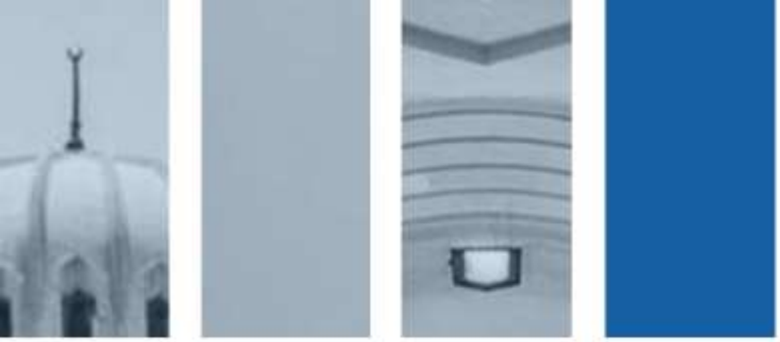


RESEARCH & PROJECT SUBMISSIONS

Program:

Course Code: MCT382
Course Name: Design of Mechatronics Systems 2

Examination Committee

XXXX
XXXX
XXXX
XXXX

Ain Shams University
Faculty of Engineering
International Credit Hours Engineering Programs (I-CHEP)

Student Personal Information for Group Work

Student Names:	Student Codes:
Mohammed yasser Mohammed	17P8214
Momen Mohsen	17P8115
Ahmed Mohammed Nasr	17P8211

Plagiarism Statement

I certify that this assignment / report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they are books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment / report has not been previously been submitted for assessment for another course. I certify that I have not copied in part or whole or otherwise plagiarized the work of other students and / or persons.

Signature/Student Name: **Mohammed Yaser Mohammed**

Momen Mohsen

Ahmed Mohammed Nasr

Date: 5/5/2021

Abstract

This report provides basic and advanced operating instructions for the **RC-XD** Robot. The information in the report will explain the mechanical, electrical and software designs used in this robot, Our vision is to solve some difficulties that pipe inspection companies face, to save the time consumed by manual inspecting procedures and to maintain a safe environment for the employees engaged in this field, Our mission is to improve the quality of the inspecting field in Egypt, to save their time, money and effort and to make robots easily available in the Egyptian market, **RC-XD** is a pipe inspection robot, it's main purpose is to inspect the corrosion, cracks , and rust inside the pipes and get its exact position and geometry in the pipe and report it.

Contents

List of Figures

Introduction

Pipeline systems deteriorate progressively over time through various means, so this product aims to create a pipeline inspection robot that has an adaptable structure to the pipe diameter, detects any cracks and cheap at the same time. Pipeline inspection robots are designed to act in inaccessible environment and also to remove the human factor from labour intensive or dangerous work environments. Our challenge is to make this robot adaptable to pipe diameters varying from **450mm** to **700mm** that can detect any inner cracks in a pipe.

1.Market Research

To study the market in Egypt we relied on two sources internet data and face to face meetings with company owners and representatives.

1.1. Statistics and surveys

Egyptian market has more than 80 companies in the inspection field excluding petroleum companies, less than 10% of them use robots in their field mainly because it’s too expensive to get a robot from abroad and it’s difficult to maintain it in Egypt.

More than 40% of yearly jobs deal with pipelines with more than 10 meters in length with different diameters ranging from 30cm to 1.2m

A normal inspection job takes 3-4 days in average and average small companies take 80+ jobs/year.

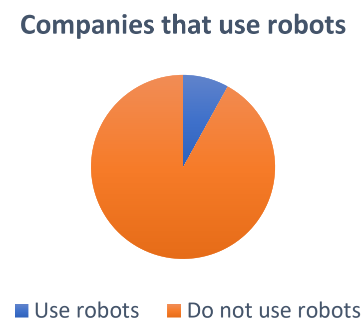


Figure 1 Companies that use robots relative to all

A technicians’ minimum wage per day ranges from 150 L.E up to 600 L.E, excluding other outlays, so approximately a job with only one technician costs nearly about 2000 L.E

1.2. Target

Inspection companies start a job with a site visit to identify what is the required job and take some measurements, then start the solving step which is to brainstorm and think how can they perform this job and what they need, then a preparation step where they gather the required tools and/or operators required, then they start the inspection.

Based on further surveys and meetings, the main target is to eliminate an exhausting and time and money consuming step which is the solving step and eliminating this step will save money, effort and time to a company.

Using these surveys and statistics we concluded how we will shape our robot according to our customers’ needs.

1.3. Value Proposition Canvas

Figure 2 Value Proposition canvass

1.3.1. Gains

- Safe inspection
- Customer Satisfaction

1.3.2. Gain Creators

- Less workers
- Record the inspection process
- More professional method

1.3.3. Pains

- Dealing with hard and extreme cases
- Time and money consuming process
- Unprofessional technicians' faults

1.3.4. Pain Relievers

- Time efficient
- Less effort
- Adjustable with different diameters
- Crack shape record

