Design and Fabrication of In-Pipe Inspection Robot for Crack Analysis and Detection Using OpenCv and Visual Odometry

Alwin Tony[1], Adil Noushad[2], Aquilas Ninan[3], Alwin A V[4], Naveen Prakash K V[5]

[1,2,3,4] UG Scholar, Department of Mechanical Engineering, Toc H Institute Of Science and Technology, Ernakulam, India

[5]Assistant Professor, Dept. of Mechanical Engineering Toc H Institute Of Science and Technology, Ernakulam, India

---

Abstract - *This project focuses on the design and enhancement of electronics control systems for pipe inspection robots. Currently, these robots are primarily manufactured offshore, resulting in high costs and inadequate support during malfunctions, leading to environmental mishaps. The objective is to leverage existing mechanical platforms and develop a reliable, robust, and cost-effective in-pipe inspection robot. Existing pipe inspection robot models worldwide exhibit significant performance variations. Many systems have limited applicability, lack reliability, and heavily rely on user experience. This project aims to overcome these challenges by developing an innovative robot capable of navigating pipelines and accurately detecting crack propagation. The robot must operate reliably in confined environments and provide technicians with digital video feedback. It will incorporate off-the-shelf components, inexpensive materials, and allow for potential on-site repairs, emphasizing traditional mechanical engineering principles. It enables the robot to autonomously distinguish between cracks and other voids, eliminating the need for additional human intervention. The proposed self-initiating distinction between cracks and voids streamlines inspections, minimizing human error and enhancing efficiency. Design of a six-legged 120 degree apart symmetric structure for the robot for easy manoeuvrability equipped with wifi module, Microcontroller, and set of cameras for computer vision. Utilises OpenCv and haar Cascading techniques and python programming to detect crack from live video feedback and shows the output to user outside. Absolute distance is also measured using visual odometry wherein change in pixels of multiple images is calculated to distance.*

Key Words: ***Pipe Inspection Robot, OpenCv, Haar Cascading, Visual Odometry, Crack analysis, Manoeuvrable Robot***

1. INTRODUCTION

Pipes are used in many industries to transport various kinds of fluids and gases, many of these pipes are long and inaccessible to technicians and workers, ensuring proper passage of the fluids and maintenance of the pipes are an important parameter for the client and end user. This project specifically focuses on pipes used in continuous long transmission of water and other fluids, pipes used in divisional transportation of water. Cracks and water leakage are a major concern for regional and national water authorities, as damages to water pipes are prone due to hindrance and negligence caused from general public, road and transport authority because of their subsequent road maintenance and building. The in-pipe robots consists a mechanism of camera and sensors to identify these crack propagation signatures and return a live feed monitoring with Program based differentiation to a technician for further facilitation of evaluation. Pipe transmission systems deteriorate progressively over time. Pipelines are one of the safest ways to transport water and other liquids. The bigger part of pipeline transmission systems are accessible by existing inspection tools or human intervention but this is limited to the section which are normally at the exterior or joints of the pipe. Corrosion or cracks does not generate largely in this areas, hence the requirement of a smaller inspection device increases. The industry is keen on receiving newer innovations that allows inspection without interrupting the routine operations. It is understood that reliable and accurate information of inspection can only be found only by direct pipe wall access. The robot (PIR) with an adaptable design may help in imbibing with the interior environment of the pipe, especially with respect to the diameter of the pipe, with high maneuverability and ability to operate under hostile conditions. For PIR we are using a six wheeled concept since it is the easiest, most energy efficient and has the potential to travel long range.

