)

La razón por la cual la matriz de salida es de 4 x 4 a pesar que la imagen de entrada es de 6 x 6 se debe a que en ciertos casos no se realiza el proceso de convolución a los píxeles del borde de la imagen y se aplica el método de zero padding. Es decir se rellenan con ceros los bordes de la matriz de salida como se muestra en la Figura 3.10. Cabe resaltar que este método asume que la información de los bordes de las imágenes no son importantes.

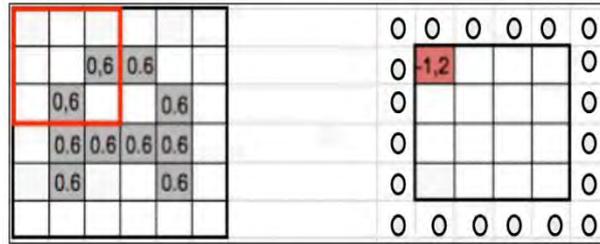


Figura 3.10. Ejemplo de zero padding aplicado al borde de la matriz de salida. (He, K., Zhang, X., Ren, S., & Sun, J., 2015)

A la obtención de las 32 matrices de salida se le conoce como mapeo de características. Como último paso se aplica una función de activación a cada elemento de cada matriz obteniendo como resultado un mapa de detección de características. En el caso de las CNNs generalmente se emplea la función de activación de tipo Rectifier Linear Unit (ReLU) que se define como:

$$f(x) = \max(0, x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

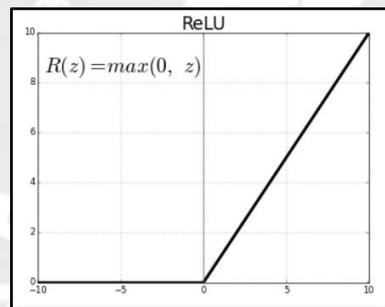


Figura 3.11. Gráfica de función de activación ReLU. (He, K., Zhang, X., Ren, S., & Sun, J., 2015)

La ventaja principal de emplear este tipo de función es que no posee regiones de saturación evitando el estancamiento en la actualización de los parámetros de la red, se obtiene la matriz del mapa de detección de características en la Figura 3.12.

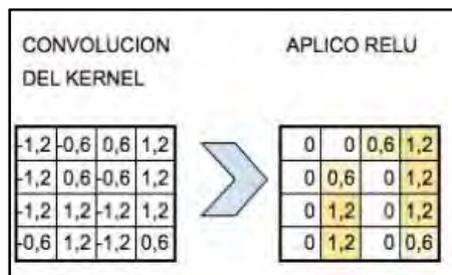


Figura 3.12. Obtención de mapa de detección de características. (He, K., Zhang, X., Ren, S., & Sun, J., 2015).

Etapa 3: Neck - Recolector de características

En esta etapa se almacenan todas las mapas de características extraídos en la etapa anterior y los ordena para que sean procesados empleando una red totalmente conectada (fully connected network) en la siguiente etapa de clasificación. El principal inconveniente es que la clasificación requiere que los mapas de características sean de un tamaño fijo. Para lograr que esto se cumpla sin importar el tamaño de las imágenes se emplean una serie de operaciones de submuestreo y sobremuestreo, a este método se le conoce como Spatial Pyramid Pooling (SPP) (He, K., Zhang, X., Ren, S., & Sun, J., 2015).

En la Figura 3.13 se muestra la estructura del tipo de SPP, donde se evidencia el procesamiento de los mapas de características a múltiples niveles permitiendo obtener predicciones en cada uno de ellos. Dichas predicciones pasan a la etapa de detección. Los mapas de características son evaluados desde el nivel superior al inferior aplicando un sobremuestreo (upsampling) y adicionándole la imagen filtrada del nivel correspondiente. De esta manera se asegura que la información empleada en la etapa de predicción mantenga un tamaño fijo sin importar el nivel en el que se encuentre (Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. 2017).

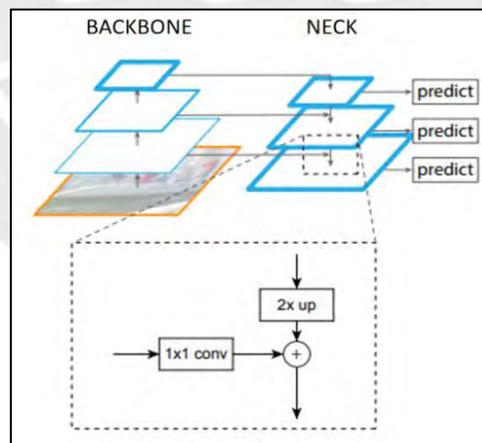


Figura 3.13 Recolección de características y empleo de SPP. (Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. 2017).

El método más empleado para realizar el submuestreo o “subsampling” es el Max-Pooling. El tamaño del filtro de Max-Pooling definirá el nuevo tamaño de la matrices de entrada de la capa intermedia. Por ejemplo, si se emplea un filtro de 2 x 2 se reducirán las dimensiones de las imágenes a la mitad. Para realizar el submuestreo

se divide la imagen en grupos de píxeles cuyo tamaño depende de las dimensiones del filtro seleccionado. Dentro de estas pequeñas divisiones se preserva el valor más alto del grupo de píxeles. De esta manera se reducen 4 píxeles a un solo píxel con el valor máximo. En la Figura 3.14 se muestra un ejemplo de Max-Pooling a una imagen de 4 x 4 píxeles obteniendo una salida de 2 x 2 con los valores más representativos.

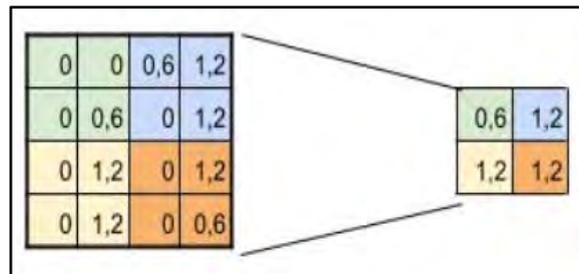


Figura 3.14 Submuestreo empleando Max-Pooling con un filtro de 2 x 2. (He, K., Zhang, X., Ren, S., & Sun, J., 2015).

Etapa 4: Head - Detección

En esta etapa se realiza la predicción en base a las características extraídas de cada imagen. El método empleado por Yolo v4 se denomina predicción densa. La predicción final está compuesta por un vector que contiene información de las coordenadas de un bounding box (cuadro delimitador), el cual contiene el objeto detectado. En la Figura 3.15 se muestra como se definen los bounding box en base a los datos de la predicción.

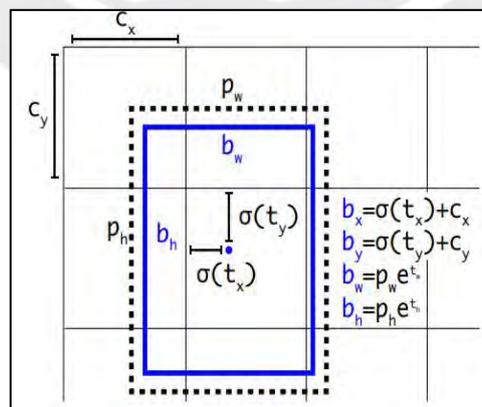


Figura 3.15 Obtención de bounding box de imagen en base al vector de predicción.

3.4 Implementación de CNN

3.4.1 Recolección y preprocesamiento de información de entrada

Se procedió con la recolección de imágenes de inspecciones internas de tuberías de agua de HDPE y concreto de tamaños a partir de 1 m de diámetro, estas imágenes fueron obtenidas de inspecciones realizadas por robots desarrollados por parte del Grupo de Innovación Tecnológica (GIT-PUCP). Se obtuvo un dataset de 71 imágenes las cuales fueron distribuidas como se muestra en la Tabla 3.6.

Tabla 3.6. Distribución de dataset para entrenamiento, validación y pruebas de CNN.

Propósito	Número de imágenes
Entrenamiento	50
Validación	14
Pruebas	7

El 70% de las imágenes serán empleadas en el entrenamiento de la red, un 20% se usan para la etapa de validación. Esta etapa permite comparar los resultados obtenidos por la red con los resultados esperados. El 10% restante se usan para pruebas. Todas las imágenes del dataset fueron etiquetadas con un bounding box por cada anomalía encontrada de manera manual. De esta manera se puede entrenar a la red y validar su precisión detectando anomalías en comparación al método manual. En la Figura 3.16 se muestra el resultado obtenido.



Figura 3.16. Izquierda: Imagen original. Derecha: Imagen con anomalía detectada por la CNN.

3.4.2 Acondicionamiento de computadora e instalación de paquetes

El equipo a emplear para el entrenamiento de la CNN posee un GPU modelo Tesla V100-SXM2-16GB. Se instala CUDA v11.4 y cuDNN para aprovechar el procesamiento del GPU y acelerar el entrenamiento. Es importante mencionar que en esta etapa se realiza el entrenamiento y pruebas preliminares en una computadora propia y la red neurodifusa será implementada posteriormente en la computadora del robot, la cual es una Nvidia Jetson Nano. Las características principales de la tarjeta se muestran en la Tabla 3.7.

Tabla 3.7. Principales especificaciones de la tarjeta Jetson Nano

Modelo de tarjeta	<p data-bbox="810 757 1023 786">Nvidia Jetson Nano</p> 
GPU	128-core Maxwell
CPU	Quad-core ARM A57 @ 1.43 GHz
Memoria	4 GB 64-bit LPDDR4 25.6 GB/s
Dimensiones	69 mm x 45 mm, 260-pin edge connector

Se instala la estructura de la CNN Yolo v5 obtenida de su repositorio en Github. Se configuran los directorios desde donde la red extrae imágenes para su entrenamiento, validación y testeo. Para emplear Yolo v5 es necesario contar con PyTorch y sus dependencias necesarias.

3.4.3 Entrenamiento de CNN

Se configura el número de clases a identificar en total, en este caso se han definido tres clases en base al tipo de anomalía: grieta, agujero y objeto extraño detectado. En base al número de clases se configuran parámetros como las subdivisiones de los mapas de características, la tasa de aprendizaje, etc. Todo esto influye en el tiempo de entrenamiento. Se realizó el entrenamiento de ambos tipos de clasificadores empleando el mismo dataset en las mismas condiciones.

Se realizó un entrenamiento preliminar empleando el clasificador Yolo v5 usando 4 diferentes parámetros como se muestra en la Tabla 3.8. Los parámetros internos de cada red o *weights* se actualizan luego de analizar el número de imágenes indicados en el tamaño de lote (*batch size*). El número de iteraciones en los que se procesarán todos los datos durante el entrenamiento está definido por los ciclos (*epochs*). En este caso se empleó un *batch size* de 16 imágenes y los *epochs* variaron entre 30 y 60. Los valores iniciales de los *weights* fueron diferentes en cada caso ya que se emplearon modelos pre-entrenados para mejorar el tiempo y precisión de la red. En el entrenamiento Train_1 y Train_2 se emplearon valores iniciales de un modelo reducido (yolo5s) y un modelo largo (yolo5l) respectivamente mientras que en Train_3 y Train_4 se usaron los valores finales obtenidos en Train_1 y Train_2.