1.3.5. Customer Jobs

- Inspection of pipes in oil and gas field
- Case study

1.3.6. Products and Services

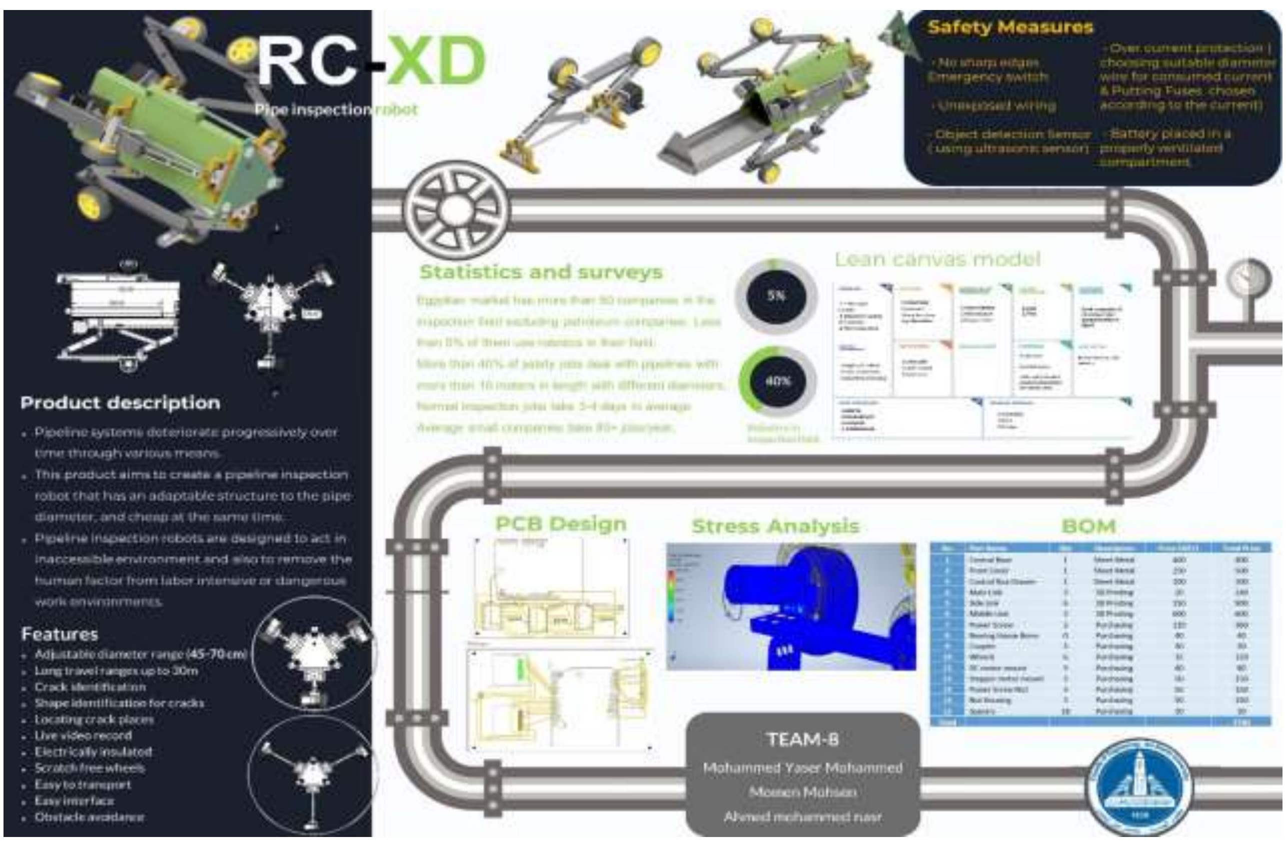
- Semi-autonomous
- Adjustable
- Scratch free material
- After sales

1.4. Lean Canvas Model

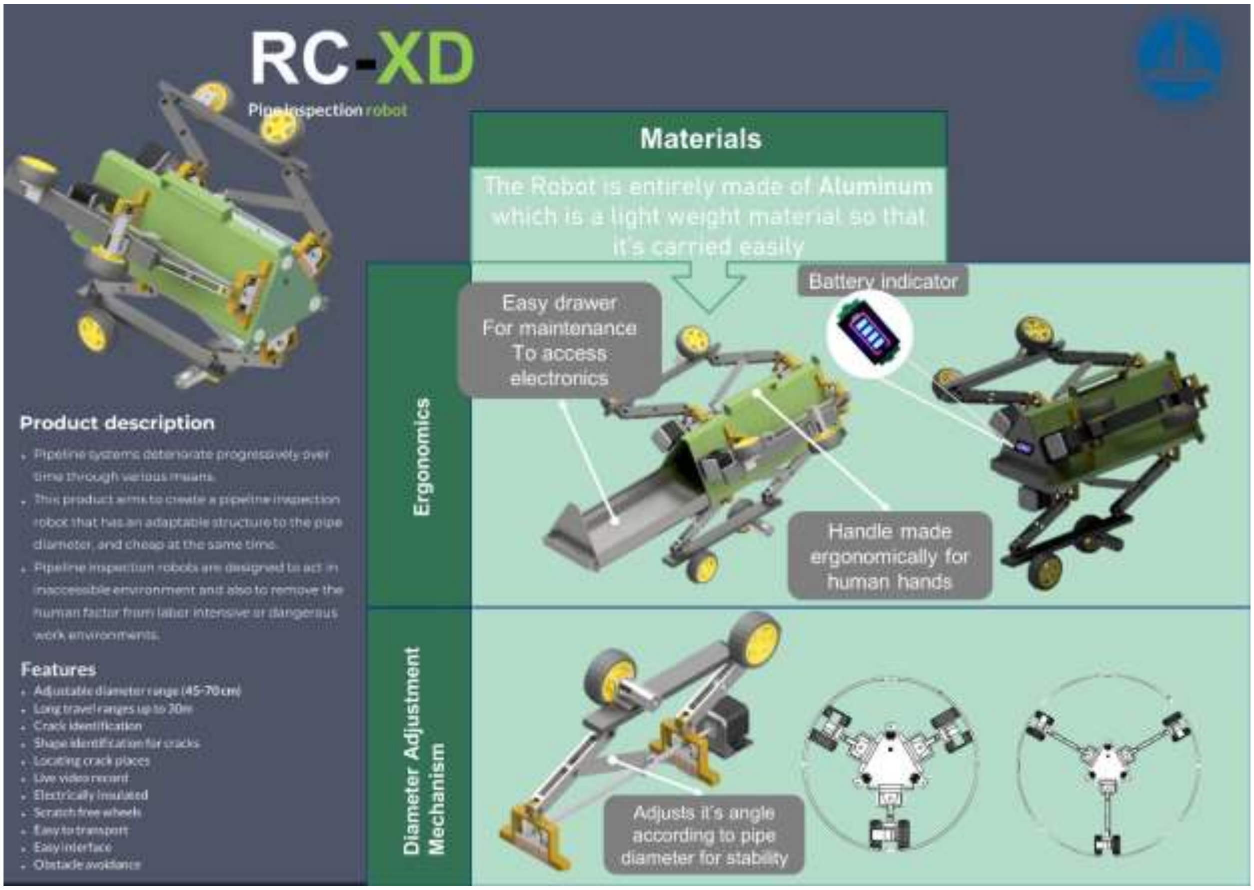
		Designed for: RC-XD	**Designed by:** Team 8	
Problem 10m+ Length Pipes Diameters variety Insulation Time consuming	**Solution** Adjustable Low cost Easy interface	**Unique Value Proposition** Easy interface Maintenance After Sales Support	**Unfair Advantage** Cost Time	**Customer Segments** Small Gas and Oil inspection companies
Existing Alternatives High cost robots Time and money consuming processes	**Key Metrics** Units sold Units rented Contracts	**High-Level Concept**	**Channels** Sales Team Conferences Company site	**Early Adopters** Below 30 Years old owners
Cost Structure Salaries Development Materials Maintenance		**Revenue Structure** Contracts Rent Buying		

Figure 3 Lean Canvas Model

1.5. Academic A1 Poster



1.6. Product Design A1 Poster



1.5. Brochure


PIPELINE INSPECTOR
RC-XD
ENOUGH SEARCHING
START SOLVING
ORDER NOW
PIPE INSPECTION
inspection refers to the process of examining pipes using either visual inspection or more complicated techniques using complicated ultrasonic based sensors .
At first for the large diameter pipes were being inspected by an technician who enters the pipe with a probe carrying the sensor to inspect the pipe from inside.
OFFICE
51 22nd Street CAI 11784
MAADI
PHONE
020145970
EMAIL
RC-XD@CO-OP.com
Locating the Cracks
Live stream to the process
Access to any pipe between 45-70cm
MISSION
To protect the labors from risk of entering a pipe , save time and money in the process of inspection
VISION
To help startup companies grow faster do more inspection jobs effortlessly with robots they can have their hands on
FEATURES
• Blockage Avoidance If the pipe is blocked ultrasonic sensor detects the blockage and the robot retracts its self from the pipe.
• Crack and Edges Detection
• Crack and Edges Measurement
• Saving the position of the crack in the name of it's photo
• Infinitely Diameter adjustment of the robot to fit any pipe diameter between 0.28m and 0.651 m.
SERVICES
• After sales technical support .
• 1 year guarantee against manufacturing problems .
• Periodic software updates to enhance productivity and accuracy
• Free tutorials on how to handle the robot and interact with it's UI.