1.1 Review of Literature

[1]2010, Manabo ONO, et al., discusses the possibilities of an Earthworm like robot that may be used for inspections of long pipelines. It uses a design similar to that of an Earthworm which uses three rubber bellows as pneumatic actuators. [2]2012, MD Rashid Ashraf, et al., presents the procedures undertaken during the design and development of a small size pipe-in robot for the purpose of boiler head inspection. This robot used the method of visual inspection using a camera probe that shall be later looked in to for the carrying out of the procedure. [3]2012, M F Yousuff, et al., discusses about a robot that can be utilized for the inspection and movement along an air-condition pipeline. It details the different parameters that are to be met while designing such a robot. [4]2012, Dongwoo Lee, et al., discusses about a methodology for navigating a pipeline using two mechanisms known as; Adaptable Quad Arm

Mechanisms (AQAM) and the Swivel Hand Mechanism (SHM). These mechanisms help in bypassing the earlier difficulties faced by in pipe robots in traversing through a pipeline. [6]2014, Majid M Moghadam, et al., presents an apparatus that can be used to traverse a pipeline effectively by using three independent rubber track units that are 120 degree apart symmetrically. The paper also goes into detail with regards to the force analysis that was done on the apparatus. The design phase is clearly mentioned and the application of each part is noted systematically. [9]2018, Sami Salama Hussein Hajjaj, et al., The paper mentions in detail about three possible methods they may use visual inspection using camera probes, assessment of corrosion coupons, or pipe Inspection using in line inspection devices for detecting places of pitting or metal loss. The design is a caterpillar design for in-pipe inspection. [10]2022, Kaned Thung-Od, et al., discussed about is one that can be used in ferromagnetic pipelines that are used to transport Oil, Gas etc. These robots have both vertical and horizontal mobility options. The design is in the shape of a train with three sections. Sensors were provided for measuring various data from within the pipe.

2. PROJECT METHODOLOGY

1. Designing of pipe inspection robot, here the design of pipe inspection robot was implemented using NX6.0, Autodesk Fusion 360 and Autodesk Tinker Hub.

2. Development of pipe inspection robot, here Fusion 360 and Tinker hub was used for the simulation before actual building of the prototype.

3. Fabrication of Model by means of manufacturing equipments. This robot consists of six legged wall caterpillars.

4. Creation of a camera web server in accordance with arduino microprocessor and testing of initiation of feedback from ESP32 Camera 1 & Camera 2.

5. Creation and testing of an open CV based python (PY Charm) interfaced program that will render the live feedback from ESP32 (1&2) and will use the contours from the library to differentiate the cracks from the feedback.

6. Assemble all the required fabricated parts. The complete assembly of six legs, six wheels, translation element between the first and second enclosure as well as centre frame was made using Galvanized Iron fixtures.

7. Relay Motors and Driver of L298N Specification was used in facilitating generic backward and forward movement of the robot.

8. Fixing of ESP32 cameras, microprocessor and Relay motors. We will be selecting 6-DC motors.

9. Making it to crawl inside a pipe by means of remote controller to capture the video image in a screen (Live video feedback). Evaluation of Visual Odometry and determining the scaling factor required for the program to convert multiple pixels from image to distance by cascading.

10. Use of HAAR Cascading to feed in the crack propagation simulative images and return the feedback.

3. DESIGN OF PIPE INSPECTION ROBOT

3.1 Design Constraints

Size and Dimension: The robot should have a compact size and dimensions to fit and navigate through pipes of varying diameters and configurations.

Maneuverability: The robot should be able to navigate through tight spaces, bends, and obstacles within the pipe network.

Robustness and Durability: The robot should be built with materials and components that can withstand harsh and corrosive environments found in pipes, ensuring long-term reliability.

Power and Energy Efficiency: The robot should be designed to operate using efficient power sources, such as batteries, to provide sufficient runtime for inspections without frequent recharging or replacement.

Sensing and Detection Capabilities: The robot should incorporate sensors and detection systems capable of accurately identifying cracks, leaks, blockages, and other pipe abnormalities.

Communication and Data Transmission: The robot should be equipped with reliable communication systems to transmit inspection data, video feedback, and control signals in real-time to operators or monitoring systems outside the pipes.