Tabla 3.8. Entrenamiento preliminar de red con clasificador Yolo v5 bajo diferentes configuraciones.

Nombre	Valores iniciales (weights)	Nº de clases	Nombre de clases	Batch size	epochs
Train_1	yolo5s	3	["crack","perforation","str_object"]	16	30
Train_2	yolo5l	3	["crack","perforation","str_object"]	16	60
Train_3	Pesos obtenidos de Train_1	3	["crack","perforation","str_object"]	16	30
Train_4	Pesos obtenidos de Train_2	3	["crack","perforation","str_object"]	16	60

En base a los resultados obtenidos de este entrenamiento preliminar se decidió realizar un entrenamiento con el clasificador YOLOv5 con las consideraciones mostradas en la Tabla 3.9 llamado red_y5.

Tabla 3.9. Entrenamiento de red con clasificador Yolo v5.

Nombre	Valores iniciales (weights)	Nº de clases	Nombre de clases	Batch size	epochs
red_y5	yolo5l	3	["crack","perforation","str_object"]	16	300

3.4.4 Pruebas con CNN entrenada

Para evaluar los resultados de rendimiento de la red entrenada red_y5 se emplea una grabación de video de una inspección previa. El tiempo de procesamiento de la

red para el testeo fue en promedio de 11.75 milisegundos y se obtuvieron los resultados mostrados en la Figura 3.17.

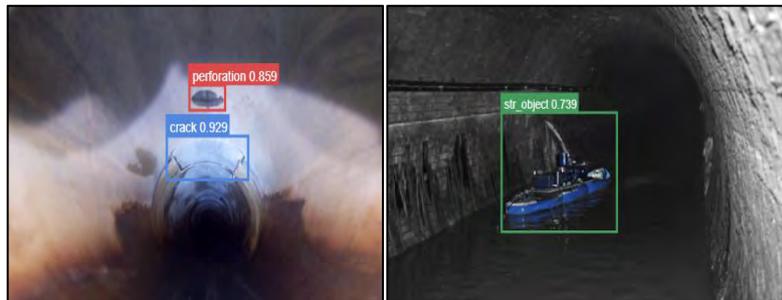


Figura 3.17. Reconocimiento automático de anomalías por parte de la CNN.

3.5 Acondicionamiento de datos de salida de CNN

La información obtenida por cada frame del video procesado con la CNN es publicada y almacenada en un archivo csv. El formato empleado para esta cadena de datos es el siguiente:

`<x> <y> <width> <height><probability> <object-class> <track-id><frames-counter>`

Donde `<x>` e `<y>` hacen referencia a las coordenadas del vértice izquierdo superior de cada bounding box y poseen un valor entre 0 y 1. Los datos `<width>` y `<height>` definen la altura y ancho del bounding box. Empleando estos datos es posible hallar el centro de cada bounding box (`x_bb, y_bb`) que servirá para controlar la orientación del robot como muestra en la Figura 3.18. Los datos almacenados en `<probability>` denotan la probabilidad de existencia de una anomalía y servirán para controlar la velocidad del robot.

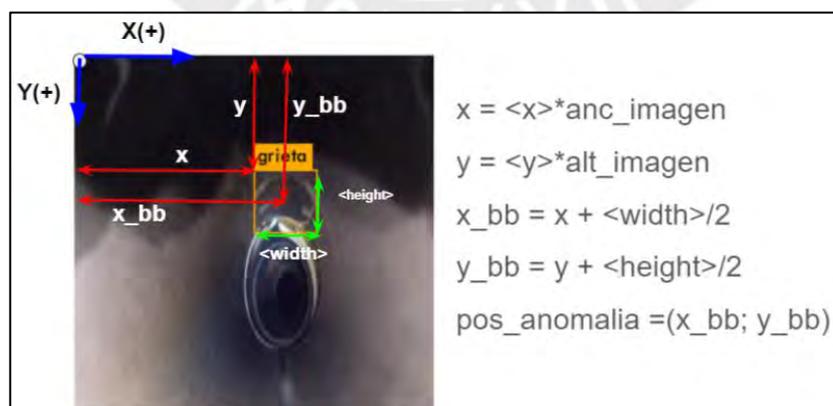


Figura 3.18. Cálculo del centro de un bounding box.

CAPÍTULO 4: DISEÑO DE CONTROLADOR

4.1 Marco teórico de controladores difusos para el control de robots móviles.

La lógica difusa, establecida por Zadeh en 1965 (Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D., 2011), se originó en la necesidad de expandir el concepto fundamental de conjunto en el que se basaba la teoría clásica. Los miembros de los conjuntos en la teoría clásica poseen diferencias claras y nítidas que permiten clasificarlos dentro de límites fijos (Woodman, O., & Harle, R., 2008). En el mundo real existen situaciones donde la información es ambigua por lo tanto no puede ser clasificada dentro de un solo conjunto sino que puede pertenecer de manera parcial a varios conjuntos en estos casos es donde se aplica la lógica difusa.

4.1.1 Teoría Clásica de conjuntos

Se tiene un conjunto universo U no vacío que contiene todos los posibles elementos que describen un contexto en particular. A cada elemento de U se le llamará "miembro". La agrupación de miembros originará un subconjunto S de manera que se cumple:

$$S \subseteq U$$

Dentro de U pueden existir más de un subconjunto y estos a su vez pueden compartir miembros o no. En la Figura 4.1 se muestra el conjunto universo U que contiene a los subconjuntos S y T los cuales comparten elementos por lo tanto la intersección entre ambos no es un subconjunto vacío ($S \cap T \neq \{ \}$).

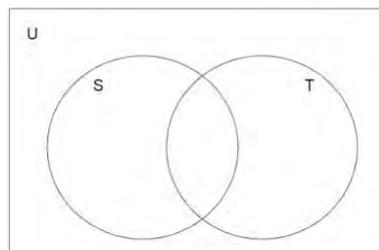


Figura 4.1. Representación clásica de dos subconjuntos con elementos en común.

En la teoría clásica cada miembro se encuentra incluido de manera completa en uno o más subconjuntos, es decir que pueden ser clasificados y pertenecen de manera total a algún subconjunto. Por lo tanto la siguiente función de pertenencia es aplicable:

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A, \end{cases}$$

4.1.2 Lógica difusa

A diferencia de la teoría clásica de conjuntos, en la lógica difusa los miembros tienen un grado de pertenencia que se encuentra entre 0 y 1 por lo que pueden pertenecer de manera parcial a diferentes subconjuntos. Para conocer el grado de pertenencia de los miembros a algún subconjunto difuso se emplean funciones características o de pertenencia. Las más usadas se muestran en la Figura 4.2: triangular, trapezoidal, gaussiana y sigmoïdal.

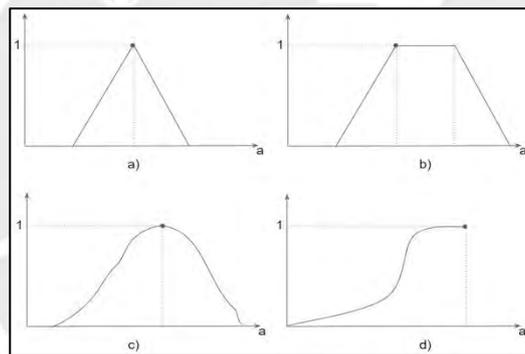


Figura 4.2. Funciones de pertenencia difusas más comunes: a) triangular, b) trapezoidal, c) gaussiana y d) sigmoïdal.

Los conjuntos difusos permiten la caracterización de manera no estricta y vaga como en el caso del lenguaje humano donde se usan diferentes variables lingüísticas para asignar características. En la Figura 4.3 se muestra un conjunto difuso que emplea funciones de pertenencia trapezoidales y triangulares para describir la velocidad de un objeto. La velocidad se clasificará en 3 conjuntos difusos: lenta (L), moderada (M) y rápida (R).

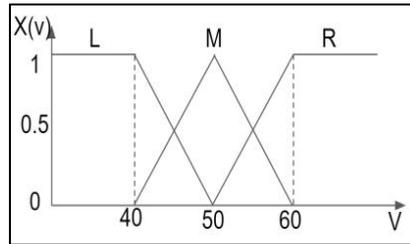


Figura 4.3. Conjunto difuso que describe la velocidad.

En la Tabla 4.1 se resumen las operaciones más comunes de los conjuntos difusos. Se emplean dos conjuntos difusos A y B.

Tabla 4.1. Operaciones de conjuntos difusos

Operación	Representación	Definición
Union	$\mu_{(A \cup B)}$	$\max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x)$
Intersección	$\mu_{(A \cap B)}$	$\min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x)$
Complemento	$\mu_{\bar{A}}(x)$	$1 - \mu_A(x)$
Producto	$\mu_{A \cdot B}(x)$	$\mu_A(x) \cdot \mu_B(x)$
Potencia	$\mu_A^\alpha(x)$	$[\mu_A(x)]^\alpha$
Diferencia	$A - B$	$(A \cap \bar{B})$

4.1.3 Estructura y aplicaciones de los controladores difusos

Los controladores difusos inician con una etapa de fusificación donde convierten la información de entrada a valores difusos. Cada variable de entrada va a estar caracterizada por diferentes conjuntos difusos. Los datos son evaluados mediante una base de reglas SI- ENTONCES empleando un Sistema de Inferencia Difuso (FIS) que modela el problema a resolver. Los modelos de IFS más usados son Mamdani, Takagi -Sugeno y Tsukamoto. La salida obtenida es defusificada para obtener un resultado con límites fijos. En la Figura 4.4 se muestra la estructura básica de un controlador difuso.

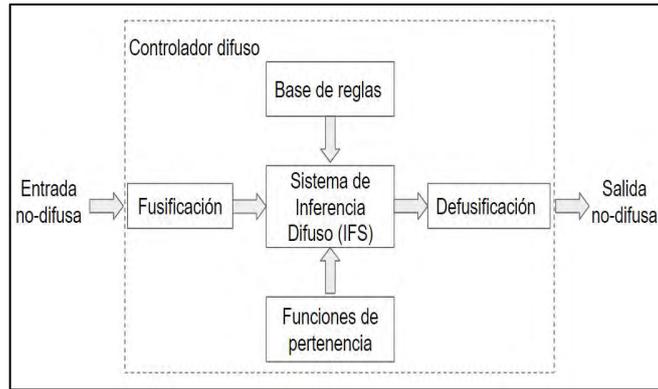


Figura 4.4. Estructura de controlador difuso.