2. Product Description

RC-XD is a semi-autonomous robot that can travel in long pipes with adjustable diameter, scan for inner cracks, locates their locations and identifies the shape of this crack, by setting the pipe diameter, length and the desired travel speed, the robot adjust itself to the pipe diameter and once placed in the pipe it starts to move with the desired speed and begins to scan for any cracks.

RC-XD is a 6 wheeled robot actuated by the four lower wheels, made generally of sheetmetal and do not need an operator to operate it.

2.1. Features

- Adjustable diameter range (**45-70 cm**)
- Long travel ranges up to 30m
- Crack identification
- Shape identification for cracks
- Locating crack places
- Live video record
- Electrically insulated
- Scratch free wheels and safe to pipes insulation materials
- Easy to transport
- Easy interface
- Obstacle avoidance

2.2. Environmental Ratings

RC-XD robot is a durable and rugged robot that can be used in harsh environments, well insulated and safe to meet any work site rules.

2.3. Main Components

The **RC-XD** front side contain an ultrasonic sensor behind it the control box which contains the Raspberry Pi, the Arduino uno Launchpad, Lithium ion battery and the PCB, after that we have the main body attached to 3 wheel assemblies with 120^0 angle between each other -to ensure stability in pipes- each has a mechanism that controls the diameter adjustment. In each mechanism there are 2 wheels, one of those wheels are actuated using a dc motor in the front only, the rest wheels are free to move.

2.4. HMI

The user will see the output data as a video feed that can identify any crack and display its dimensions and also the robot will store images of the cracks named with the distance measured and the angle at which the crack was identified

3.Mechanical Design

3.1. RC-XD Original

Figure 4 Mechanical design view1

Figure 4 Mechanical design view2

3.2. RC-XD Prototype

Figure 4 Mechanical design view1

Figure 5 Mechanical design view2

Figure 6 Mechanical design view3

3.3. Overall Dimensions

RC-XD’s overall length is approximately 70cm in minimum diameter and 40cm in max diameter.

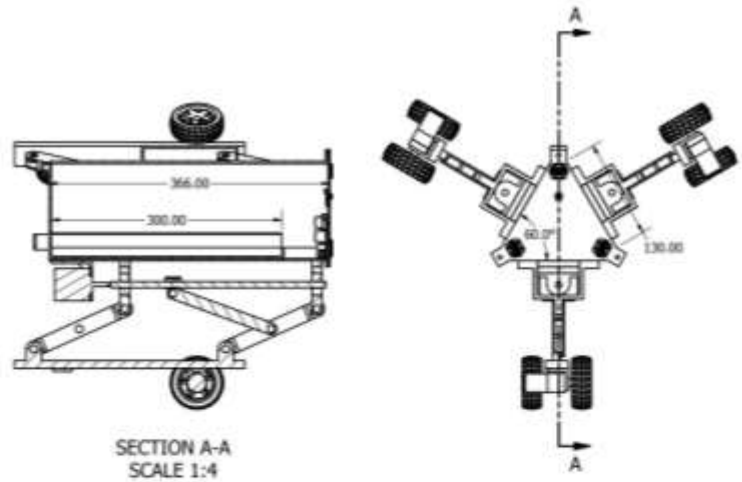


Figure 6 Overall dimensions

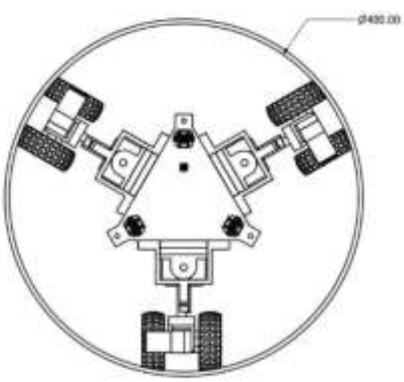


Figure 8 Minimum Diameter

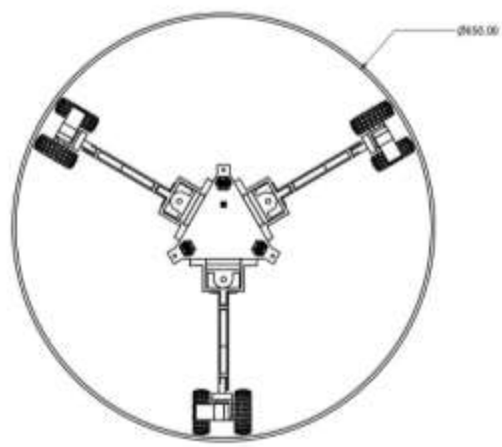


Figure 7 Maximum Diameter

3.4. Control Box

Control box contains the camera and a servo motor, Power screw actuator, Raspberry PI and the PCB.