3.2 Design Specifications

To complete the project it is very crucial to have the specifications before the start of the initial development of the project. This will ensure the conformity with the set of standards. Some of the specifications are:-

1. Video streaming 1440p
2. Vertical movement within the pipe (Max. crawl of 10km/h)
3. Wireless control (up to 1200 m)
4. Galvanized Iron body with aluminium endowments
5. 48 Megapixel camera with wide angle approach sensor for distance ranging
6. Remote Control using pycharm integration and remote XY for elongated control of the robot

7. Rechargeable Power suit, which is waterproof to run the movement.

3.3 Design of Parts and Assembly

The pipe inspection robot consists of a mechanism of six-legged 120 – degree apart rotor based traction tire set-up which is docked into multiple transition element which is welded into the main central frame, additional push element with spring loaded risers are inserted to eliminate the change arising within external walls of pipe and the tire sub-clearance.

Dimensions are L=200mm, H=80mm diameter, and inner diameter = 17mm, Thickness of extended part= 8mm. there are total of six extended parts on center frame on 70 mm from one side and another 3 on 170mm from one end and have a hole of diameter 6mm.

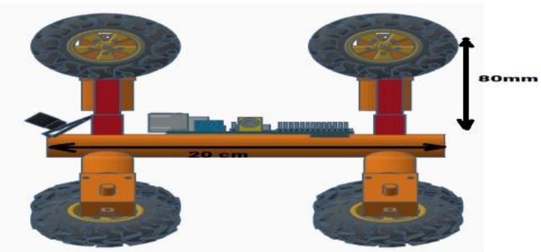

Fig -1: Central Frame of the Robot using Autodesk Fusion 360

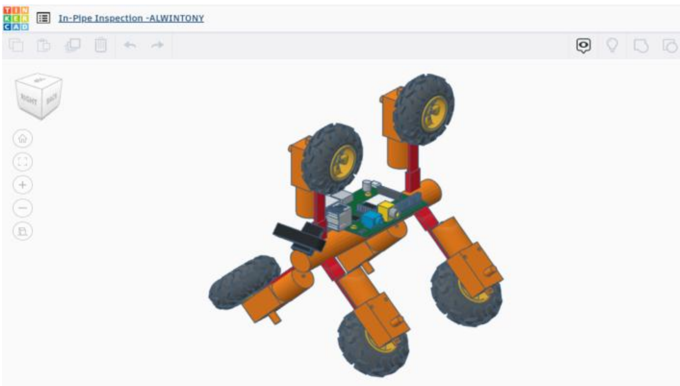

Fig -2: Overall Dislocated assembly of Design (Fusion 360)

3.4 Power Calculation

3.4.1 Assumptions

1. The force required for movement is equal to the weight of the robot.
2. The weight of the robot is 0.716kg.
3. Surface of the pipe is very smooth.
4. Maximum speed of the robot is 1m/s.
5. The efficiency is 80%.

3.4.2 Calculations

If the speed of the robot is 1 m/second and the robot weighs 0.716 kg, we can calculate the power required for the motor using the formula:
Power = (Force × Velocity) / Efficiency

First, let's calculate the force required for movement. Since the robot is on a flat surface, we can assume the force required is equal to the weight of the robot:

Force = Weight = 0.716 kg × 9.8 m/s^2 (acceleration due to gravity) = 6.8 N

Assuming an efficiency of 80% or 0.8, we can substitute these values into the formula:

Power = (6.8 N × 1 m/s) / 0.8 = 8.5 W

Considering the factor of safety we take 10 watts DC geared motor.

4. DETAILED DESCRIPTION OF PARTS USED

4.1 Arduino Uno:

Microcontroller: The Arduino Uno is built around the ATmega328P microcontroller, which is an 8-bit AVR microcontroller.

Clock Speed: The ATmega328P runs at a clock speed of 16 MHz

Digital I/O Pins: The Arduino Uno has a total of 14 digital input/output pins. Among them, 6 can be used as PWM (Pulse Width Modulation) outputs.

Analog Inputs: It has 6 analog input pins, labeled A0 to A5, which can also function as digital I/O pins.

Programming: The Arduino Uno can be programmed using the Arduino programming language, which is a simplified version of C++.

Power Supply: The Uno can be powered using a USB connection, a DC power jack (7-12V), or by supplying 5V directly to the 5V pin or 3.3V to the 3.3V pin.

Memory: The ATmega328P on the Uno has 32KB of flash memory for program storage, 2KB of SRAM, and 1KB of EEPROM for data storage.

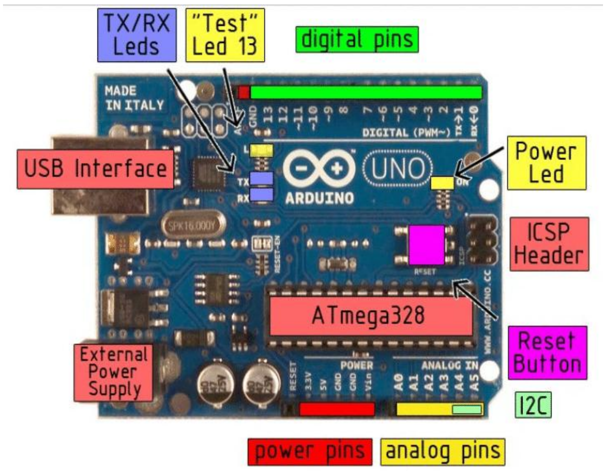

Fig- 3:- Graphical Schematic of Arduino UNO Microcontroller

The Arduino Uno operates by following a specific sequence of operations. It starts with a power supply provided through a USB connection, DC power jack, or direct voltage input. Once powered, the microcontroller runs a boot loader program stored in its memory, allowing for easy uploading of sketches. After initialization, the Arduino Uno executes the setup () function where variables are initialized and pin modes are configured. The main part of the program is the loop () function, which is repeatedly executed. Here, you can implement project functionality, read sensor inputs, and control actuators. The board supports interrupts for immediate response to external events.

4.2 ESP32 CAM

The ESP32CAM is a development board that integrates the ESP32-S module and a camera sensor. It combines the power of the ESP32 microcontroller and the capability of capturing images and video using a camera module. Here are some details about the ESP32CAM:ESP32 Microcontroller: The ESP32CAM is based on the ESP32-WROOM-32 module, which features a dual-core ESP32 microcontroller. The ESP32 is a powerful and versatile microcontroller with built-in Wi-Fi and Bluetooth capabilities. Camera Sensor: The ESP32CAM includes a camera sensor module that allows for capturing images and video. The camera module is typically based on the OV2640 or OV7670 sensor, offering different resolutions and capabilities. Flash Memory: The ESP32CAM board is equipped with 4MB of flash memory, which is used for storing the firmware and program code.

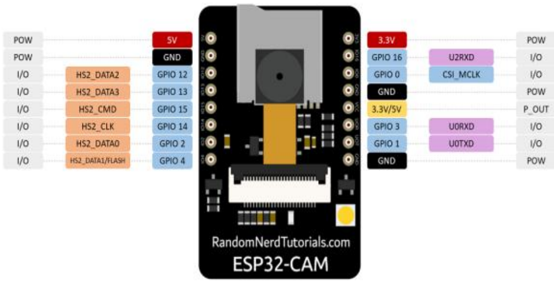

Fig 4:- Graphical Schematic of ESP32 Camera and its interfaces

4.3 ESP 8266 Wifi Module

The ESP8266 Wi-Fi module is a popular and affordable Wi-Fi module that has gained significant popularity among developers and hobbyists for its capability to add Wi-Fi connectivity to various electronic projects. Here are some details about the ESP8266 Wi-Fi module: Microcontroller: The ESP8266 module is itself a self-contained microcontroller system-on-chip (SoC) with integrated Wi-Fi capabilities. It features a 32-bit ten silica Xtensa LX106 microcontroller unit (MCU).Wi-Fi Connectivity: The ESP8266 module supports Wi-Fi connectivity, allowing devices to connect to local networks and the internet. It supports the 802.11b/g/n Wi-Fi standards, providing reliable wireless communication. Flash Memory: The ESP8266 module typically comes with onboard flash memory for storing firmware, program code, and data. The available flash memory varies across different models and versions of the module. GPIO Pins: The ESP8266 module provides a number of general-purpose input/output (GPIO) pins that can be used for digital input/output, analog input, and various communication protocols such as UART, I2C, and SPI. The exact number of GPIO pins varies depending on the specific module variant.

4.4 Channel Relay Module

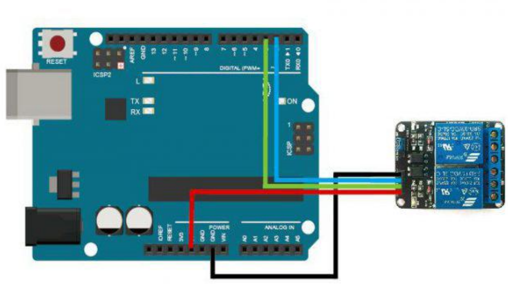

Fig 5:- Connection between arduino microcontroller and Channel relay module

A 5V 2-channel relay module is an electronic module that allows you to control high-power devices using low-power Signals from a microcontroller or other control circuit.

Here are some details about a typical 5V 2-channel relay module: Number of Channels: A 5V 2-channel relay module has two separate relay channels. Each channel can control a different electrical load independently. Relay Type: The relays used in the module are usually electromechanical relays. These relays consist of a coil and one or more sets of contacts that can switch high currents or voltages. Input Voltage: The module is designed to be driven by a 5V DC power supply. It is compatible with most microcontrollers and development boards that operate at 5V logic levels.

Control Signal: The module typically uses a low-level control signal (e.g., logic low) to activate the relays. This control signal is provided by a microcontroller or any other digital output device. Relay Contacts: Each relay channel usually has a common (COM), normally opens (NO), and normally closed (NC) contact. The COM contact is connected to the power supply or the load, while the NO and NC contacts are used to switch the load on and off. Opt-isolation: Some relay modules include opt-isolation circuitry, which provides electrical isolation between the control circuit and the high-power side (relay contacts). This helps protect the control circuit from voltage spikes or disturbances caused by the switching of inductive loads.

5. OVERVIEW OF PROGRAMS CREATED FOR ANALYSIS

5.1 RemoteXY Code for Movement of Robot

```
#include <RemoteXY>
#define REMOTEXY_SERIAL Serial
#define REMOTEXY_SERIAL_SPEED 115200
#define REMOTEXY_WIFI_SSID "RemoteXY"
#define REMOTEXY_WIFI_PASSWORD "12345678"
#define REMOTEXY_SERVER_PORT 6377
#pram pack (push, 1)
uint8_t RemoteXY_CONF[] =   // 44 bytes
 { 255,2,0,0,0,37,0,16,31,1,10,48,24,24,15,15,4,26,31,79,
 78,0,31,79,70,70,0,10,48,24,75,15,15,4,26,31,79,78,0,31,
 79, 70,70,0 };
struct {
 uint8_t pushSwitch_1; // =1 if state is ON, else =0
 uint8_t pushSwitch_2; // =1 if state is ON, else =0
 uint8_t connect_flag;  // =1 if wire connected, else =0
} RemoteXY;
#pragma pack (pop)
#define PIN_PUSHSWITCH_1 4
#define PIN_PUSHSWITCH_2 5
void setup()
{
 RemoteXY_Init ();
 pinMode (PIN_PUSHSWITCH_1, OUTPUT);
 pinMode (PIN_PUSHSWITCH_2, OUTPUT);
}
Void loop ()
{
 RemoteXY_Handler ();
 digitalWrite(PIN_PUSHSWITCH_1,
(RemoteXY.pushSwitch_1==0)?LOW:HIGH);
 digitalWrite(PIN_PUSHSWITCH_2,
(RemoteXY.pushSwitch_2==0)?LOW:HIGH);
}
```

5.2 OpenCv Python Program for Crack Detection

```
import cv2
import numpy as np
import urllib.request
stream_url = 'http:/192.168.12.269/video'
threshold1 = 30
threshold2 = 100
def process_frame(frame):
   gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
      edges = cv2.Canny(gray, threshold1, threshold2)
   cv2.imshow('Crack Detection', edges)
def main():
   stream = urllib.request.urlopen(stream_url)
   bytes = bytes()
   while True:
      # Read a frame from the video stream
      bytes += stream.read(1024)
      a = bytes.find(b'\xff\xd8')
      b = bytes.find(b'\xff\xd9')
      if a != -1 and b != -1:
         jpg = bytes[a:b+2]
         bytes = bytes[b+2:]
               frame cv2.imdecode(np.frombuffer(jpg,
dtype=np.uint8), cv2.IMREAD_COLOR)
  process_frame(frame)
            if cv2.waitKey(1) & 0xFF == ord('q'):
         break
  stream.release()
   cv2.destroyAllWindows()
if __name__ == '__main__':
   main()
```

5.3 HAAR Cascading Program for distance measurement using visual odometry

```
url = http://102.167.652.29:81/stream
cap = cv2.VideoCapture(url)
prev_frame = None
dist_traveled_pixels = 0
dist_traveled_centimeters = 0
prev_points = None
scaling_factor = 0.1235
lk_params = dict(winSize=(15, 15), maxLevel=2,
criteria=(cv2.TERM_CRITERIA_EPS |
cv2.TERM_CRITERIA_COUNT, 10, 0.03))
displacement_threshold = 2.0  # Adjust this value based on
the expected motion magnitude
while True:
ret, frame = cap.read()
if not ret:
break
gray = cv2.cvColor(frame, cv2.COLOR_BGR2GRAY)
```

```
if prev_frame is not None:
if prev_points is not None:
next_points, status, _ =
cv2.calcOpticalFlowPyrLK(prev_frame, gray, prev_points,
None, **lk_params)
good_old = prev_points[status == 1]
good_new = next_points[status == 1]
if len(good_old) > 0 and len(good_new) > 0:
displacements = good_new - good_old
displacement_magnitudes = np.linalg.norm(displacements,
axis=1)
average_displacement =
np.mean(displacement_magnitudes)
if average_displacement > displacement_threshold:
dist_traveled_pixels += average_displacement
dist_traveled_centimeters = dist_traveled_pixels *
scaling_factor
prev_points = cv2.goodFeaturesToTrack(gray, 100, 0.3, 7)
prev_frame = gray.copy()t
text = f"Distance Traveled: {dist_traveled_centimeters:.2f}
centimeters"
cv2.putText(frame, text, (10, frame.shape[0] - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1,
cv2.LINE_AA)
cv2.imshow("Frame", frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
break
cap.release()
cv2.destroyAllWindows()
```

6 DISTANCE MEASUREMENT USING VISUAL ODOMETRY

Visual odometry, combined with OpenCv and Python programming, offers a powerful approach for detecting the location of a pipe inspection robot within a pipe network. Visual odometry relies on the analysis of visual data captured by the robot's onboard camera to estimate its position and orientation within the pipe. By leveraging OpenCv, a versatile computer vision library, and Python programming, the robot can extract and process visual features, track their movements, and calculate its location. The process begins with feature detection, where OpenCv feature extraction algorithms identify distinct visual features such as corners or edges within the captured images. These features act as reference points for subsequent tracking and location estimation. The features are then matched across consecutive frames, utilizing OpenCv feature matching algorithms, to establish correspondences and track their motion. Python programming is utilized to implement the visual odometry pipeline. The tracked features are processed using various techniques such as motion estimation, pose estimation, and triangulation. To ensure accurate location detection; camera calibration is a critical step. By accurately calibrating the camera, distortion and other camera-specific factors are taken into account, improving the accuracy of the location estimation. Real-time location detection is achieved by implementing the visual odometry pipeline efficiently. Python's versatility and OpenCv optimized functions allow for fast processing and tracking of visual features, enabling the robot to estimate its location in real-time during the inspection process. It is important to note that visual odometry for pipe inspection robots using OpenCv and Python has its limitations. Factors such as lighting conditions, Occlusions and dynamic environments within the pipes can impact the accuracy of feature detection and tracking. Therefore, careful consideration must be given to address these challenges and develop robust algorithms that can handle various pipe inspection scenarios.

7 CRACK DETECTION AND RESULTS

The pipeline undergoes testing by intentionally introducing different types of cracks with varying sizes and surface areas. These cracks include major, minor, and elongated ones. The propagation of these cracks can be a significant cause of intermittent pipeline failures, leading to disruptions in product transmission. To effectively inspect the pipe's interior, a pipe inspection robot equipped with an ESP-32 Camera is utilized. It captures detailed footage and transmits real-time feedback via a live IP address. This IP address is then utilized in a Python program that invokes OpenCv for further analysis. OpenCv accesses the live video feedback and processes it to identify irregular shapes and contours. OpenCv enhances the analysis by applying a mask to differentiate the crack from the rest of the system. This masking technique helps in isolating and highlighting the crack, making it easily identifiable. The crack is distinctly marked with a red contour, allowing personnel to effortlessly determine its position within the extensive pipeline network. By employing this comprehensive inspection process, pipeline operators can proactively address cracks and potential failures, ensuring the uninterrupted transmission of products and maintaining the integrity of the pipeline system.

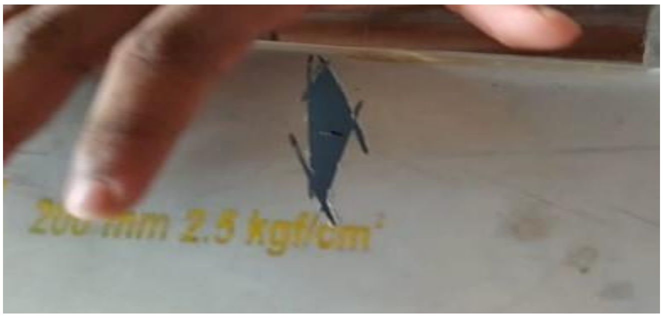

Fig 6: Crack induced on pipe with marked edges to highlight the irregularity

Fig 7:- Robot inspecting the interior walls of the Test Pipe

Fig 8:- Crack detection using OpenCv (from test cracks)

8 CONCLUSIONS

In conclusion, the project work on a pipeline inspection robot utilizing Arduino Uno, RemoteXY, relay module, DC motors, RemoteXY app for movement control, ESP32-CAM, and an OpenCv Python program for crack detection offers a comprehensive solution for efficient and accurate pipeline inspection. The integration of Arduino Uno provides a reliable microcontroller platform for controlling the robot's movements and coordinating various components. The RemoteXY library enables easy communication between the robot and the RemoteXY app, allowing intuitive control of the robot's forward and backward movement. The relay module plays a crucial role in controlling the DC motors, facilitating the robot's locomotion. The inclusion of the ESP32-CAM module enhances the robot's capabilities by enabling live video streaming from the pipeline. The OpenCv Python program leverages the power of computer vision and image processing techniques, specifically canny edge detection, to detect cracks in real-time. This provides valuable insights into the condition of the pipeline and helps identify potential issues for timely maintenance. By combining these technologies, the pipeline inspection robot offers several advantages. It enhances efficiency by automating the inspection process, reducing the need for manual intervention and saving time. The accuracy of crack detection is improved through the utilization of computer vision algorithms, providing detailed visual information for analysis and decision-making , For instance, implementing autonomous navigation, advanced sensing technologies, and data analytics capabilities can further enhance the robot's performance and provide more comprehensive insights into the pipeline's condition. The pipeline inspection robot project utilizing Arduino Uno, RemoteXY, relay module, DC motors, RemoteXY app, ESP32-CAM, and OpenCv Python program showcases a practical and effective solution for pipeline inspection. It combines hardware, software, and computer vision techniques to improve efficiency, accuracy, and safety in pipeline maintenance and management. The project serves as a foundation for further advancements and research in the field of pipeline inspection robotics.

REFERENCES

[1]M.F. Yusoff, B.S.K.K. Ibrahim, H. Hamzah, H.A. Kadir Development of Air Conditional Route Wireless Inspection Robot International Symposium on Robotics and Intelligent Sensors (2012), pp. 874-880 (IRIS 2012)

[2] Md Raziq Asyraf Md Zin, Khairul Salleh Mohamed Sahari, Juniza Md Saad, Adzly Anuar, and Abd Talip Zulkarnain, Development of a Low Cost Small Sized In-Pipe RobotInternational Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012), pp 1469-1475.

[3] Dongwoo Lee, Jungwan Park, Dongjun Hyun, GyungHwan Yook, Hyun-seok Yang Novel mechanisms and simple locomotion strategies for an in-pipe robot that can inspect various pipe type Mechanism and Machine Theory, 56 (2012), pp. 52-68

[4] Majid M. Moghaddam, Mohammadreza Arbabtafi, A. Hadi In-pipe Inspection Crawler Adaptable to The Pipe Interior Diameter International Journal of Robotics and Automation, 26 (2) (2011), pp. 135-145

[5] Megumi Ikeuchi, Taro Nakamura, and Dai Matsubara, Development of an In-pipe Inspection Robot for Narrow Pipes and Elbows using Pneumatic Artificial Muscles, 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems October 7-12, 2012. Vilamoura, Algarve, Portugal, pp 926-931.

[6] Yazdekhasti, S.; Piratla, K.R.; Sorber, J.; Atamturktur, S.; Khan, A.; Shukla, H. Sustainability analysis of a leakage-monitoring technique for water pipeline networks. *J. Pipeline Syst. Eng. Pract.* 2020, *11*, 04019052

[7]Rodríguez-Olivares, N.A.; Cruz-Cruz, J.V.; Gómez-Hernández, A.; Hernández-Alvarado, R.; Nava-Balanzar, L.; Salgado-Jiménez, T.; Soto-Cajiga, J.A. Improvement of

ultrasonic pulse generator for automatic pipeline inspection. *Sensors* 2018, *18*, 2950

[8] Henry, R.; Chablat, D.; Porez, M.; Boyer, F.; Kanaan, D. Multi-objective design optimization of the leg mechanism for a piping inspection robot. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Buffalo, NY, USA, 17 August 2014; p. V05AT08A001.

[9] Singh, W.S.; Rao, B.; Sasi, B.; Vaidyanathan, S.; Jayakumar, T.; Raj, B. Giant Magneto-Resistive (GMR) Sensors for Non-Destructive Detection of Magnetic Flux Leakage from Sub-Surface Defects in Steels. In Proceedings of the International Conference on Sensors and Related Networks (SENNET-2009), VIT University, Vellore, India, 1 December 2009;

[10] Kreutzbruck, M.; Neubauer, A.; Pelkner, M.; Reimund, V. Adapted gmr array used in magnetic flux leakage inspection. In Proceedings of the 18th World Conference on Nondestructive Testing, Durban, South Africa, 16–20 April 2012;