Los robots para inspección de tuberías generalmente poseen sistemas de visión que permiten al operario conocer el entorno del robot durante su desplazamiento, se aprovecha el video recolectado para la identificación de anomalías dentro de la tubería. En (Mester, G., 2016) se presenta un vehículo robótico cuyo sistema de sensado se basa en una cámara omnidireccional y un emisor láser. Las imágenes obtenidas son pre-procesadas, se extraen sus características más resaltantes y se clasifican empleando un IFS. La base de reglas dependen del área, perímetro, circularidad, convexidad y ángulo de orientación de la anomalía encontrada.

Tabla 4.2. Base de reglas para clasificación de fallas dentro de tuberías. (Mester, G., 2016)

	Annular Crack	Vertical Crack	Corrosion
Area A_i	$A_i < TA_i$	$A_i < TA_i$	$A_i \geq TA_i$
Perimeter L_i	$L_i < TL_i$	$L_i < TL_i$	$L_i \geq TL_i$
Circularity C_i	$C_i < TC_i$	$C_i < TC_i$	$C_i \geq TC_i$
Convexity Cov_i	$Cov_i < TCov_i$	$Cov_i < TCov_i$	$Cov_i \geq TCov_i$
Boundary Eccentricity E_i	$E_i \geq TE_i$	$E_i \geq TE_i$	$1 \leq E_i \leq TE_i$
Angle θ_{iMaxA}	$\theta_{iMaxA} < T\theta_{iMaxA}$	$\theta_{iMaxA} > T\theta_{iMaxA} + 60^\circ$	-

Se detalla en los métodos a emplear para procesar la información recolectada por cámaras de Circuito Cerrado de Televisión (CCTV) montadas en robots de inspección (Bayar, G., Konukseven, E. I., & Koku, A. B., 2008). Las características evaluadas son la diferencia de intensidad entre los píxeles vecinos, la característica de la forma identificada y la relación de compacidad. En la siguiente Figura se muestra el diagrama de flujo del proceso de identificación de anomalías empleado en esta investigación.

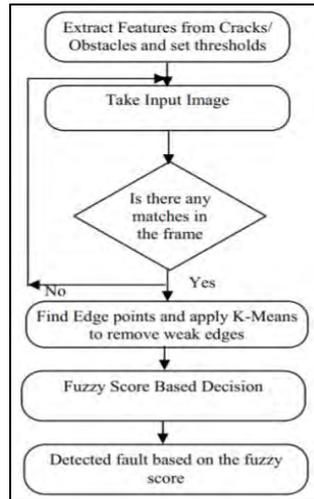


Figura 4.5. Diagrama de flujo para detección de fallas en tuberías empleando cámaras CCTV. (Bayar, G., Konukseven, E. I., & Koku, A. B., 2008)

Existen algoritmos difusos especializados para el reconocimiento de patrones como el agrupamiento Quick Fuzzy C-Mean (QFCM) presentado en (Xue-Fei, W., & Hua, B., 2009). Este algoritmo se base en la minimización de la función de costo:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2, 1 \leq m < \infty$$

Donde u es el grado de pertenencia de x en el subconjunto difuso j . Es posible identificar defectos y categorizarlos empleando los parámetros de área, ángulo de inclinación, longitud y grosor. Para esta investigación se emplearon imágenes recolectadas por cámaras en entornos de trabajo reales obteniendo con bastante precisión la ubicación de las anomalías como se muestra en la Figura 4.6.

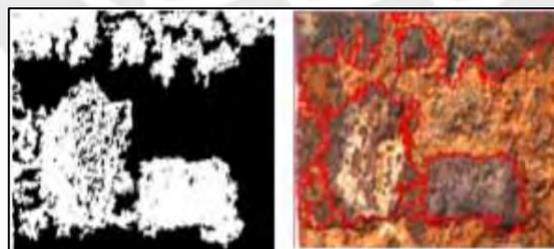


Figura 4.6 Extracción de características de interior de tubería empleando un sistema difuso (Xue-Fei, W., & Hua, B., 2009).

4.2 Diseño de Controlador de Velocidad y Orientación del Robot (CVOR)

Las variables de control del sistema serán la velocidad (v) del robot y su ángulo de orientación (θ). Las entradas del controlador difuso será el error de la orientación y

velocidad lineal del robot los cuales se hayan restando los valores deseados y los valores actuales del robot. Las salidas del controlador serán valores de voltaje para los motores del lado izquierdo y derecho del controlador. En la Figura 4.7 se muestra el bloque del controlador implementado con las entradas y salidas descritas acoplado a un bloque de cálculo de errores.

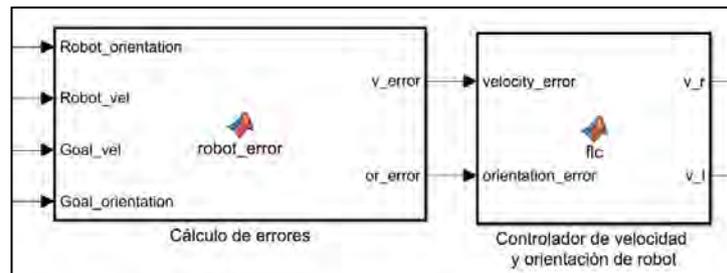


Figura 4.7 Bloque del controlador a implementar.

En la Figura 4.8 se muestran las funciones de pertenencia del error de la orientación del robot el cual puede variar entre -90° y 90° . Y en la Figura 4.9 se muestra las funciones de pertenencia del error de la velocidad lineal del robot que varía de 0 a 2.5 m/s.

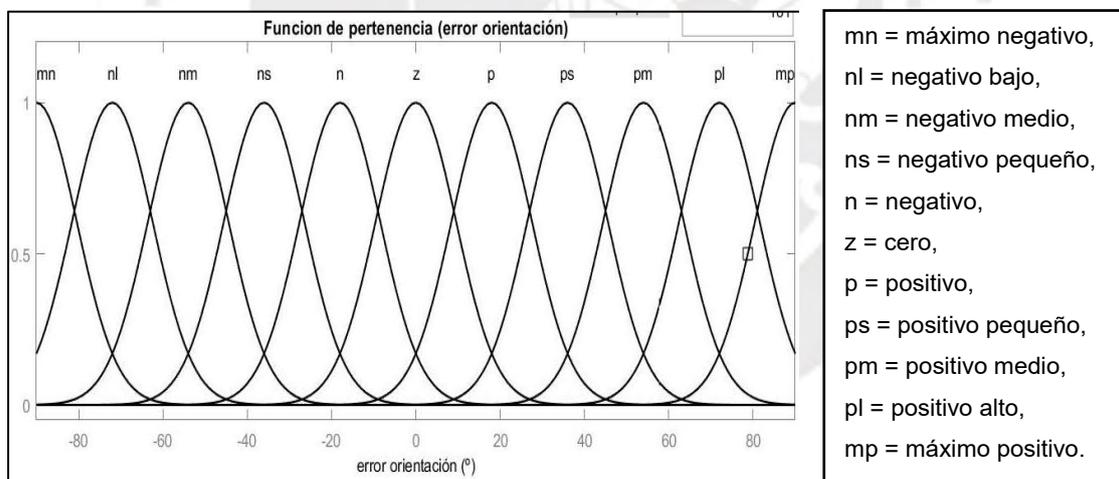


Figura 4.8. Funciones de pertenencia del error de orientación en grados ($^\circ$).

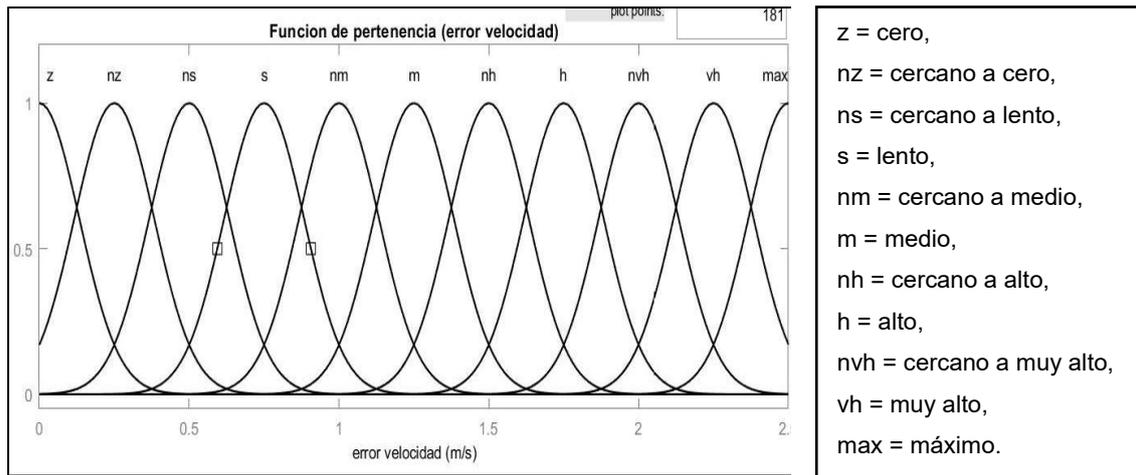


Figura 4.9. Funciones de pertenencia del error de velocidad en m/s.

Las salidas del controlador son el voltaje de alimentación de los motores del lado derecho y lado izquierdo. Al emplear los mismos motores en todas las ruedas del robot, las funciones de pertenencia son iguales. En la Figura 4.10 se muestra la salida para el voltaje suministrado a los motores del lado derecho.

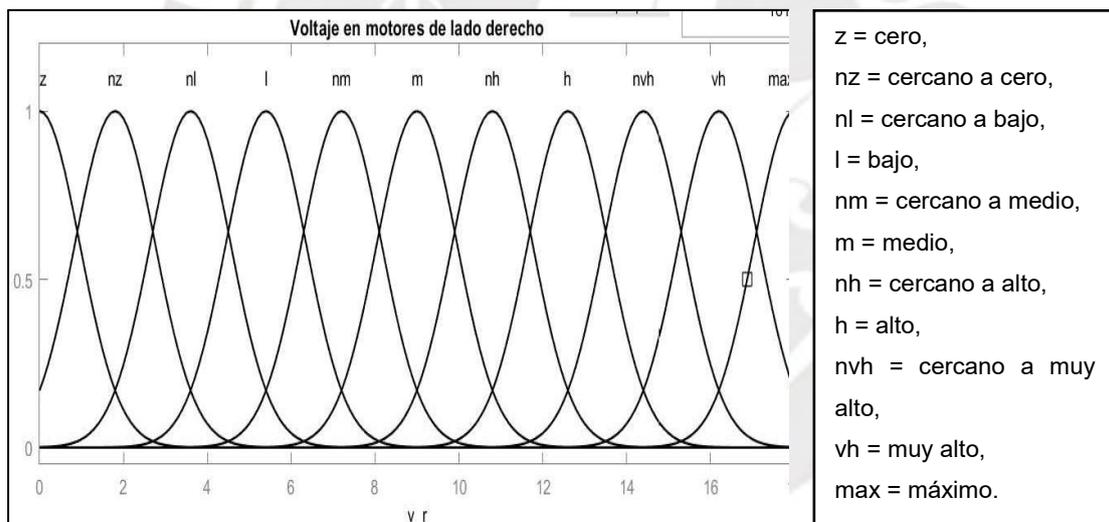


Figura 4.10. Funciones de pertenencia para el voltaje de los motores del robot.