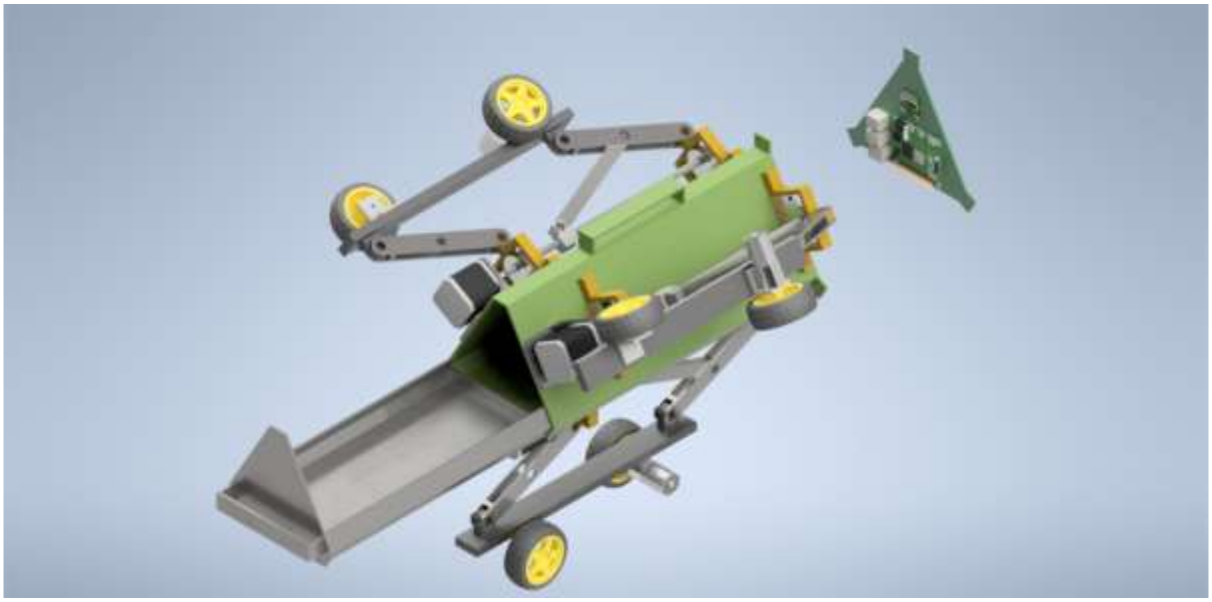


Figure 9 Control Box

3.4. Control Box in Prototype

Control box contains the camera and a servo motor, Power screw actuator, Raspberry PI and the PCB.

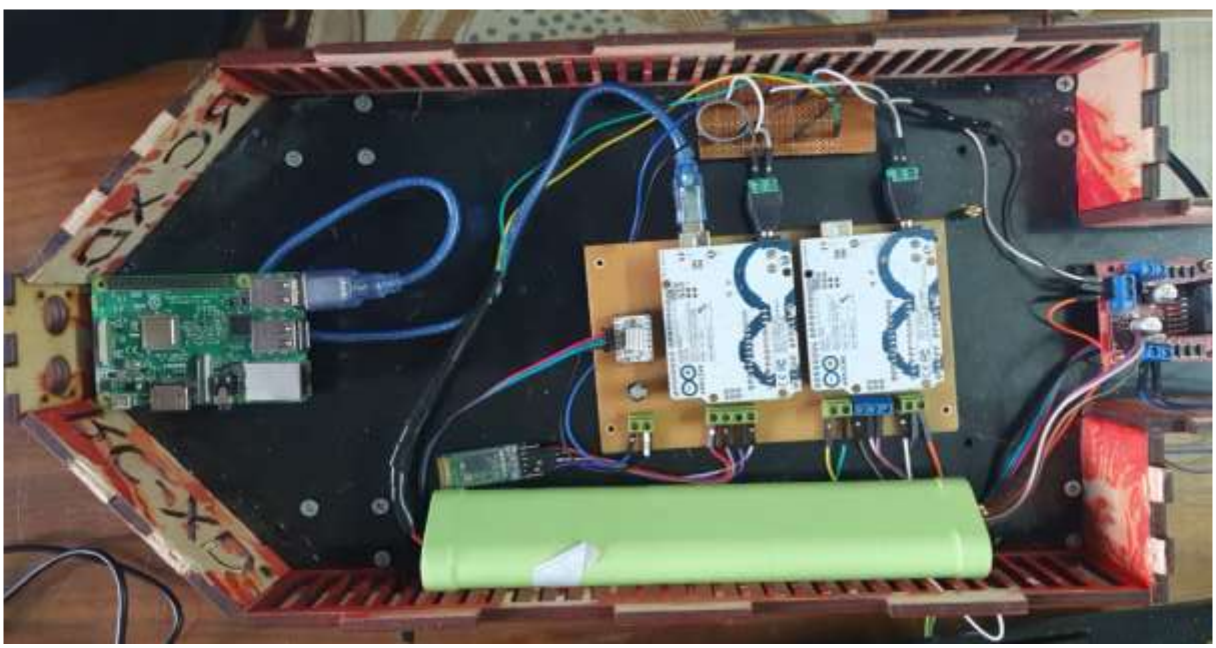


Figure 10 Control Box

3.5. Diameter Adjustment Mechanism

In a nutshell it's a Four-Bar Mechanism actuated by a Four-Bar Mechanism. The power screw actuator receives a signal with the desired diameter, then the power screw rotates with the set angle moving the bushes a certain distance, and the bushed moves a triangular plate attached to it that moves relative to a fixed plate bolted to the base.

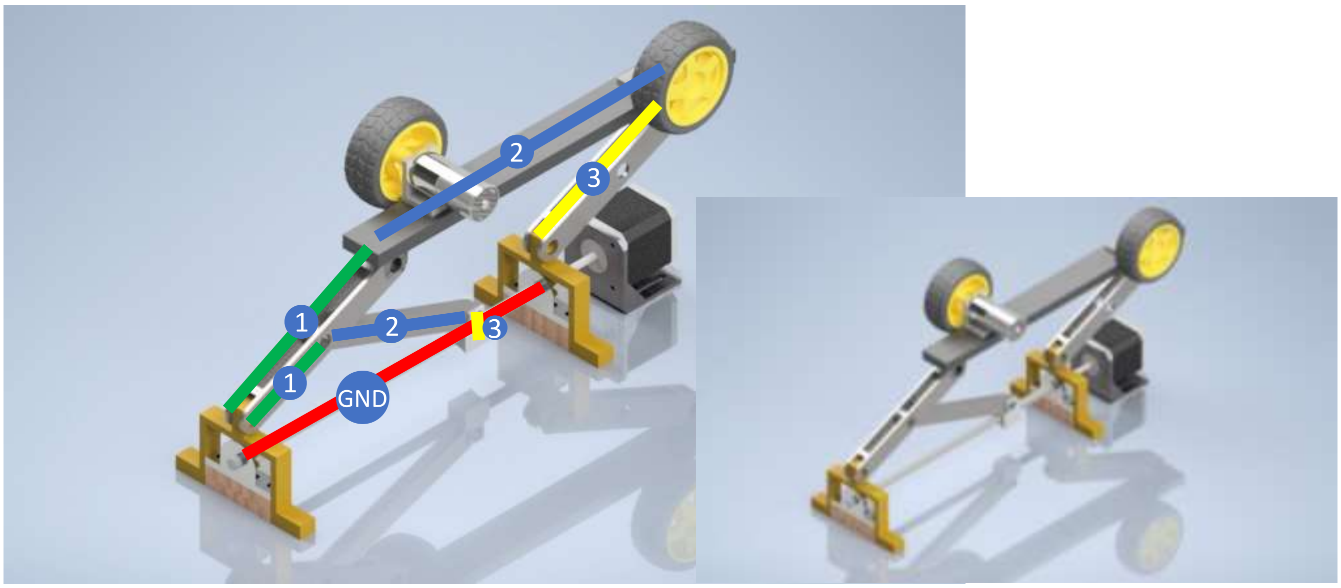


Figure 13 Diameter Adjustment Mechanism

Attached to the moving plate three links connected to the arms holding the wheels so moving the power screw nut changes the angle of the hinged arms changing the central distance of the wheel.