4.2.1 Tabla de reglas

Se define la tabla de reglas con sus valores representativos correspondientes. El primer valor define el voltaje a los motores del lado izquierdo y el segundo, los motores del lado derecho.

Tabla 4.3. Tabla de Reglas a emplear en el controlador neuro-difuso.

Error velocidad /Error orientación	mn	nl	nm	ns	n	z	p	ps	pm	pl	mp
z	z,m	z,n m	z,l	z,nl	z,nz	z,z	nz,z	nl,z	l,z	nm,z	m,z
nz	z,nh	z,m	z,nm	z,l	z,nl	nz,nz	nl,z	l,z	nm,z	m,z	nh,z
ns	z,h	z,nh	z,m	z,nm	nz,l	nl,nl	l,nz	nm, z	m,z	nh,z	h,z
s	z,nvh	z,h	z,nh	nz,m	nl,nm	l,l	nm,nl	m,n z	nh,z	h,z	nvh,z
nm	z,vh	z,nv h	nz,h	nl,nh	l,m	nm,n m	m,l	nh,n l	h,nz	nvh, z	vh,z
m	z,max	nz,v h	nl,nv h	l,h	nm,n h	m,m	nh,n m	h,l	nvh,n l	vh,n z	max, z
nh	z,max	z,m ax	nz,vh	nl,nv h	nm,h	nh,nh	h,nm	nvh, nl	vh,nz	max, z	max, z
h	z,max	z,m ax	z,ma x	l,vh	m,nvh	h,h	nvh,m	vh,l	max, z	max, z	max, z
nvh	z,max	z,m ax	z,ma x	z,ma x	nh,vh	nvh,nv h	vh,nh	max, z	max, z	max, z	max, z
vh	z,max	z,m ax	z,ma x	z,ma x	z,max	vh,vh	max,z	max, z	max, z	max, z	max, z
max	z, max	z,m ax	z,ma x	z,ma x	z,max	max,m ax	max,z	max, z	max, z	max, z	max, z

4.3 Implementación de Lógica Difusa

Para implementar la lógica difusa del sistema se emplea el Toolbox Fuzzy Logic Designer de Matlab, agregando y configurando las entradas y salidas definidas en el capítulo anterior. En la Figura 4.12 se muestra la configuración implementada en Matlab para la lógica difusa empleando el método Mamdani.

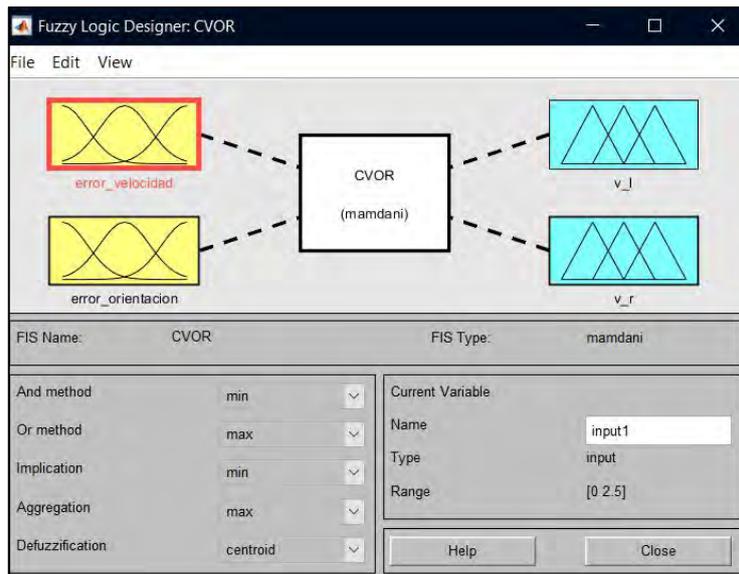


Figura 4.11. Implementación de lógica difusa en Matlab.

En la Figura 4.12 se muestran el comportamiento de cada salida en base al valor de las entradas. Se aprecia un comportamiento inverso entre ambas salidas lo cual va de acuerdo a lo observado en la Tabla de Reglas.

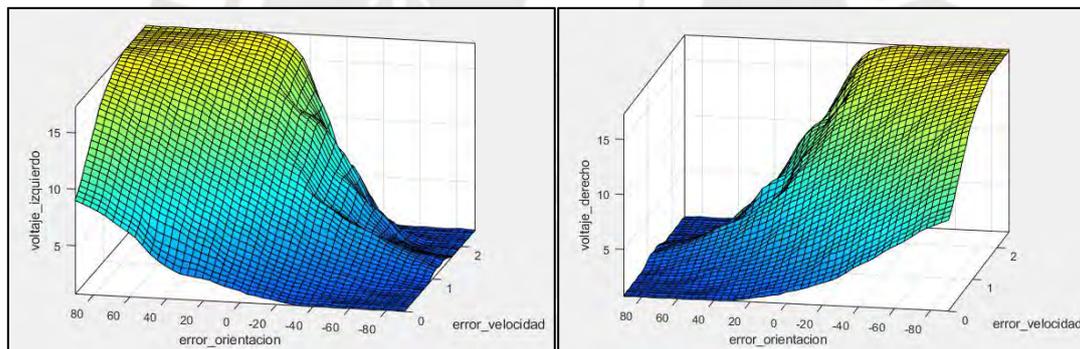


Figura 4.12. Gráficos de superficie para cada salida en base al valor de las entradas. Izquierda: Voltaje suministrado a motores izquierdos. Derecha: Voltaje suministrado a motores derechos.

Se observa que para un error de orientación de 0° , el voltaje del robot es proporcional al error de la velocidad en ambos casos, para estas condiciones todos los motores se mueven a la misma velocidad. Si el error de orientación es positivo, significa que el robot debe girar hacia la derecha por lo tanto el voltaje suministrado a los motores izquierdos siempre es mayor. Si el error de orientación es negativo, el voltaje suministrado a los motores derechos son mayores.

Se realiza una prueba de funcionamiento del controlador diseñado ingresando el valor de la orientación y velocidad deseados de manera manual. En la Figura 4.13 se muestra que para un error de velocidad de 1.1 m/s y un error de orientación de

33°, el voltaje de los motores del lado izquierdo y del lado derecho son de 11.2 V y 4.83 V respectivamente ya que el robot debe avanzar a una velocidad baja mientras gira hacia la derecha.

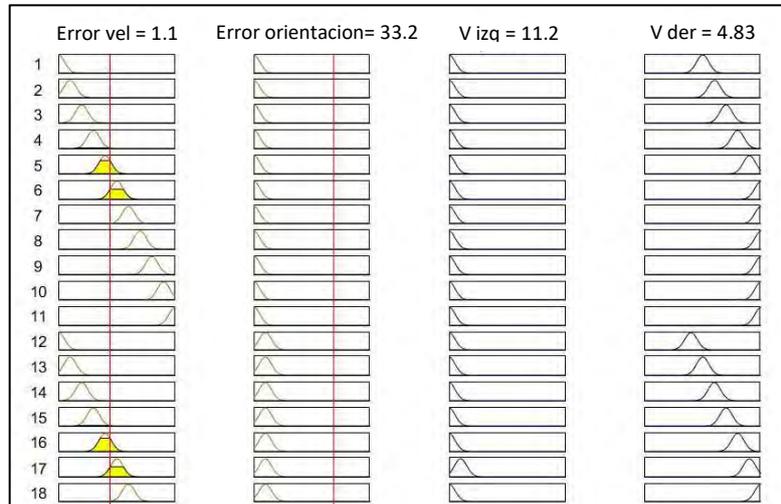


Figura 4.13. Pruebas de validación de controlador diseñado.

4.4 Integración de SDF y CVOR

Para el funcionamiento correcto del sistema es necesario integrar el SDF y el controlador diseñado, por ello se implementa una función de cálculo de error. Del SDF se obtienen salidas que describen la probabilidad de existencia (e) de anomalías y su posición en las imágenes (p). Ambas salidas tienen valores que se encuentran entre 0 y 1. En base al procesamiento de esta información es posible acondicionar las entradas como se muestra en la Figura 4.14.

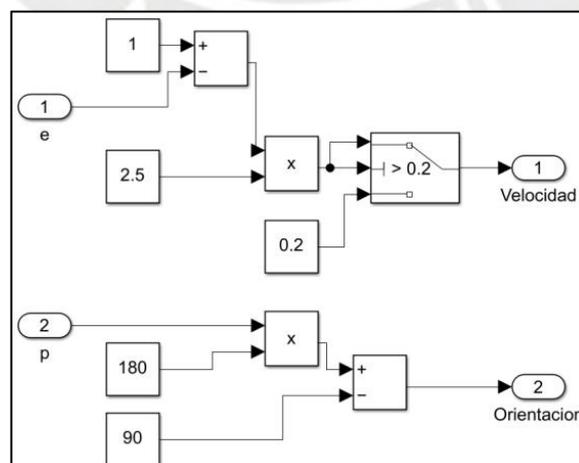


Figura 4.14 Acondicionamiento de salidas de la CNN para obtener la velocidad y orientación deseada del robot.

La velocidad mínima del robot es de 0.2 m/s para asegurar que el robot siempre avance y su velocidad máxima es de 2.5 m/s. La posición de la anomalía afecta la orientación del robot, el cual se encuentra entre -90° y 90° . En el caso que no se identifique ninguna anomalía, el valor de la orientación será de 0° . En la Figura 4.15 se detalla la estructura empleada desde el SDF al CVOR donde se obtienen como salida los voltajes de cada motor.

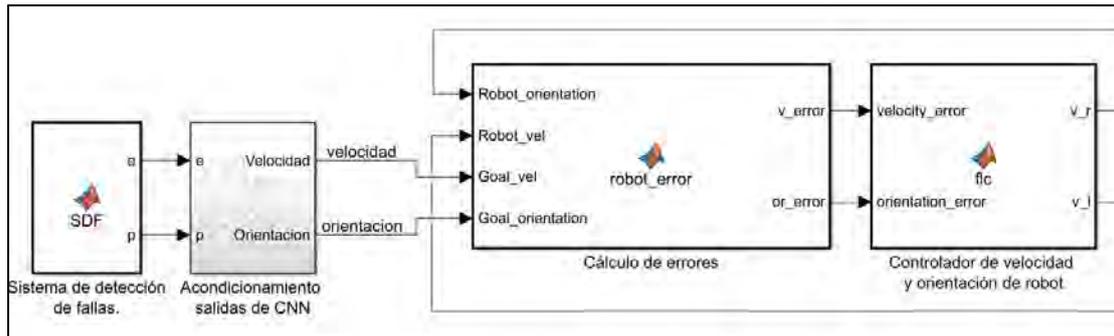


Figura 4.15 Integración de Sistema de Detección de Fallas y Controlador de Velocidad y Orientación del robot.

Se evalúa la integración del SDF y CVOR analizando un video de una inspección en tuberías matrices de alcantarillado y las respectivas salidas obtenidas. En la Figura 4.16 se muestran los resultados obtenidos de esta prueba preliminar.

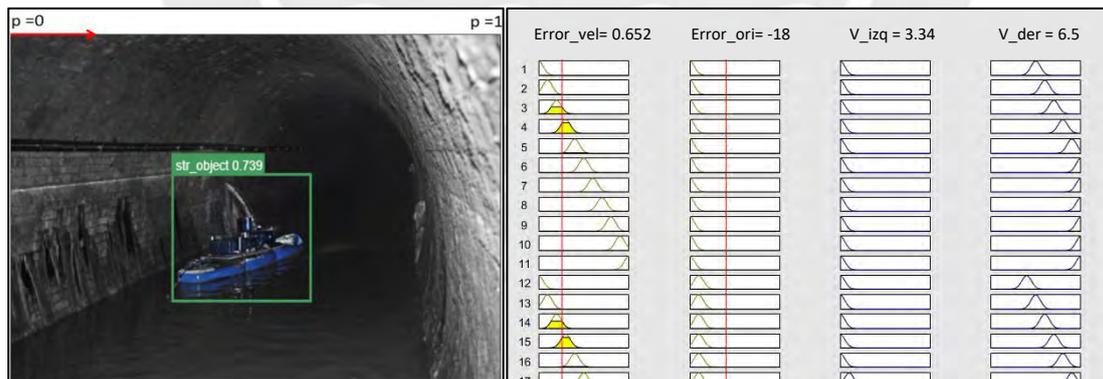


Figura 4.16 Pruebas de integración del SDF con el CVOR.

En estas pruebas se considera que las condiciones iniciales del robot son:

Velocidad de robot = 0,

Orientación de robot = 0.

En el instante inicial, el SDF identifica un objeto extraño con una probabilidad de existencia de $e = 0.739$, y se encuentra ubicado ligeramente a la izquierda de la imagen en una posición $p = 0.4$. Empleando las operaciones mostradas en el bloque de acondicionamiento se obtiene lo siguiente:

Velocidad deseada = 0.6525 m/s,

Orientación deseada = -18° .

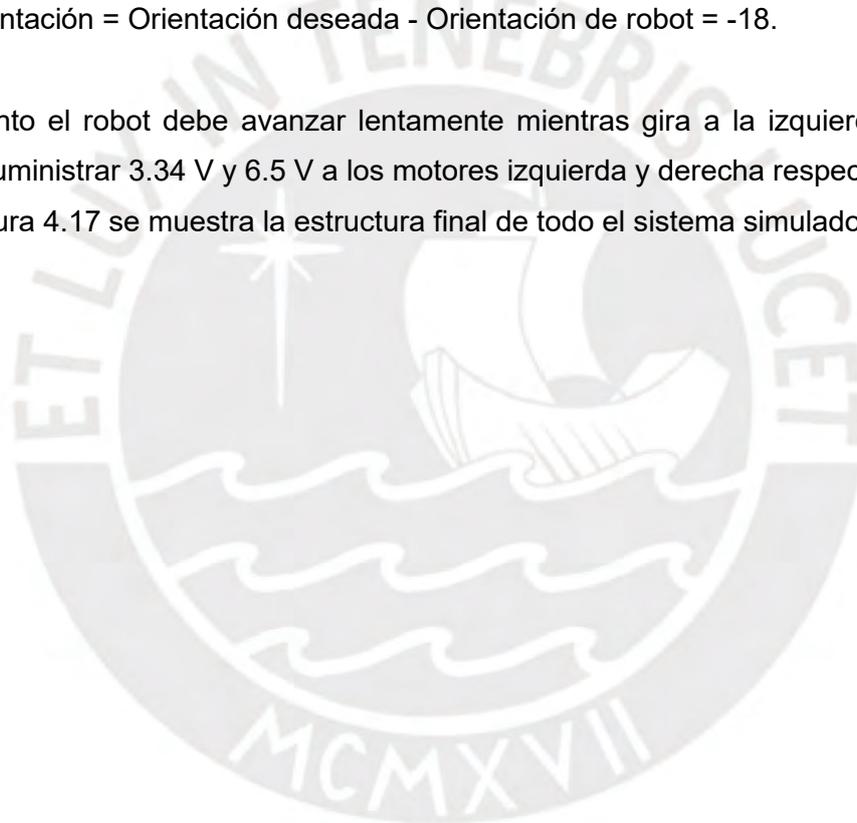
Los errores de la velocidad y orientación obtenidos en el bloque de cálculo son:

Error velocidad = Velocidad deseada - Velocidad de robot = 0.6525,

Error orientación = Orientación deseada - Orientación de robot = -18 .

Por lo tanto el robot debe avanzar lentamente mientras gira a la izquierda lo cual implica suministrar 3.34 V y 6.5 V a los motores izquierda y derecha respectivamente.

En la Figura 4.17 se muestra la estructura final de todo el sistema simulado,



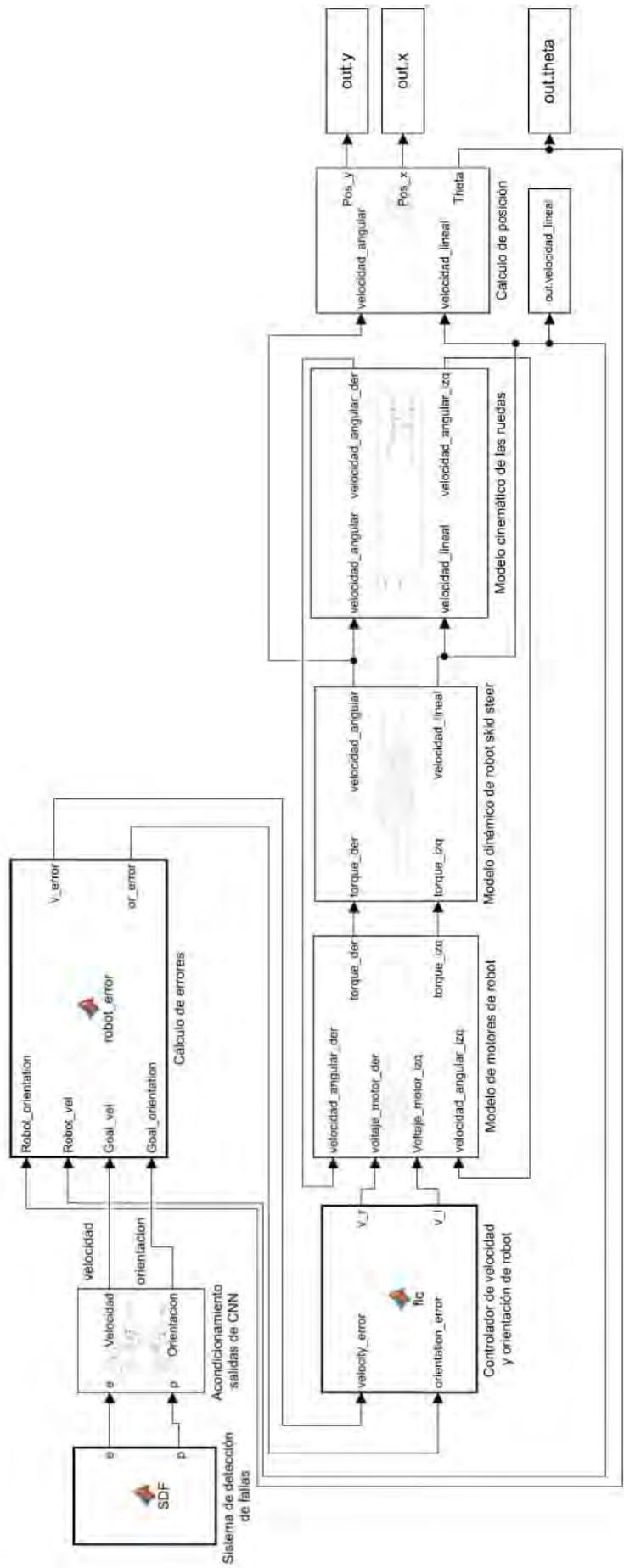


Figura 4.17 Diagrama empleado para la simulación del comportamiento autónomo del robot empleando el controlador diseñado.

CAPÍTULO 5: PRUEBAS Y RENDIMIENTO DE SISTEMA

5.1 Rendimiento del Sistema de Detección de Fallas

Para determinar si un objeto ha sido correctamente identificado se emplea el valor de Intersección sobre Unión (IoU) calculado de la siguiente manera:

$$IoU(A, B) = \frac{A \cap B}{A \cup B}$$

Donde A representa los píxeles que la red identifica como ocupados por un objeto a clasificar mientras que B representa los píxeles que realmente son parte del objeto a identificar. Se dice que un objeto ha sido correctamente identificado por una red si su valor de IoU es mayor a un umbral, generalmente se emplea 0.5.

Para cuantificar el resultado obtenido en el entrenamiento se emplean valores de *mean Average Precision* (mAP) a 0.5, el cual evalúa la precisión promedio de todas las clases a identificar considerando como predicciones correctas a los que superen un valor de 0.5 en su IoU. Los valores de *box loss* representan que tan cercanos son las dimensiones de cada *bounding box* predecida por la red con los delimitados de manera manual en las imágenes de entrada. Un menor valor de este indicador significa un acercamiento más ajustado. Los valores de *cls loss* miden la exactitud de la predicción de cada *bounding box* predecido, un valor menor implica una gran exactitud. En la Tabla 5.1 se muestran los resultados obtenidos de cada entrenamiento realizado al identificador.

Tabla 5.1. Resultados obtenidos del Sistema de Detección de Fallas entrenando el clasificador bajo diferentes configuraciones.

Nombre	Runtime (s)	mAP 0.5	Entrenamiento		Validación	
			box_loss	cls_loss	box_loss	cls_loss
Train_1	566	0.035	0.084	0.031	0.083	0.027
Train_2	1489	0.536	0.047	0.014	0.052	0.012
Train_3	427	0.281	0.067	0.027	0.065	0.022
Train_4	994	0.382	0.050	0.018	0.051	0.013

Se observa que en el caso de Train_2 el tiempo de entrenamiento fue mucho mayor y presenta una mayor precisión promedio (mAP) y un menor valor de box_loss y cls_loss, siendo el clasificador que presenta un resultado más preciso. En la Figura 5.1 se muestra la variación en cada iteración del cls_loss y box_loss en la etapa de entrenamiento. Un menor valor implica una mayor exactitud en las predicciones.