3.6. Stress Analysis

Assuming Worst Case:

- Robot's weight is 10 kg
- Only two wheel carries the whole weight
- Motor Radial Load = 60 N.m (From datasheet)

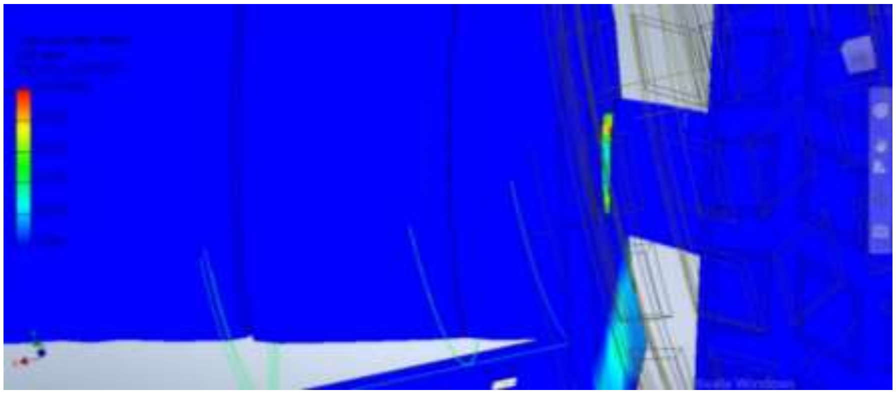


Figure 15 Stress Analysis

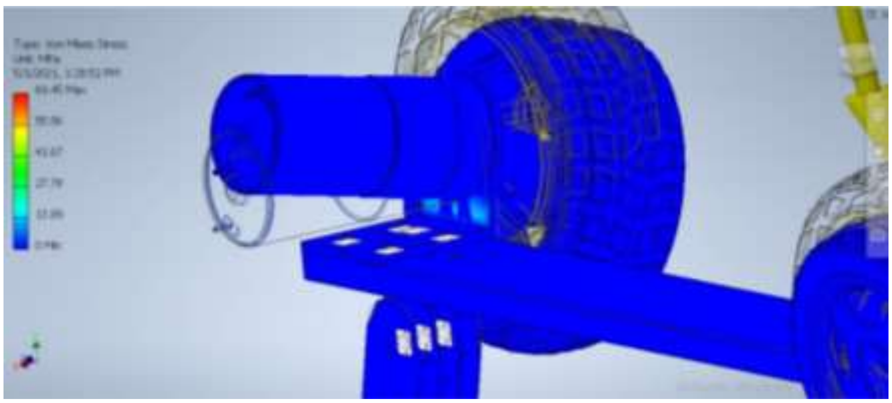


Figure 14 Stress Analysis

3.7. Mechanical BOM

No.	Part Name	Qty	Description	Price EGP/1	Total Price
1	Central Base	1	Sheet Metal	400	400
2	Front Cover	1	Sheet Metal	250	500
3	Control Box Drawer	1	Sheet Metal	100	200
4	Main Link	3	3D Printing	20	240
5	Side Link	6	3D Printing	150	900
6	Middle Link	3	3D Printing	600	600
7	Power Screw	3	Purchasing	120	360
8	Bearing House 8mm	0	Purchasing	40	40
9	Coupler	3	Purchasing	30	30
10	Wheels	6	Purchasing	15	120
11	DC motor mount	3	Purchasing	40	80
12	Stepper motor mount	3	Purchasing	50	150
13	Power Screw Nut	3	Purchasing	50	150
14	Nut Housing	3	Purchasing	50	150
15	Spacers	18	Purchasing	30	30
Total					**3760**

Figure 11 Mechanical BOM

4. Electrical Design

Electrical Schematics is designed to fit the operation principles determined and depends on the actuator sizing that was done will be mentioned later, and the selected sensors, so the battery was roughly estimated. And the calculations determined that a 12V battery of 6000mAh is required.

So, to limit the size and weight we decided to use Lithium ions batteries and its characteristics also suits our needs.

4.1. Sensors Selection

RC-XD uses five sensors, three encoders for each motor for position feedback, one camera to detect the cracks and an ultrasonic sensor to detect obstacles if there is any.

4.1.2. Raspberry PI Camera NV

Raspberry Pi Night Vision Camera plugs directly into the CSI connector on the Raspberry Pi.

This Raspberry Pi night vision camera uses the same OV5647 as the standard Raspberry Pi camera, and is therefore able to deliver a crystal clear 5MP resolution image, or 1080p HD video recording at 30fps.

4.1.3. DC Motor Encoder

Attached to each DC Motor is an encoder which transmits a set number of electronic pulses for each rotation of its shaft so by using simple mathematical formulas it is easy to calculate the travelled distance.

4.2. Actuator Sizing

Estimated total weight from CAD is 10 kg driven by two DC motors and the two actuated wheels have the same parameters and conditions.

Wheel outer diameter = 66 mm

Assumed wheel mass = 0.3 kg

Rolling Coefficient of friction between wheel and pipe = 0.3

Assume all pipes has no inclination.

Assumed fixed speed operation = 2 rad/sec

t_{acc}= 1 sec

$J_{tot}=J_m+J_{eff}$ (on each actuated wheel)

$J_{actuated\ wheel}$ = (0.5*m*r^2)=(0.5*0.3* $(66 \times 10^{-3} / 2)^2$)= $1.6335*10^{-4}$ kg·m ^{2}

Using kinetic conversion theorem:

- $J_{body} = m*r^2=10*(66 \times 10^{-3} / 2)^2= 10.89*10^{-3}$ kg·m ^{2}
- $J_{rolling\ wheel} = 0.5*m*r^2= 0.5*0.3* (66 \times 10^{-3} / 2)^2= 1.6335*10^{-4}$ kg·m ^{2}

$J_{eff} = 1.6335*10^{-4} + 10.89*10^{-3} + 1.6335*10^{-4} = 11.22*10^{-3}$

$J_{tot}=J_m+J_{eff} = 11.22*10^{-3} + 11.22*10^{-3} = 0.02243$ kg·m ^{2}

T_r (assumed resistive torque) = 9.8(5+(0.3*3))*0.3* $(66 \times 10^{-3} / 2)$ = 0.57 Nm