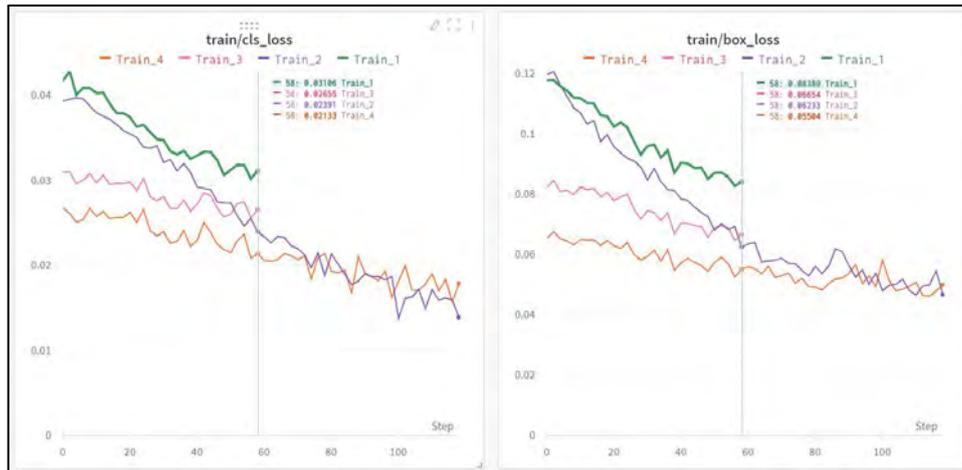


Figura 5.1. Evaluación de la exactitud de las predicciones de los clasificadores entrenados.

En la Figura 5.2 se muestran los gráficos de rendimiento de las redes entrenadas y el consumo de GPU, se evidencia que en el caso de Train_2 el consumo de recursos es de casi 30% más frente al entrenamiento de las otras redes.

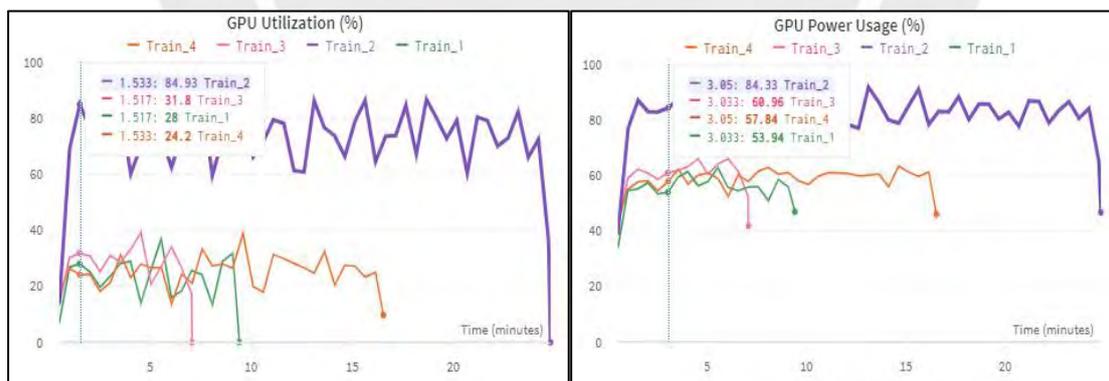


Figura 5.2. Comparación de consumo de GPU en cada clasificador entrenado.

Para mejorar aún más la precisión del clasificador se procedió a entrenar la red train_2 por un tiempo mayor. En la Tabla 5.2 se muestran los resultados obtenidos luego de reentrenar la red train_2, la cual se espera emplear en el sistema robótico.

Tabla 5.2. Rendimiento del clasificador luego del reentrenamiento de la CNN.

Nombre	Runtime (s)	mAP 0.5	Entrenamiento		Validación	
			box_loss	cls_loss	box_loss	cls_loss
train_2	8100	0.755	0.014	0.001	0.028	0.001
train_2 (antes de reentrenamiento)	1489	0.536	0.047	0.014	0.052	0.012

En la Figura 5.3 se muestra la mejora de precisión de la red al entrenarla por un tiempo mayor.

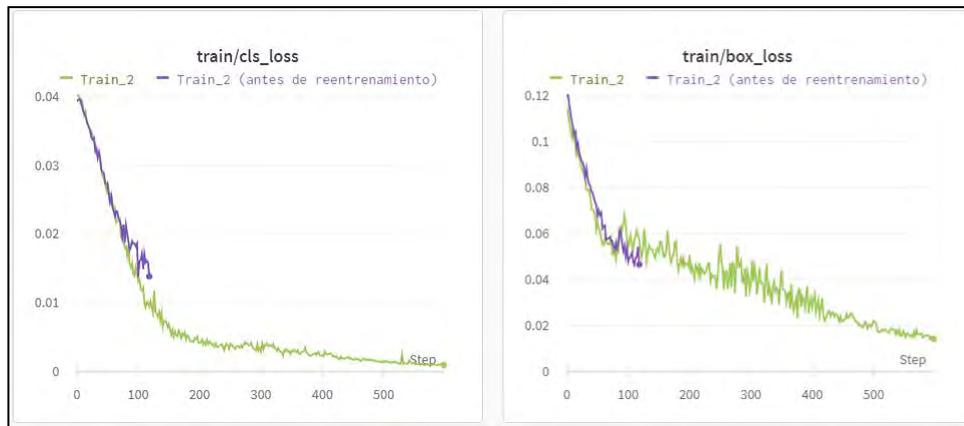


Figura 5.3. Evaluación de la mejora de la red Train_2 al entrenarla por un tiempo mayor.

En la Figura 5.4 se muestra la comparación del consumo de GPU en las redes entrenadas. Se evidencia que en el caso de la red y_2 reentrenada, se obtuvo un % consumo de recursos del GPU similar. La mayor diferencia ocurre en el tiempo de uso el cual es proporcional al tiempo que toma el entrenamiento.

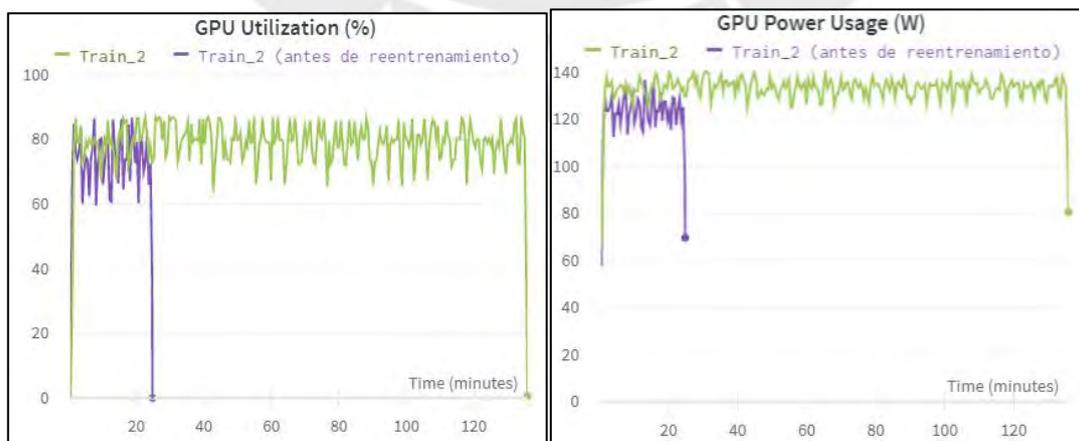


Figura 5.4. Evaluación del consumo de recursos del GPU durante el entrenamiento.

En la Figura 5.5 se muestran algunos de los resultados obtenidos con el Sistema de Detección de Fallas, el cual diferencia entre perforaciones, grietas y objetos extraños.

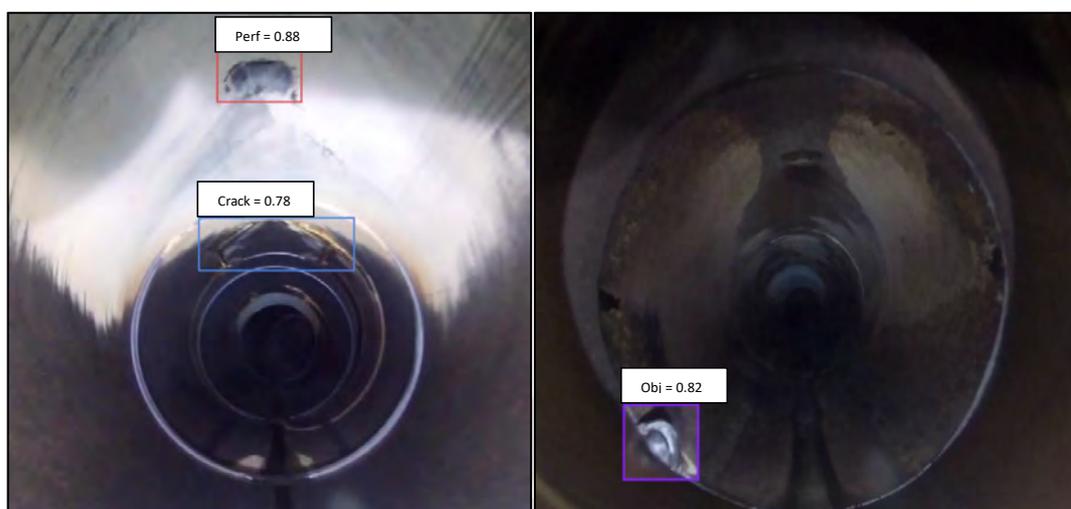


Figura 5.5 Resultados obtenidos por el SDF.

5.2 Resultados del sistema en general empleando vídeo de inspección

Para probar el funcionamiento del sistema en general se empleó un video de inspección que no fue usado en el entrenamiento ni en la validación del SDF. Se procesó en tiempo real un video de 10 minutos de duración. Se reconocieron dos tipos de anomalías: agujero y objeto extraño, en ambos casos ubicados en la parte izquierda de la tubería. En la Figura 5.6 y 5.7 se muestran las secuencias de imágenes obtenidas de cada anomalía identificada. La primera anomalía se reconoció durante el minuto 3:45 al 4:00. En el caso de la segunda anomalía, fue reconocida durante el minuto 7:15 al 7:35.

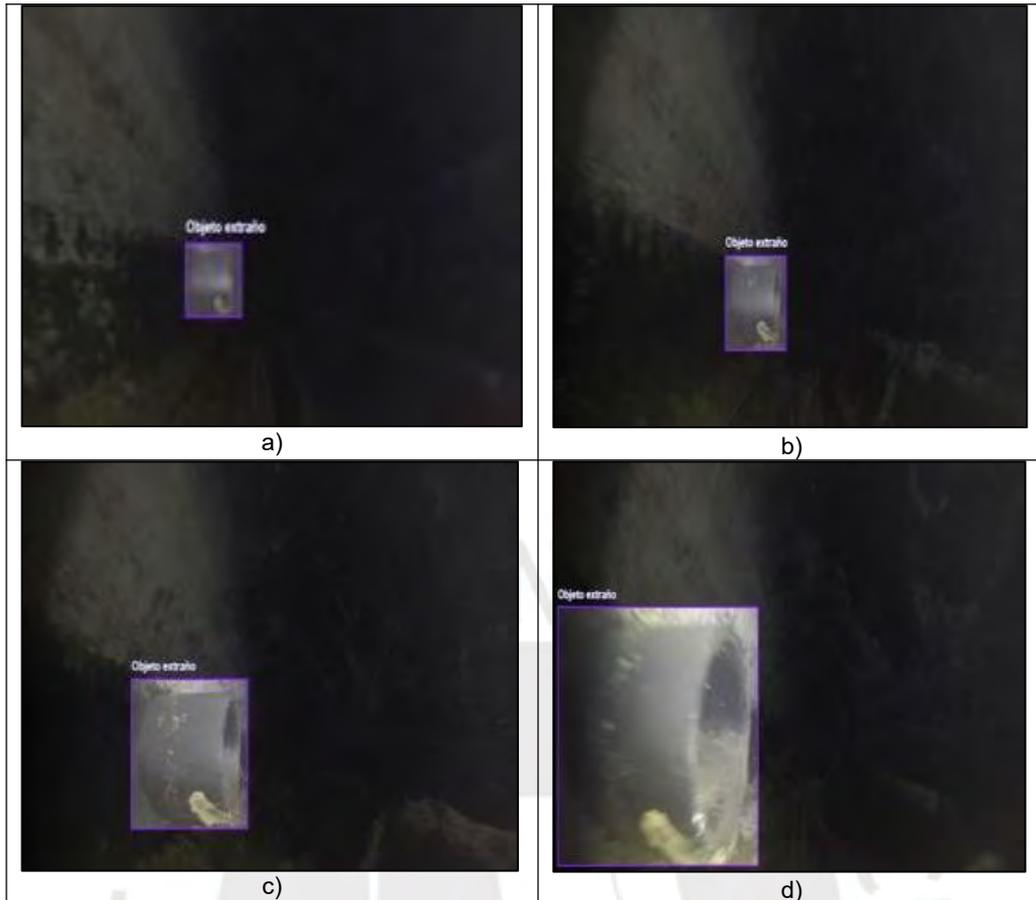


Figura 5.6. Secuencia de reconocimiento automático de objeto extraño (tubería de PVC).

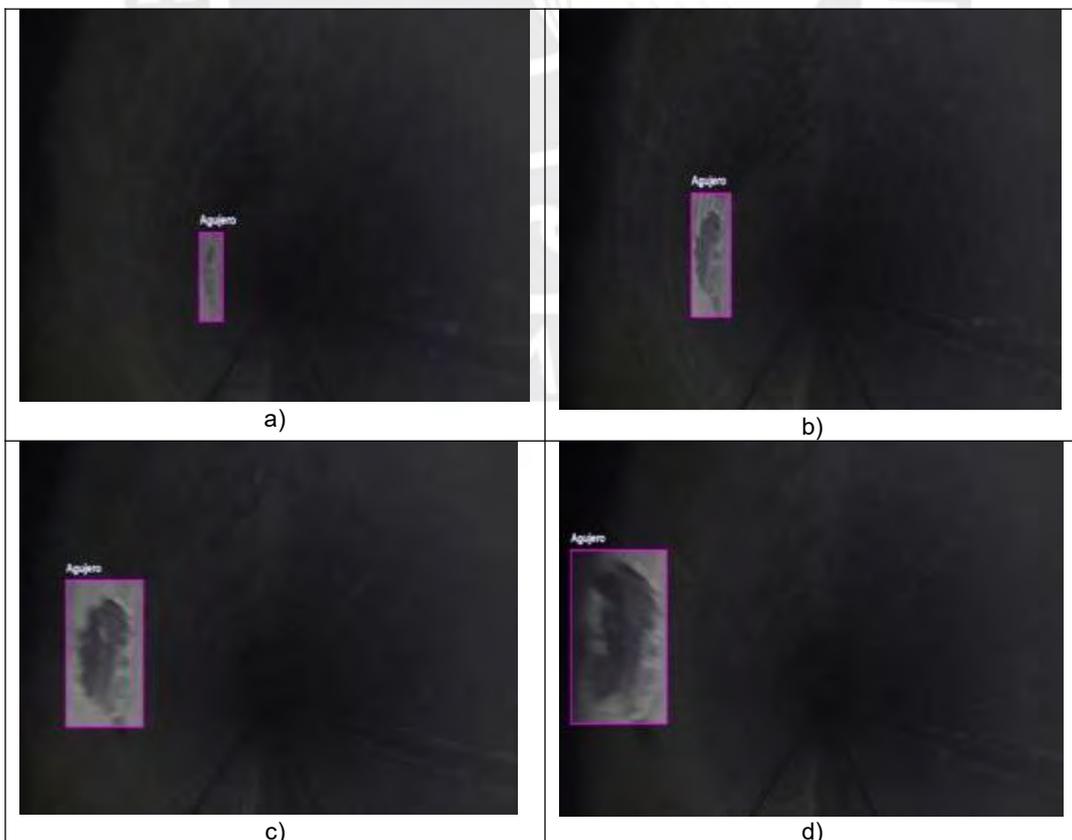


Figura 5.7. Secuencia de reconocimiento de agujero dentro de tubería.

En la Figura 5.8 se muestra de manera general la variación de p y e a lo largo del video. Se observa claramente los momentos en los cuales el robot reconoce las anomalías mencionadas anteriormente.

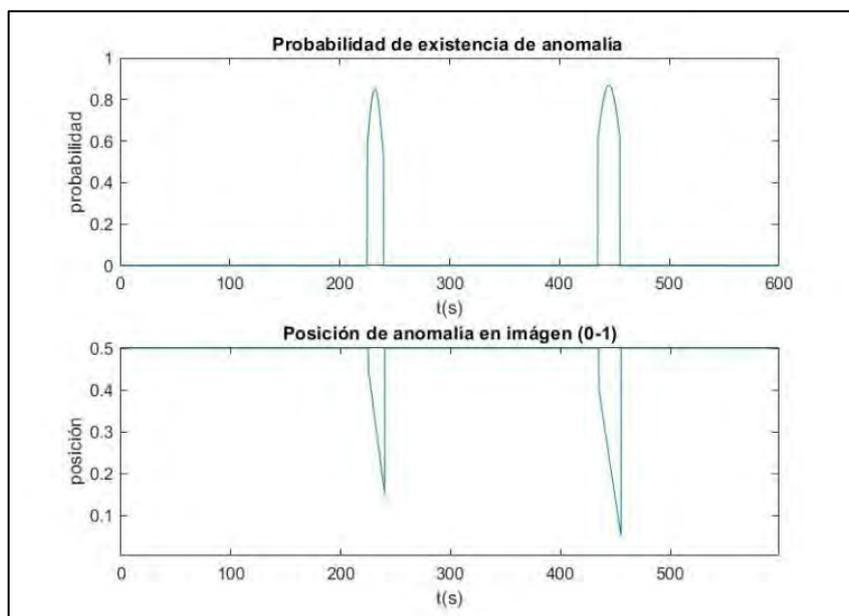


Figura 5.8 Valores de probabilidad y posición de anomalías reconocidas a lo largo del video.

La probabilidad de existencia de las anomalías llega a ser mayor de 0.8 al reconocer el objeto extraño y la perforación. La variable posición mantiene el valor de 0.5 cuando no se detecta ninguna anomalía. En este caso, ambas anomalías están ubicadas en el lado izquierdo de la tubería por lo que los valores de posición son menores a 0.5.

En la Figura 5.9 se muestran los resultados obtenidos en la salida del controlador, la velocidad (v) del robot se mantiene en 0.5 m/s durante la mayor parte del recorrido mientras que disminuye hasta 0.2 m/s ante la presencia de anomalías. Una vez que la anomalía ya no es percibida por la cámara del robot, su velocidad vuelve a aumentar y se mantiene constante hasta encontrar otra anomalía. El ángulo de orientación de la cámara (δ) se mantiene en 0° enfocando el centro de la tubería en todo momento hasta encontrar alguna anomalía. Al identificar algún indicio de esta, comienza a girar de manera para mantenerlo en su campo de visión por mayor tiempo y recolectar más información para un mejor diagnóstico.

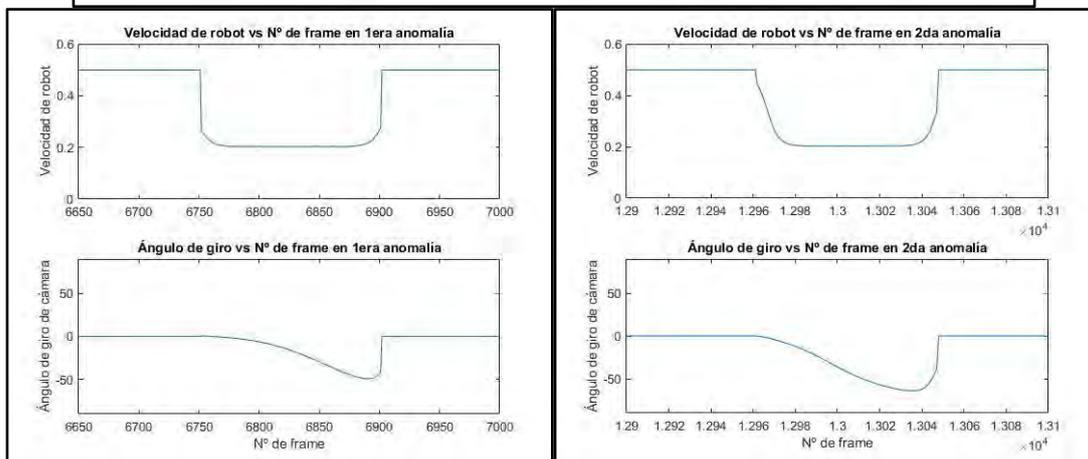
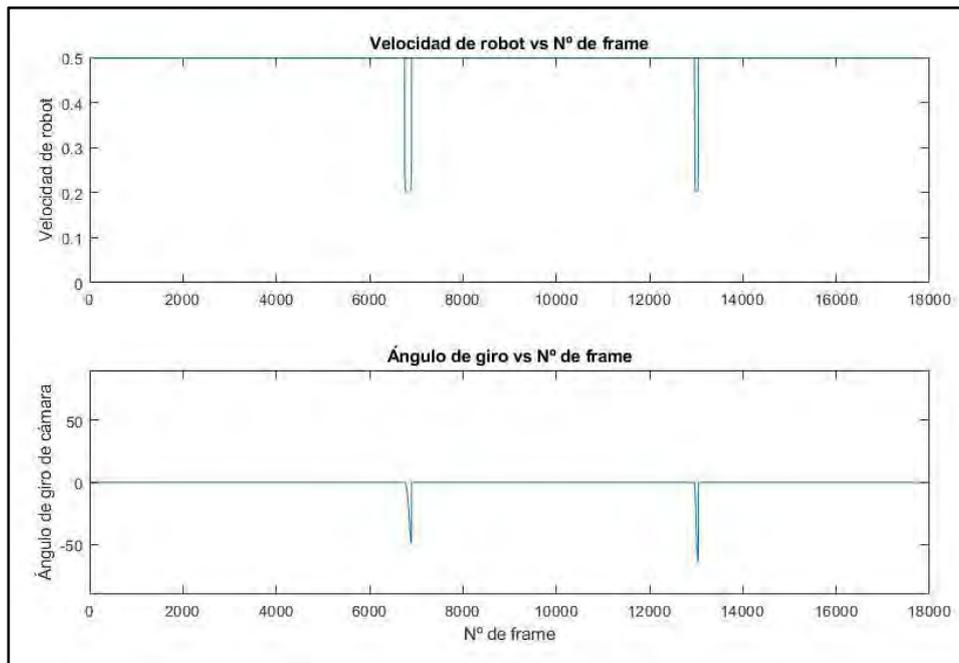


Figura 5.9. Variación de las salidas del controlador durante identificación de fallas.

En las pruebas realizadas, ambas anomalías identificadas se encontraban en el lado izquierdo de la tubería por lo que el ángulo de orientación disminuyó de manera continua hasta que ya no es posible seguir enfocando la anomalía. También se puede apreciar que en el instante que alguna falla deja se aleja del campo de visión de la cámara esta vuelve a su posición normal de manera rápida.

CONCLUSIONES

- Se logró diseñar un controlador neuro-difuso para el reconocimiento autónomo de 3 tipos de anomalías dentro de tuberías empleando un sistema robótico permitiendo controlar la velocidad de desplazamiento del robot en un rango de 0.2 a 0.5 m/s y el ángulo de orientación de la cámara del robot de -90° a 90° . Esto facilita la captura de una mayor cantidad de información gráfica de cada posible anomalía y aumenta la calidad del diagnóstico del estado de la tubería.
- Se contó con información limitada para el armado del dataset que consistió en 64 imágenes de diferentes anomalías encontradas durante la inspección interna de ductos y tuberías. Se emplearon 50 imágenes en el entrenamiento y 14 imágenes se usaron para la etapa de validación de la red. La red entrenada con mejor resultado presenta una precisión promedio (mAP) de 75.5% y sensibilidad de 73% empleando el clasificador Yolo v5.
- Se implementó un script en python para la extracción en tiempo real de datos obtenidos por la CNN y la evaluación de dichos datos empleando la lógica difusa mostrada. De esta manera se aseguró la interacción entre el componente neuronal y difusa del controlador neuro-difuso.
- Se realizaron pruebas empleando un video de inspección real grabado por el robot presentado en el Capítulo 2, se observó que el controlador fue capaz de reconocer un objeto extraño y agujero. No hubo indicios de lecturas falsas en todo el recorrido. Se comprobó que la velocidad del robot se reduce al reconocer anomalías y el ángulo de giro de la cámara corresponde a la posición de la anomalía en todo momento.
- Se concluye que con el controlador diseñado permite minimizar la intervención de un operador para el reconocimiento de anomalías dentro de tuberías primarias de agua. Asimismo permite automatizar el diagnóstico del estado de la tubería inspeccionada.

RECOMENDACIONES

- Para mejorar el rendimiento del controlador diseñado se recomienda mejorar la etapa de entrenamiento de la CNN aumentando el número de imágenes que conforman el dataset. Se recomienda realizar inspecciones adicionales con el robot bajo diferentes condiciones. Asimismo esto permitiría aumentar la cantidad de tipos de anomalías a reconocer y se podría dividir el tipo de anomalía “grietas” en “grietas horizontales” y “grietas verticales”.
- Se recomienda emplear una computadora con mayor potencia para facilitar la etapa de entrenamiento y permitir aumentar la resolución de las imágenes hasta 600 x 900 píxeles como mínimo y los FPS a 60. Esto permitirá capturar más detalles de las anomalías además que permitirá aumentar el tiempo de respuesta del robot disminuyendo el tiempo de inspección.
- Se recomienda considerar mecanismos que eviten salpicaduras de agua o barro en la cámara del robot o permitan limpiarlo en caso se ensucie ya que esto afectaría drásticamente el desempeño del controlador. Otra opción es el uso de múltiples cámaras, pero se debe considerar el costo computacional y energético.

BIBLIOGRAFIA

Yamashita, A., Matsui, K., Kawanishi, R., Kaneko, T., Murakami, T., Omori, H., ... & Asama, H. (2011, December). Self-localization and 3-D model construction of pipe by earthworm robot equipped with omni-directional rangefinder. In 2011 IEEE International Conference on Robotics and Biomimetics (pp. 1017-1023). IEEE.

Streich, H., & Adria, O. (2004, April). Software approach for the autonomous inspection robot MAKRO. In IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 (Vol. 4, pp. 3411-3416). IEEE.

Brunete, A., Torres, J. E., Hernando, M., & Gambao, E. (2007, December). Multi-drive control for in-pipe snakelike heterogeneous modular micro-robots. In 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO) (pp. 490-495). IEEE.

Adams, W., Sridar, S., Thalman, C. M., Copenhaver, B., Elsaad, H., & Polygerinos, P. (2018, April). Water pipe robot utilizing soft inflatable actuators. In 2018 IEEE International Conference on Soft Robotics (RoboSoft) (pp. 321-326). IEEE.

Dvajasvie, G., Farisha, B. P., Babu, S. N., Saheen, K. P., & Binoy, N. C. (2018, June). Leak Detection in Water-Distribution Pipe System. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1-4). IEEE.

Hu, Y., Song, Z., & Zhu, J. (2012, September). Estimating the posture of pipeline inspection robot with a 2d laser rang finder. In 2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI) (pp. 401-406). IEEE

Saenz, J., Elkmann, N., Stuerze, T., Kutzner, S., & Althoff, H. (2010, October). Robotic systems for cleaning and inspection of large concrete pipes. In 2010 1st International Conference on Applied Robotics for the Power Industry (pp. 1-7). IEEE.

Le, D. V. K., Chen, Z., & Rajkumar, R. (2019). Multi-sensors in-line inspection robot for pipe flaws detection. IET Science, Measurement & Technology.

Mehrandezh, M., Najjaran, H., Paranjape, R., & Poozesh, S. (2008, September). Dynamic analysis and control of a robotic pipe crawler. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 4182-4182). IEEE.

Murtra, A. C., & Tur, J. M. M. (2013, April). IMU and cable encoder data fusion for in-pipe mobile robot localization. In 2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA) (pp. 1-6). IEEE.

Pinto, D. S. S., & da Silva, K. R. G. (2016, October). Robot position control in pipes using Q Learning. In 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 004609-004613). IEEE.

Jayakumar, D., Jaganath, R., & Selvarasu, R. (2016, November). Defect identification in pipelines using inspection robot. In 2016 Online International Conference on Green Engineering and Technologies (IC-GET) (pp. 1-5). IEEE

SUNASS. (2020). *Benchmarking regulatorio 2020 de las Empresas Prestadoras (EP)*. Dirección de Fiscalización.

Torres, L., Verde, C., Carrera, R., & Cayetano, R. (2014). Algoritmos de diagnóstico para fallas en ductos. *Tecnología y ciencias del agua*, 5(4), 57-78

González, O., Verde, C., & Torres, L. (2017). Leak estimation method for complex pipelines with branch junctions. *Journal of Pressure Vessel Technology*, 139(2), 021701.

Khan, M. S. (2017, October). An approach for crack detection in sewer pipes using acoustic signals. In 2017 IEEE Global Humanitarian Technology Conference (GHTC) (pp. 1-6). IEEE.

Norli, P., Frijlink, M., Standal, Ø. K. V., Bjåstad, T. G., Prieur, F., & Vallée, E. (2018, October). Ultrasonic Detection of Stress Corrosion Cracks in Pipe Samples Using Guided Waves. In 2018 IEEE International Ultrasonics Symposium (IUS) (pp. 1-4). IEEE.

Ulapane, N., Alempijevic, A., Vidal-Calleja, T., Miro, J. V., Rudd, J., & Roubal, M. (2014, June). Gaussian process for interpreting pulsed eddy current signals for ferromagnetic pipe profiling. In 2014 9th IEEE Conference on Industrial Electronics and Applications (pp. 1762-1767). IEEE.

Vidal-Calleja, T., Miró, J. V., Martín, F., Lingnau, D. C., & Russell, D. E. (2014, September). Automatic detection and verification of pipeline construction features with multi-modal data. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 3116-3122). IEEE.

Baturone, A. O. (2005). *Robótica: manipuladores y robots móviles*. Marcombo.

Madow, A., Martinez, J. L., Morales, J., Blanco, J. L., Garcia-Cerezo, A., & Gonzalez, J. (2007, October). Experimental kinematics for wheeled skid-steer mobile robots. In 2007 IEEE/RSJ international conference on intelligent robots and systems (pp. 1222-1227). IEEE.

Mohareri, O., Dhaouadi, R., & Rad, A. B. (2012). Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks. *Neurocomputing*, 88, 54-66.

Qu, D., Huang, Z., Gao, Z., Zhao, Y., & Song, G. (2018, December). An automatic system for smile recognition based on CNN and face detection. In 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO) (pp. 243-247). IEEE.

Zhang, H., Lan, X., Zhou, X., Tian, Z., Zhang, Y., & Zheng, N. (2018, November). Visual manipulation relationship network for autonomous robotics. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)* (pp. 118-125). IEEE.

Nguyen, T. V., Liu, L., & Nguyen, K. (2016, November). Exploiting generic multi-level convolutional neural networks for scene understanding. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)* (pp. 1-6). IEEE.

Vriesman, D., Junior, A. B., Zimmer, A., & Koerich, A. L. (2019, November). Texture CNN for thermoelectric metal pipe image classification. In *2019 IEEE 31st international conference on tools with artificial intelligence (ICTAI)* (pp. 569-574). IEEE.

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Zhu, X., Lyu, S., Wang, X., & Zhao, Q. (2021). TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 2778-2788).

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9), 1904-1916.

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117-2125).

Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). *Introduction to autonomous mobile robots*. MIT press.

Woodman, O., & Harle, R. (2008, September). Pedestrian localisation for indoor environments. In *Proceedings of the 10th international conference on Ubiquitous computing* (pp. 114-123).

Xue-Fei, W., & Hua, B. (2009, November). Automated assessment of buried pipeline defects by image processing. In *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems* (Vol. 4, pp. 583-587). IEEE.