$\ddot{\Theta}_{acc}$ = 2/0.1=20 rad/sec

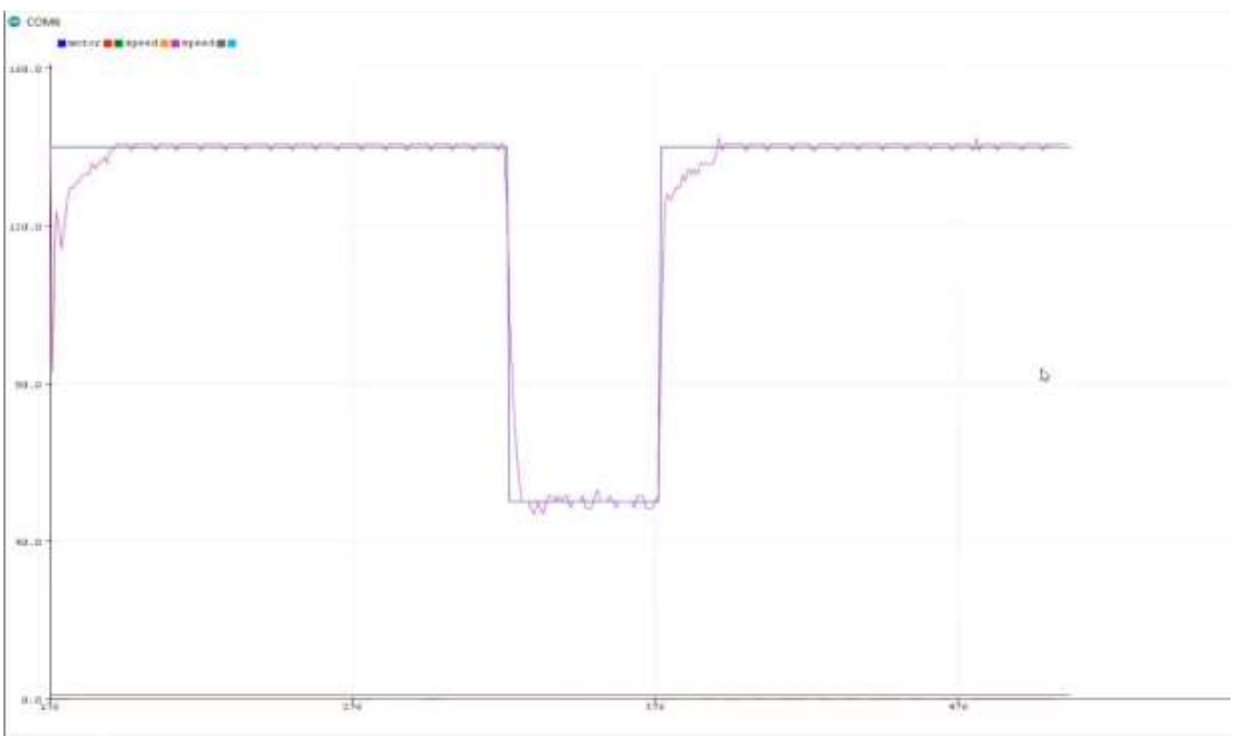
Peak torque required = T_{max} = 0.02243*20 + 0.57 = 1.5 Nm

4.3. PID Control

PID Parameters : Kp = 1.1, Ki = 4.8, Kd = 0.05;

The PID Graph in No load condition

The PID in real conditions



4.4. Battery Sizing

Average current draw by each component:

- Arduino uno board 0.05 A
- Raspberry pi 2.5 A
- Three driving DC motors 3.5 A
- Power screw three Stepper motor 5.25 A
- Ultrasonic sensor 0.015A
- Camera 0.5A

For motors I(amp) = (Nω*2π/V*60)

Consumed Power = ((0.05+2.5+0.015+0.5)*5) +((3.5+5.25)*12) = 120 W

Assumed total power losses 10 W

Total power = 130 W

Battery Voltage 12V

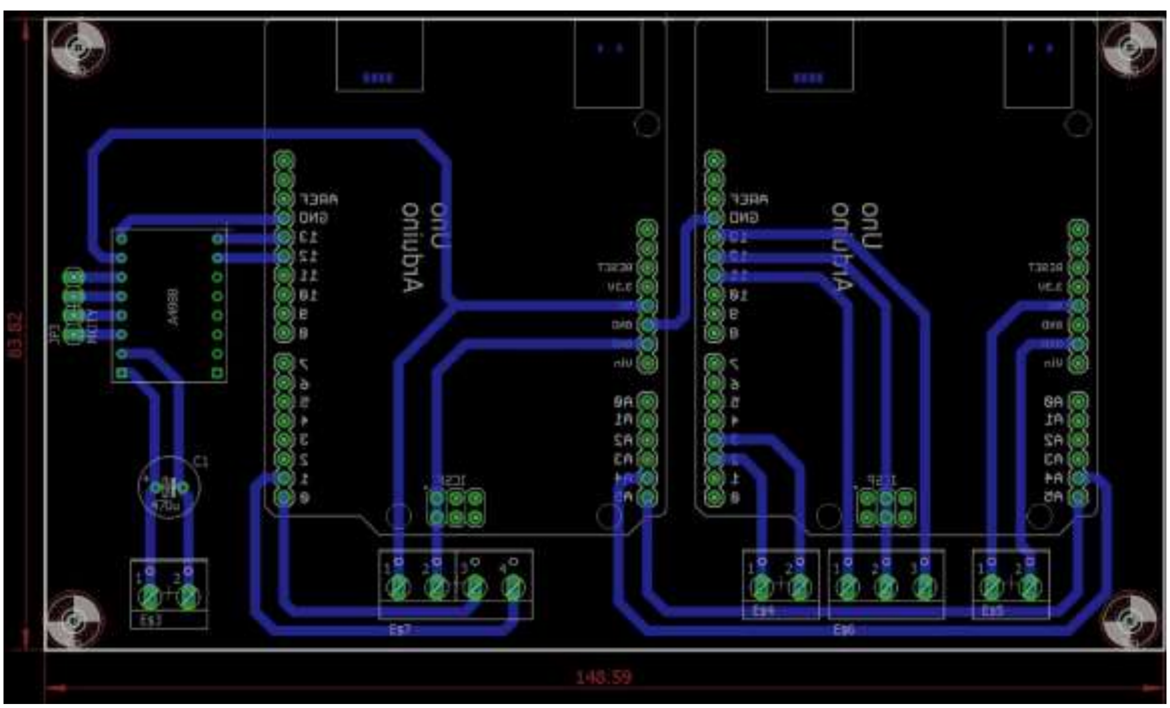
Total current = 10 A

Operating time 30 minutes

Battery size needed = 5 Ah

4.5.PCB Design

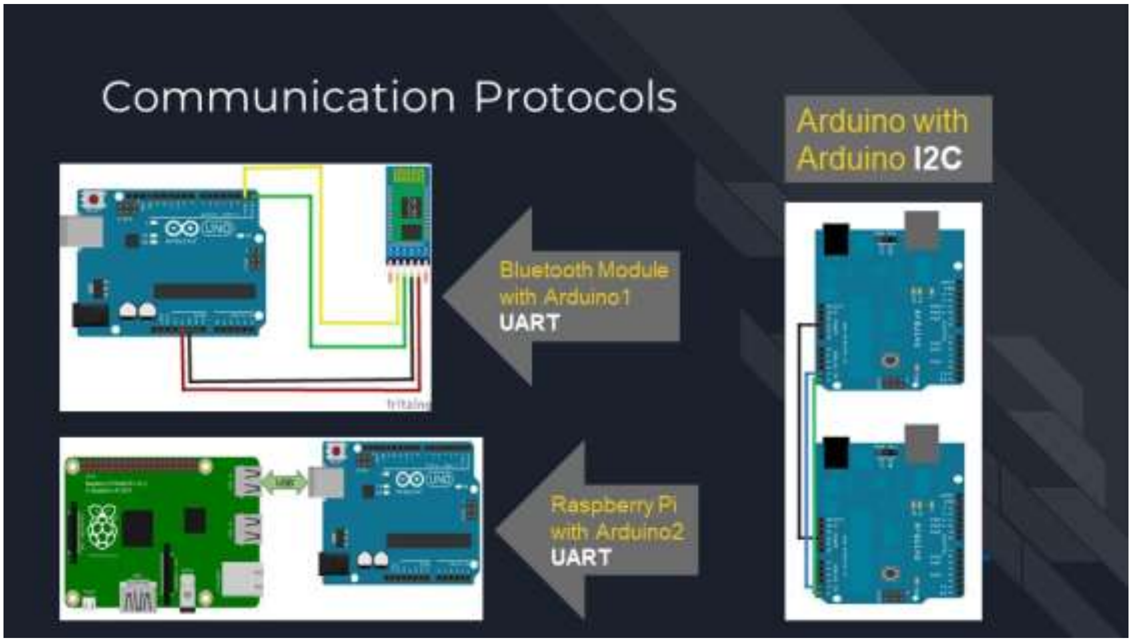
4.5.1. PCB Schematic



4.5.2. Communication Protocols

- Arduino1 controls the stepper motor and the bluetooth module
- Arduino2 controls the Dc Motors with the pid control
- Raspberry pi controls the camera with the image processing

We tried to optimize our resources and use the different communication protocols available in the uno. So Arduino1 takes the diameter and distance commands from a mobile application using the Bluetooth module using **UART**, then it sends the data to Arduino2 using **I2C** protocol to begin moving the report, and in parallel the raspberry pi vision module works and when a crack is detected it sends a character to Arduino2 using **UART** to slow down to be able to see the crack.



4.6. Electrical BOM

No.	Part Name	Qty	Price EGP/1	Total Price
1	Raspberry PI	1	1250	1250
2	DC Motor	3	200	600
3	Ultrasonic Sensor	1	45	45
4	PI Camera	1	450	450
5	Battery	1	750	750
6	Charging Port	1	50	50
7	Start/Stop Button	2	20	40
8	Arduino uno	1	160	160
9	PCB	1	200	200
10	Battery Indicator	1	50	50
11	DC Motor driver	3	95	285
12	Stepper driver	3		
13	Encoder	3	100	300
14	LED	3		
15	Stepper motor	3	100	300
Total				**3500**

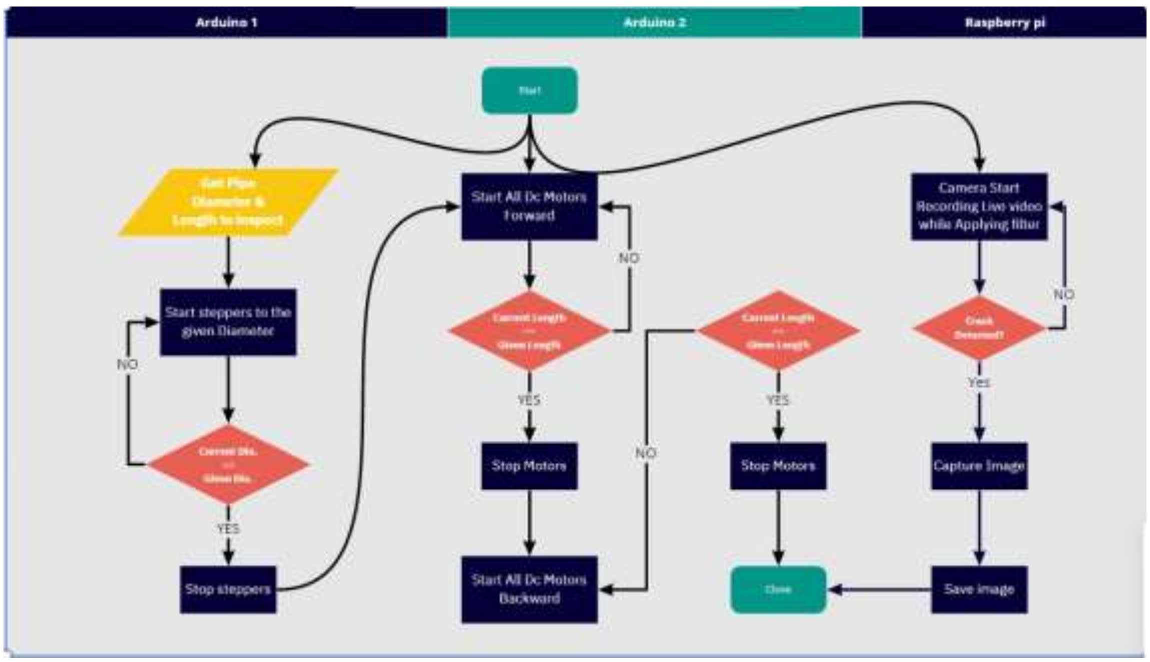
Figure 12 Electrical BOM

6. MAIN CODE

RC-XD's main purpose is to identify any crack, locate its position, get its shape and save photos named with the identified crack position and degree with its dimensions.

Our code is responsible for all **RC-XD's** functions, defining crack dimensions and recognizing any crack located inside a pipe.

6.1 FlowChart



6.1.1 Crack Recognition

```
# Python program to explain cv2.imshow() method

# importing cv2


from scipy.spatial import distance as dist
import numpy as np
import cv2
from imutils import perspective
from imutils import contours
import imutils
#from serial import Serial
import serial
import time

if __name__ == '__main__':
 ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1)      #Open port with
                                                             baud rate

# capture frames from a camera
```

```
cap = cv2.VideoCapture(0)

pixelsPerMetric =3.139

def midpoint(ptA, ptB):
    return ((ptA[0] + ptB[0]) * 0.5, (ptA[1] + ptB[1]) * 0.5)

num=1


# loop runs if capturing has been initialized
while (1):

    # reads frames from a camera
    ret, frame = cap.read()

    # converting BGR to HSV

    # Reading same image in another
    # variable and converting to gray scale.
    gray = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
    gray = cv2.GaussianBlur(frame, (7, 7), 0)

    # canny fcn is used to detect the edges if the image has no edges it
produce a
    # none image (black) , but if edges were detected in marks the edges with
white color
    edges = cv2.Canny(gray, 0, 255)

    edges = cv2.dilate(edges, None, iterations=15)
    edges = cv2.erode(edges, None, iterations=15)

    # Display an original image
    cv2.imshow('Original', frame)  #uncomment to see the original window
frame

    # discovers edges in the input image image and
    # marks them in the output map edges
    edges = cv2.Canny(frame, 100, 200)

    # Display edges in a frame
    cv2.imshow('Edges', edges)

    # image color check to see if edges exist after canny fcn
    if cv2.countNonZero(edges) == 0:
        print("Image has no edges")
    else:
        print("detected edges in the image")
        ser.write('q'.encode('utf-8'))
        time.sleep(1)
        # contourin the found edges
        (cnts, _) = cv2.findContours(edges.copy(), cv2.RETR_TREE,
                                     cv2.CHAIN_APPROX_SIMPLE)
        (cnts, _) = contours.sort_contours(cnts)

    count = 1
    # Going through every contours found in the image.
```

```
    for c in cnts:
        # if the contour is not sufficently large, ignore it
        if cv2.contourArea(c) < 100:
            continue

            # compute the rotated bounding box of the contour
        orig = frame.copy()
        box = cv2.minAreaRect(c)
        box = cv2.cv.BoxPoints(box) if imutils.is_cv2()
        else    cv2.boxPoints(box)
        box = np.array(box, dtype="int")
        cv2.drawContours(orig, [box.astype("int")], -1, (0, 255, 0), 2)
        box = perspective.order_points(box)

        for (x, y) in box:
            cv2.circle(orig, (int(x), int(y)), 5, (0, 0, 255), -1)

        (tl, tr, br, bl) = box
        (tltrX, tltrY) = midpoint(tl, tr)
        (blbrX, blbrY) = midpoint(bl, br)

        (tlblX, tlblY) = midpoint(tl, bl)
        (trbrX, trbrY) = midpoint(tr, br)

        cv2.circle(orig, (int(tltrX), int(tltrY)), 5, (255, 0, 0), -1)
        cv2.circle(orig, (int(blbrX), int(blbrY)), 5, (255, 0, 0), -1)
        cv2.circle(orig, (int(tlblX), int(tlblY)), 5, (255, 0, 0), -1)
        cv2.circle(orig, (int(trbrX), int(trbrY)), 5, (255, 0, 0), -1)
        cv2.line(orig, (int(tltrX), int(tltrY)), (int(blbrX), int(blbrY)),
                     (255, 0, 255), 2)
        cv2.line(orig, (int(tlblX), int(tlblY)), (int(trbrX), int(trbrY)),
                     (255, 0, 255), 2)

        dA = dist.euclidean((tltrX, tltrY), (blbrX, blbrY))
        dB = dist.euclidean((tlblX, tlblY), (trbrX, trbrY))

        dimA = dA / pixelsPerMetric
        dimB = dB / pixelsPerMetric

        cv2.putText(orig, "D1 = {:.1f}mm".format(dimA),
                        (int(20), int(20)), cv2.FONT_HERSHEY_SIMPLEX,
                        0.65, (255, 100, 122), 2)
        cv2.putText(orig, "D2 = {:.1f}mm".format(dimB),
                        (int(20), int(50)), cv2.FONT_HERSHEY_SIMPLEX,
                        0.65, (255, 100, 122), 2)

        cv2.imwrite("frame%d.jpeg" %num,orig)   # save frame as JPEG file
        num += 1
    k = cv2.waitKey(5) & 0xFF
    if k == 27:
        break   # exit if Escape is hit# press for Esc key to stop  or close
                the window

# Close the window
cap.release()
```

```
# De-allocate any associated memory usage
cv2.destroyAllWindows()
```

6.3. Output Samples

Here are some output samples saved for random crack pictures for testing:

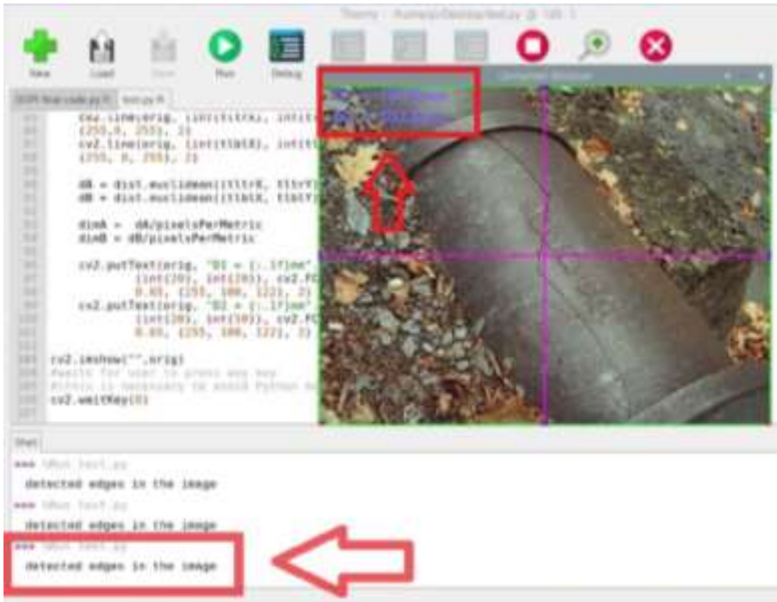


Figure 13 crack detection sample 1

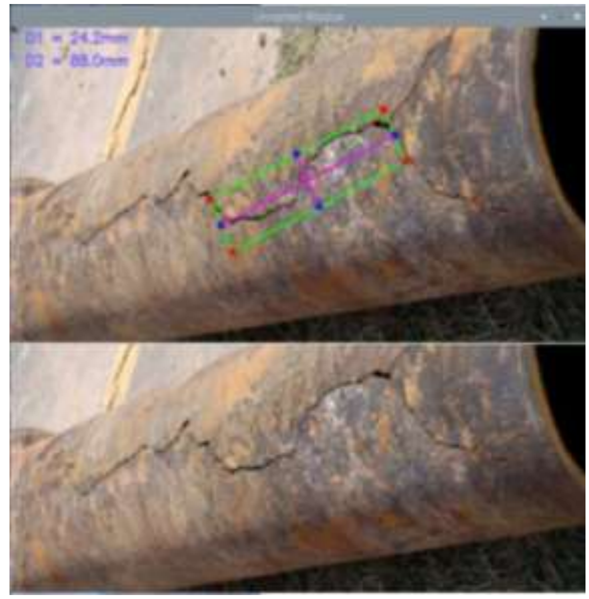


Figure 14 crack detection sample 2

These are Real life photos for cracks detected by the camera: