



An-Najah National University

Faculty of Engineering & Information Technology

Computer Engineering Department

Graduation Project 2

GrowBot Pro

Students:

Shahd Noman Hamad

Raia Abdullah Ishtaya

Supervisor:

Dr. Haya Samana

hayasam@najah.edu

September 7, 2024

Acknowledgment

First and foremost, we want to express our heartfelt gratitude to our parents and friends for their unwavering support throughout our academic journey. Your encouragement has been the driving force behind our determination, and this achievement is as much yours as it is ours.

We extend a special thanks to our supervisor, Dr. Haya Samaana, for her invaluable guidance and dedication. Her expertise has been instrumental in shaping our project and bringing it to fruition.

We are also grateful to the faculty members of the College of Engineering and the Computer and Electrical Engineering Department. Your knowledge and guidance have been the cornerstone of our success, and we deeply appreciate your contributions.

A sincere thank you goes to Alaa Al-Zarba, the university carpenter, for his support and helpful advice throughout the project.

Finally, to our classmates and friends, thank you for the companionship and the shared experiences that made this journey truly memorable.

With sincere appreciation,

Shahd Hamad

Raia Ishtaya

Disclaimer

This report was created by Shahd Hamad and Raia Ishtaya from the Computer Engineering Department at the Faculty of Engineering, An-Najah National University. Please note that it hasn't been revised beyond minor editorial fixes, so it might still have some language or content errors. The opinions, outcomes, and recommendations in this report are entirely those of the authors. An-Najah National University is not responsible for any consequences if this report is used for anything other than its intended purpose.

Contents

List of Figures	5
Abstract	6
List of Figures	6
1 Introduction	7
1.1 General background	7
1.2 Objectives	7
1.3 Significance	7
1.4 Organization of the report	8
2 Theoretical Background and Previous Work	9
2.1 Historical Context	9
2.2 Existing Literature	9
2.3 Evolution of Agricultural Technology and Related Work	10
3 Methodology	12
3.1 Standards and Specifications	12
3.2 Hardware Components	12
3.2.1 Microcontrollers	12
3.2.2 Motors and drivers	14
3.2.3 Sensors	16
3.2.4 Power devices	16
3.2.5 Other Devices	17
3.3 Experimental Procedures	25
3.4 Arduino Code	26
4 Discussion	35
4.1 Mechanical Challenges	35
4.2 Electrical Challenges	36

5 Conclusion and Future Work	38
5.1 Summary	38
5.2 Conclusion	38
5.3 Recommendations	39
5.4 What We Have Learned	40
5.5 Future Work	40
Bibliographic	42

List of Figures

3.1	Arduino Mega 2560 Microcontroller Board	13
3.2	ESP32-DevKitC ESP32-WROOM-32U Core Board	13
3.3	DFPlayer	13
3.4	Servo Motor	14
3.5	NEMA 23 Stepper motor and DIV268N driver	14
3.6	Wiper Motor	15
3.7	Drill Motor	15
3.8	DC Motor	15
3.9	Ultrasonic Sensor	16
3.10	Batteries EXP1290 and NP18-12R	16
3.11	Power Bank	17
3.12	Relay	18
3.13	Driver IBT-2	18
3.14	Plastic cylinder container	19
3.15	Funnels	19
3.16	Pump	19
3.17	SS UnEqual Tee 316 Grade	20
3.18	Hoses	20
3.19	Rake	21
3.20	Digger	21
3.21	Wheels	22
3.22	External structure	22
3.23	articulated arm	22
3.24	Wooden Hold	23
3.25	Wires	23
3.26	Collectors	23
3.27	Alarm Sound Device	24
4.1	Before and After Resolve Gear Issue	36

Abstract

The "Farmer Robot" represents a distinctive hardware project developed for graduation, poised to revolutionize the realm of agriculture, the robot equipped with wheels that allow it to move in all directions and across various terrains. This project focuses on drilling into various types of soil, planting the seeds, and providing irrigation to aid their growth. It also involves clearing the land before planting using a land rake. The project brings together mechanical engineering, robotics, and agriculture to streamline and accelerate the farming process while reducing the need for manual labor. The process of designing and building it underwent various modifications and developments to ensure its effectiveness in real-world agricultural tasks. We built this project with various motor , including stepper motor and DC motors,We also used servo, and ultrasonic sensors.We needed axle and a metal frame to build the structure of the robot. The microcontroller we used is an Arduino Mega, and we also used an ESP32 module. Lastly, we handled interfacing with external components such as a pump.

Chapter 1

Introduction

1.1 General background

The agricultural sector is a crucial element in ensuring global food security, and technological advancements are essential to meet the growing demand for effective farming practices. Traditional methods are often labor-intensive and time-consuming, highlighting the need for innovative solutions to enhance productivity. In this context, our project introduces a specialized farming robot designed to revolutionize agricultural processes by automating key tasks such as planting, irrigation, and soil preparation before planting.

1.2 Objectives

The core aim of this project is to create an autonomous agricultural robot that can independently traverse farm fields and carry out essential functions such as soil cultivation, excavation, seed planting, and irrigation. The objective is to minimize reliance on manual labor, enhance operational productivity, and support sustainable farming practices. By combining robotics with agriculture, the project seeks to tackle the challenges encountered by contemporary farmers and propel the industry into a new phase of technological advancement.

1.3 Significance

The significance of this work lies in its ability to revolutionize traditional agricultural methods. With the growing global population and increasing pressures on food production, automating essential agricultural tasks has become crucial to meet the rising demand and achieve greater efficiency in farming. This project aims to enhance agricultural operations by providing automated solutions to improve seed planting, irrigation, and soil preparation before planting. In addition to improving efficiency, this project also contributes to better resource utilization

and environmental sustainability. By reducing the need for manual labor and offering innovative agricultural techniques, the project seeks to minimize waste and increase productivity, thereby helping to preserve the environment and achieve a balance between agricultural needs and sustainable practices.

Moreover, aligning with market demands and expectations highlights the importance of this work in addressing contemporary challenges in agriculture, making it a vital project for achieving long-term improvements in the agricultural sector.

1.4 Organization of the report

This report is organized into several sections. The introduction provides an overview of the project and its objectives. The second section describes the scope and boundaries of the work. The third section outlines the methodology and procedures followed in completing the project. The fourth section presents the results and findings, including any challenges encountered and how they were overcome. The fifth section discusses the significance and potential impact of the project. Finally, the conclusion summarizes the key points of the report and provides recommendations for future work. Appendices are also included to provide additional information and data relevant to the project.

Chapter 2

Theoretical Background and Previous Work

2.1 Historical Context

The development of autonomous agricultural technology stems from the ongoing pursuit of greater efficiency and sustainability in farming. The journey began with the introduction of tractors, which aimed to reduce the reliance on manual labor. However, it quickly became clear that more advanced systems were needed for fully autonomous farming.

By the late 20th century, precision agriculture emerged, utilizing technologies such as Global Positioning System (GPS) and remote sensing. These innovations improved the accuracy of agricultural practices and supported data-driven methods. Despite these advancements, the vision of fully autonomous machines performing complex tasks gained significant momentum. In agriculture and irrigation, early efforts focused on enhancing precision, while recent decades have seen a rise in the use of robotics and artificial intelligence, redefining possibilities in farming. Our project, an autonomous farmer robot, builds on this historical context with the aim of contributing to the ongoing evolution of autonomous agricultural systems. This historical framework positions our project as an innovative step towards increasing efficiency and sustainability in modern agriculture.

2.2 Existing Literature

Research on autonomous farm robots reveals significant advancements in automating soil preparation and irrigation. Precision agriculture has evolved with robotic systems designed for tasks such as soil tilling and watering, improving efficiency and sustainability. Studies highlight innovations in autonomous irrigation systems and soil preparation robots, which enhance pre-

cision and resource management. Integrated robotic systems that combine soil preparation and irrigation tasks are particularly notable, offering streamlined solutions for modern agriculture.

2.3 Evolution of Agricultural Technology and Related Work

The process of farming has undergone many stages, evolving from manual methods to the current mechanical ones.

- In the early 19th century, the early seed drill, designed by Jethro Tull, was one of the first machines used in agriculture. This seed drill was pulled by horses and had rotating parts that distributed seeds in the soil at regular intervals. Although the machine was simple, it was effective in reducing the manual labor involved in the planting process
- Mechanical Seed Planters (Mid to Late 19th Century) Planet Jr. Seed Planter (1890s), was a significant advancement in farming technology. It featured adjustable gears for varying seed depths and spacing, and was made from cast iron, steel, and wood. The planter was pushed across the field, with a V-shaped furrower at the front and metal wheels at the back to cover the seeds. It could be attached to a tractor or other transport, enhancing its efficiency compared to manual sowing methods.
- Modern Automated Seed Sowing Robots (Late 20th Century to Early 21st Century)
 - 1- Naïo Technologies' Oz (2013) Designed by the French company Naïo Technologies in 2013, the Oz robot is specifically engineered for agricultural tasks such as planting, weeding, and soil preparation. It is equipped with a global positioning system (GPS) and obstacle detection sensors, allowing for remote control and monitoring via Wi-Fi.
 - 2-The FarmBot Genesis, developed by inventor Rory Aronson in the United States in 2016, is an open-source farming robot tailored for small-scale and home-based agriculture. It features a gantry system that allows for precise planting of seeds, efficient watering of plants, and careful monitoring of growth. The robot is equipped with modular tools that facilitate various agricultural tasks, including drilling, planting, and watering. It is connected to the internet, enabling users to control and monitor the robot remotely through a web application. Additionally, the FarmBot Genesis includes sensors to track soil moisture, temperature, and plant growth, and it can be programmed to plant a variety of crops with high accuracy.
 - 3- The Automated Seeder and Planter Robot (2019) Developed by AgXeed in the Netherlands in 2019, the Automated Seeder and Planter Robot represents a significant advancement in agricultural automation. This fully autonomous robot is designed for large-scale seeding and planting operations, featuring precision seed drills that ensure accurate placement of seeds. Utilizing GPS and machine learning technologies, it navigates fields with remarkable accuracy and collects valuable data. The robot is versatile, capable of planting various crops and adjusting seeding rates in response to soil conditions. Additionally, it provides real-time monitoring of field conditions and maintains precise control over seed depth.
 - 4- SeederBot (2020) The SeederBot, developed by Autonomous Tractor Corporation in the USA in 2020, is an advanced autonomous seeder robot designed for precision planting on large farms. It employs sophisticated AI algorithms to optimize seed

spacing and depth, ensuring efficient and accurate seeding. The SeederBot can operate in conjunction with other robotic systems to achieve comprehensive farm automation. Equipped with sensors that analyze soil conditions, including moisture and temperature, it provides real-time data to farmers, facilitating better decision-making and optimizing crop yields. Our project shares a common sub-process with previous fieldwork, including soil drilling techniques, seed distribution, and irrigation. However, our machine includes several unique additional features that set it apart from its predecessors. Our equipment is equipped with mobile app capabilities for remote control of the robot's movement and the drilling and planting process. Additionally, it has obstacle detection and alerting systems to notify of any encountered obstacles.

Chapter 3

Methodology

In this chapter, we provide a comprehensive explanation of the materials, methods, and standards used in the development of the autonomous farming robot. The methodology covers the design and integration of various components, each of which plays a crucial role in ensuring the effective and seamless execution of tasks related to planting, irrigation, and soil preparation.

3.1 Standards and Specifications

Our design adheres to engineering standards to ensure reliability and compatibility. Notably, the IEEE 802.11 standard is employed for communication protocols, focusing on seamless connectivity and efficient data exchange within the system. Additionally, the design rigorously complies with safety standards, enhancing overall safety and user well-being.

3.2 Hardware Components

3.2.1 Microcontrollers

Arduino Mega 2560

The Arduino Mega 2560 microcontroller board, built around the ATmega2560 chip, offers a robust set of features for various applications. It includes 54 digital input/output pins, 15 of which can be used for pulse-width modulation (PWM) outputs. The board also provides 16 analog inputs, 4 hardware serial ports (UARTs), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. This extensive feature set makes it an ideal choice for our project needs. Given the large number of devices that need to interface with the microcontroller, the Arduino Uno was deemed insufficient, leading us to select the Arduino Mega 2560 as the central microcontroller for our project.

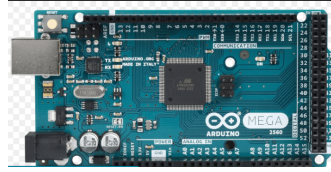


Figure 3.1: Arduino Mega 2560 Microcontroller Board

ESP32-DevKitC ESP32-WROOM-32U Core Board In our project, both the ESP32 and Arduino Mega microcontrollers were essential for establishing serial communication. To streamline user connectivity, we developed a web server-like access point, allowing users to connect to the planting machine network from any mobile device within their office. This setup enabled remote placement of pot orders.

Upon receiving an order via the serial communication channel, the Arduino Mega would promptly initiate the pot preparation process.



Figure 3.2: ESP32-DevKitC ESP32-WROOM-32U Core Board

DFPlayer with SDCard DFPlayer Mini is a small, low-cost MP3 player module that integrates with microcontrollers such as Arduino. It can play audio files stored on an SD card, and is typically used in projects that require audio playback. In this project, it was used to make a sound when sensing an ultrasonic signal when a robot approaches an obstacle to stop it.

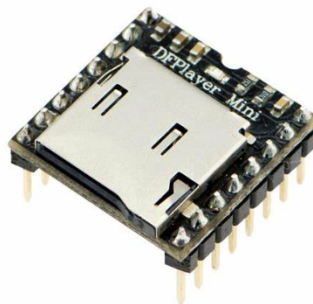


Figure 3.3: DFPlayer

3.2.2 Motors and drivers

Servo motor



Figure 3.4: Servo Motor

Servo motors are commonly used for their precision in controlling position, speed, and torque. They are integral to many machines and devices that require exact and controlled movement. A typical servo motor includes a small DC motor, a gearbox, and a control circuit. This control circuit provides feedback to ensure the motor reaches and maintains the exact position specified by input signals, often with accuracy within a few degrees. In our project, we used small servo motors with a rotation range of approximately 180 degrees (90 degrees in each direction). These servos were installed at the end of the seed container to manage the precise release of seeds during the planting process.

NEMA 23 Stepper motor and DIV268N driver The NEMA 23 stepper motor is known for its high torque and precision, making it ideal for demanding applications such as CNC machines and 3D printers. It typically delivers 200 steps per revolution, allowing for precise motion control. This combination is paired with the DIV268N motor, which supports high current and various microstepping options, ensuring smooth and precise motor control. The DIV268N also includes over-current and over-heat protection, enhancing reliability in various precision tasks. It was used in our project to facilitate the robot's movement to the right and left.

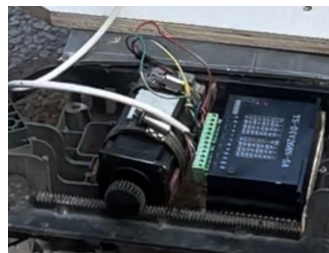


Figure 3.5: NEMA 23 Stepper motor and DIV268N driver

Wiper Motor

We used three motors for our vehicle: two to power the rear wheels and one to drive the digger.



Figure 3.6: Wiper Motor

The wiper motor is crucial for rotating the digger to excavate soil and reverse its direction to remove any debris after planting. Additionally, the motors for the rear wheels enable the robot to move forward and backward efficiently

Drill Motor We used a drill motor to move the digger up and down. Initially, we tried using a wiper motor, but its speed was insufficient for our needs. The drill motor provided the necessary power and control for effective operation of the digger.



Figure 3.7: Drill Motor

Small DC Motor We used two small DC motors: one to drive the hose that dispenses water with seeds, and the other to operate the rake that moves up and down.



Figure 3.8: DC Motor

3.2.3 Sensors

Ultrasonic Sensor In our project, the ultrasonic sensor played a crucial role in monitoring the remaining quantities of seeds and water. This was achieved by accurately measuring distances, allowing real-time estimates of the remaining contents. This capability enabled the system to replenish components promptly as needed, ensuring continuous operation. Additionally, the sensor served another vital function by detecting obstacles through its ability to sense any obstructions in its path, thereby enhancing operational safety and efficiency.

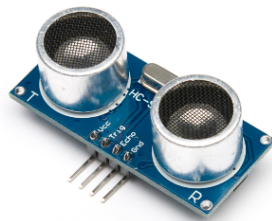


Figure 3.9: Ultrasonic Sensor

3.2.4 Power devices

Batteries EXP1290 and NP18-12R For our farm robot project, we chose two 12V LiPo batteries, one 9Ah and the other 30Ah, due to their high energy density and lightweight design. This battery offers a good balance between capacity and weight, ensuring extended runtime while minimizing impact on the overall performance of the robot. It provides enough power for the robot's motors and electronics, making it an ideal choice for reliable and efficient operation.



Figure 3.10: Batteries EXP1290 and NP18-12R

Arduino Power Cable The Arduino Power Cable connects an Arduino board to a power source, such as a computer's USB port or a wall adapter, providing necessary power for the board's operation and enabling programming and data transfer. In our project, we used the Arduino Power Cable to ensure a stable 5-volt power supply to the Arduino board, which is essential for its proper functionality.

ESP Power Cable The ESP Power Cable serves as the connection between an ESP32 or ESP8266 microcontroller and its power source, such as a USB port or a dedicated power adapter. It ensures that the microcontroller receives a stable and adequate voltage, which is crucial for its operation. In our project, we utilized the ESP Power Cable to provide consistent power to the ESP32, ensuring reliable performance and uninterrupted operation of the wireless communication and control features.

Power Bank In our project, each microcontroller, the ESP32 and the Arduino Mega, is powered by its own separate power bank. This setup ensures that each device receives a stable and dedicated power supply, reducing the risk of power-related issues and optimizing the performance of both microcontrollers independently.



Figure 3.11: Power Bank

3.2.5 Other Devices

Relay In our project, we utilize three relays to control various components and devices. Relays act as electrically operated switches, allowing us to manage high-power loads with low-power control signals from the microcontrollers. The use of three relays provides the flexibility to handle multiple functions, such as controlling motors, actuators, or other high-current devices, ensuring the efficient operation of the robot's systems.

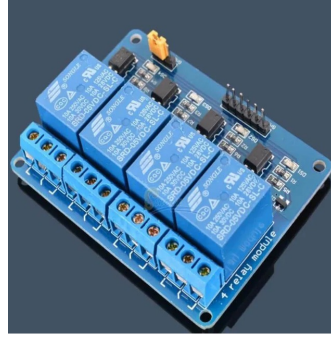


Figure 3.12: Relay

Driver IBT-2 The IBT-2 is a versatile motor driver designed for controlling DC and stepper motors. It features an H-bridge design that allows bidirectional control, handling high currents up to 43A per channel. Suitable for a range of applications, including robotics and automation, the IBT-2 interfaces with microcontrollers through PWM signals for precise speed and direction control. Its robust performance makes it an excellent choice for driving powerful motors in various electronic projects.

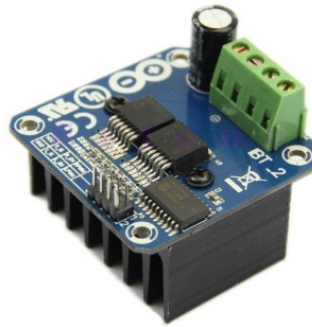


Figure 3.13: Driver IBT-2

Plastic cylinder container In our project, we use two plastic cylinder containers: one for holding seeds and the other for storing water. These containers are designed to efficiently manage and separate the seed and water supplies, ensuring smooth operation during planting and irrigation processes. The plastic construction provides durability and ease of handling, making them ideal for our agricultural automation needs.

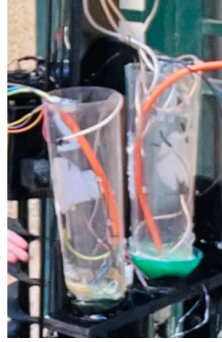


Figure 3.14: Plastic cylinder container

Funnels



Figure 3.15: Funnels

Each plastic cylinder in our project is equipped with a funnel at its end. The funnels facilitate easy filling and dispensing of seeds and water, ensuring a smooth and controlled transfer of materials. This setup helps prevent spills and improves the efficiency of the planting and irrigation processes.

Pump In our project, we utilize a pump connected to the water funnel to automate the irrigation process. The pump efficiently transfers water from the container through the funnel, providing precise and controlled watering to the plants. This setup ensures consistent moisture levels and supports the overall functionality of the robot in maintaining healthy plant growth.



Figure 3.16: Pump

SS UnEqual Tee 316 Grade In our project, we use an SS Unequal Tee 316 Grade fitting. This fitting has three openings: the first connects to the pump, the second is linked to the seed

dispenser, and the third is used for the combined output of water and seeds. This configuration allows for efficient distribution and management of materials during the planting and irrigation processes.



Figure 3.17: SS UnEqual Tee 316 Grade

Hoses We also utilized hoses of various sizes in our project. These hoses facilitate the transport



Figure 3.18: Hoses

of water and seeds through the system, ensuring efficient and flexible connections between the different components. The use of different hose sizes allows us to tailor the flow rates and connectivity to meet the specific needs of the planting and irrigation processes.

Rake We also employed a rake to comb and prepare the soil. The rake helps in leveling and smoothing the soil, ensuring a well-prepared bed for planting. This step is crucial for optimal seed placement and soil condition, contributing to the overall efficiency and effectiveness of the planting process.



Figure 3.19: Rake

Digger We also used a digger to excavate the soil. The digger is essential for creating the necessary planting holes, which facilitates accurate seed placement and prepares the soil for effective planting. This tool enhances the efficiency of soil preparation and contributes to the overall success of the planting process.



Figure 3.20: Digger

Wheels We also used four wheels designed for soil handling. These wheels aid in maneuvering the robot across the field, ensuring smooth movement and stability on various terrain types. They help distribute the robot's weight evenly and reduce soil disturbance, enhancing the overall efficiency of the farming operations.



Figure 3.21: Wheels

External structure We also designed and built the external structure of the robot, ensuring it could withstand various agricultural tasks while providing support for internal components such as motors, pipes, the digger, and wheels. The materials were selected to be lightweight yet durable, ensuring efficient movement and the ability to endure different environmental conditions in agricultural fields.



Figure 3.22: External structure

articulated arm We used an articulated arm to steer the front wheels left and right.



Figure 3.23: articulated arm

Wooden Hold We used a wooden support to hold the digger and facilitate its up-and-down movement.



Figure 3.24: Wooden Hold

Wires We used 4 types of wires: male-to-male, female-to-female, male-to-female wires for various connections, and intercom wires.



Figure 3.25: Wires

Collectors We also used wire collectors to organize and manage the wiring system efficiently. These collectors ensured that the wires were neatly arranged, reducing clutter and preventing tangling, which helped maintain a clean and functional setup for the robot's electrical components.

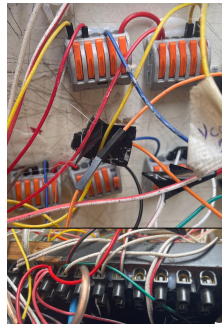


Figure 3.26: Collectors

Alarm Sound Device We incorporated an alarm sound device into our system to enhance safety. This device is triggered when the ultrasonic sensor detects an obstacle at a specified distance.



Figure 3.27: Alarm Sound Device

3.3 Experimental Procedures

- **Forward and Backward Movement**

The robot's forward and backward movement is controlled by the rear wheels, each of which is powered by a dedicated wiper motor. These motors are crucial for driving the robot in both directions, providing the necessary propulsion for navigation. The setup ensures that the robot can move efficiently and maneuver effectively, allowing it to perform its tasks with precision and stability.

- **Right and Left Movement**

The robot's movement to the right and left is facilitated by a stepper motor attached to a pulley system. This pulley operates on a toothed gear, which helps to streamline and smooth out the motion. This setup enables precise and controlled lateral movement, enhancing the robot's maneuverability and allowing it to navigate its environment effectively.

- **Digger Movement**

The digger rotates clockwise to dig the soil and prepare the hole, and counterclockwise after planting to push the accumulated dirt back into the hole. This operation is controlled by a motor, allowing for effective soil preparation and coverage.

- **Wooden Frame Movement**

The wooden frame that supports the digger moves downward during the digging process to allow for effective soil excavation and moves upward once the digging is complete. This motion is achieved using a drill motor.

- **Rake Movement**

The rake moves downward to begin leveling the soil and upward when the process is complete, using a DC motor mounted on a steering mechanism (typically used in children's toys).

- **Planting and Irrigation system**

The planting and irrigation system operates as follows: After completing the excavation, a DC motor mounted on a steering mechanism (typically used in children's toys) is connected to a hose. This hose is attached to one of the elbow fittings. The steering mechanism adjusts the position of the hose fitting, and then a servo opens to release seeds through one of the elbow fittings. Simultaneously, the pump connected to another elbow fitting starts working, mixing seeds with water and directing the mixture into the hole. Once the process is complete, the steering mechanism returns to its original position. The elbow fitting has three openings: one for the pump, one for the seeds, and one for mixing water with seeds.

- **Alarm system**

The alarm system activates when the ultrasonic sensor detects an obstacle at a specified

distance. Upon detection, it sends a signal that triggers the alarm, which then halts the robot's movement to prevent collision or damage.

3.4 Arduino Code

```

1 #include <Servo.h>
2 #define TRIG_PIN 40 // Trigger pin of the ultrasonic sensor
3 #define ECHO_PIN 41 // Echo pin of the ultrasonic sensor
4 #define MAX_DISTANCE 200 // Maximum distance to measure (in cm)
5 //move forward and backward
6 #define CW 22 //CW is defined as pin #7//
7 #define CCW 23 //CCW is defined as pin #8//
8 #define CW1 24//CW is defined as pin #7//
9 #define CCW1 25 //CCW is defined as pin #8//
10 //servo to seed and pump to water
11 Servo myServo1; // Create a Servo object for the servo on pin 11
12 int relayPin =7; // Pin connected to re lay IN pin pump
13 //move car left and right
14 // Define pins for stepper motor
15 const int dirPin = 32;
16 const int stepPin = 34;
17 const int enPin = 33;
18 int flag=0;
19 // Define pins for motor 1 (up and down)
20 const int relayup = 10;
21 const int relaydown = 11;
22 // Define pins for motor 2 (right and left) hafara
23 int R_IS =38 ;
24 int R_EN =40;
25 int R_PWM =42;
26 int L_IS = 39;
27 int L_EN =41 ;
28 int L_PWM = 43;
29
30 ///
31 // Define pins for plowing relays
32 const int relayPinDirection1 = 8;
33 const int relayPinDirection2 = 9;
34
35 ///

```

```
36 // Define pins for seed
37 const int relay_waterseed1 =4;
38 const int relay_waterseed2= 5;
39 void setup() {
40   pinMode(TRIG_PIN, OUTPUT);
41   pinMode(ECHO_PIN, INPUT);
42   // Initialize the pins as outputs for stepper motor
43   pinMode(dirPin, OUTPUT);
44   pinMode(stepPin, OUTPUT);
45   pinMode(enPin, OUTPUT);
46   digitalWrite(enPin, LOW); // Enables the motor driver
47
48   // Initialize the pins as outputs for motor 1 up and down
49   pinMode(relayup, OUTPUT);
50   pinMode(relaydown, OUTPUT);
51   digitalWrite(relayup, HIGH);
52   digitalWrite(relaydown, HIGH );
53
54 //set up motor 2
55   pinMode(R_IS, OUTPUT);
56   pinMode(R_EN, OUTPUT);
57   pinMode(R_PWM, OUTPUT);
58   pinMode(L_IS, OUTPUT);
59   pinMode(L_EN, OUTPUT);
60   pinMode(L_PWM, OUTPUT);
61   digitalWrite(R_IS, LOW);
62   digitalWrite(L_IS, LOW);
63   digitalWrite(R_EN, HIGH);
64   digitalWrite(L_EN, HIGH);
65
66   //relay
67   pinMode(CW, OUTPUT); //Set CW as an output//
68   pinMode(CCW, OUTPUT); //Set CCW as an output//
69   //relay
70   pinMode(CW1, OUTPUT); //Set CW as an output//
71   pinMode(CCW1, OUTPUT); //Set CCW as an output//
72
73   // Turn off all relays initially
74   digitalWrite(CW, HIGH);
75   digitalWrite(CCW, HIGH);
76   digitalWrite(CW1, HIGH);
```

```
77   digitalWrite(CCW1,HIGH);
78
79   ///

80 myServo1.attach(6); // Attach the first servo to pin 11
81 myServo1.write(0); // Initialize the first servo at 0 degrees
82
83 ///

84 pinMode(relayPin, OUTPUT);
85 digitalWrite(relayPin, HIGH); // Ensure the relay is off initially //
86 // Initialize the pins for plowing relays
87 pinMode(relayPinDirection1, OUTPUT);
88 pinMode(relayPinDirection2, OUTPUT);
89 digitalWrite(relayPinDirection1, HIGH);
90 digitalWrite(relayPinDirection2, HIGH);
91
92 ///

93 pinMode(relay_waterseed1, OUTPUT);
94 pinMode(relay_waterseed2, OUTPUT);
95 digitalWrite(relay_waterseed1, HIGH);
96 digitalWrite(relay_waterseed2, HIGH);
97
98 // Initialize serial communication
99 Serial.begin(9600); // Communication with PC
100 Serial1.begin(9600); // Communication with ESP
101 }
102 void loop() {
103   delay(3000);
104   seedtoleft();
105   delay(3000);
106   water();
107   delay(3000);
108   seedtoright();
109   delay(3000);
110   if (Serial1.available())
111     { // of the command to determine the action
112       ///

112 //command.trim(); // able()) {
113       String command = Serial1.readStringUntil('\n'); // Read the command
114       as a string
115       Serial.println("Received: " + command); // Print the received
116       command for debugging


```

```
116 // Remove the "POST Data:" prefix if it exists
117 command.replace("POST Data:", "");
118 command.trim();
119 Serial.println(command); // Print the received command for
    debugging
120
121 // Use the first charRemove any leading or trailing whitespace
122
123 if (command.equals("forward")) {
124     moveForward();
125 } else if (command.equals("backward")) {
126     moveBackward();
127 } else if (command.equals("left")) {
128     turnLeft();
129 } else if (command.equals("right")) {
130     turnRight();
131 } else if (command.equals("stop")) {
132     stopMotors();
133 }
134 else if (command.equals("plowing")) {
135     plowing();
136 } else if (command.equals("agriculture")) {
137     moveDrilling();
138 }
139 else {
140     Serial.println("Unknown command");
141 }
142 }
143 }
144
145 void moveDrilling() {
146     Serial.println("Running moveMotor1Down and moveMotor2Left for 5
    seconds");
147     moveMotor1Down();
148     delay(9000);
149
150     moveMotor2Left();
151     delay(15000);
152     // Wait for 5 seconds
153 // Step 2: Stop moveMotor2Left and call moveMotor1Up for 5 seconds
154 stopMotors(); // Stop moveMotor2Left
```

```
155 Serial.println("Running moveMotor1Up for 5 seconds");
156 moveMotor1Up();
157 delay(20000); // Wait for 5 seconds
158 stopMotors(); // Stop moveMotor2Left
159 // Step 3: Call the water function
160 delay(3000);
161 seedtoleft();
162 delay(3000);
163 water();
164 delay(3000);
165 seedtoright();
166 delay(3000);
167 moveMotor1Down();
168 delay(9000); // Wait for 5 seconds
169 stopMotors();
170 moveMotor2Right();
171 delay(7000); // Wait for 5
172 stopMotors();
173 moveMotor1Up();
174 delay(9000);
175 stopMotors();
176 }
177
178 void moveForward() {
179 Serial.println("Moving forward"); // Debug message
180 digitalWrite(CW,HIGH); //Motor runs clockwise//
181 digitalWrite(CCW,LOW); //Motor runs clockwise//
182 digitalWrite(CW1,HIGH); //Motor runs clockwise//
183 digitalWrite(CCW1,LOW); //Motor runs clockwise//
184 }
185
186 void moveBackward() {
187 Serial.println("Moving backward"); // Debug message
188 digitalWrite(CW, LOW); //Motor runs counter-clockwise//
189 digitalWrite(CCW, HIGH); //Motor runs counter-clockwise//
190 digitalWrite(CW1, LOW); //Motor runs counter-clockwise//
191 digitalWrite(CCW1, HIGH); //Motor runs counter-clockwise//
192
193 }
194
195 void turnLeft() {
```

```
196 Serial.println("Turning left"); // Debug message
197 digitalWrite(dirPin, LOW);
198 for (int x = 0; x < 50; x++) { // Reduced to 50 steps
199     digitalWrite(stepPin, HIGH);
200     delayMicroseconds(1500);
201     digitalWrite(stepPin, LOW);
202     delayMicroseconds(2000);
203 }
204 }
205
206 void turnRight() {
207     Serial.println("Turning right"); // Debug message
208     digitalWrite(dirPin, HIGH);
209     for (int x = 0; x < 50; x++) { // Reduced to 50 steps
210         digitalWrite(stepPin, HIGH);
211         delayMicroseconds(1500);
212         digitalWrite(stepPin, LOW);
213         delayMicroseconds(2000);
214     }
215 }
216 // Motor 1 control functions (up and down)
217 void moveMotor1Up() {
218     Serial.println("Moving motor 1 up");
219     // Rotate the motor forward at 50% speed
220     digitalWrite(relayup, LOW);
221     digitalWrite(relaydown, HIGH);
222 }
223 void moveMotor1Down() {
224     Serial.println("Moving motor 1 down");
225     // Rotate the motor backward at 50% speed
226     digitalWrite(relayup, HIGH);
227     digitalWrite(relaydown, LOW);
228 }
229
230 // Motor 2 control functions (left and right)
231 void moveMotor2Left() {
232     Serial.println("Moving motor 2 left");
233     analogWrite(R_PWM, 200);
234     analogWrite(L_PWM, 0);
235
236 }
```

```
237
238 void moveMotor2Right() {
239     Serial.println("Moving motor 2 right");
240     analogWrite(R_PWM, 0);
241     analogWrite(L_PWM, 200);
242
243 }
244 void stopMotors() {
245     Serial.println("Stopping motors"); // Debug message
246     digitalWrite(relayup, HIGH);
247     digitalWrite(relaydown, HIGH);
248     analogWrite(R_PWM, 0);
249     analogWrite(L_PWM, 0);
250     digitalWrite(CCW, HIGH);
251     digitalWrite(CW, HIGH);
252     digitalWrite(CCW1, HIGH);
253     digitalWrite(CW1, HIGH);
254
255 if(flag==1)
256 {
257
258     //UP THE plowing
259     digitalWrite(relayPinDirection1, LOW);
260     digitalWrite(relayPinDirection2, HIGH);
261     delay(3000);
262     //UP THE plowing
263     digitalWrite(relayPinDirection1, HIGH);
264     digitalWrite(relayPinDirection2, HIGH);
265     delay(1000);
266     flag=0;
267 }
268
269     stepMotor ();
270     delay(1000);
271 }
272
273 void seedtoleft() {
274     // First sequence of relay commands
275     for (int i = 0; i < 100; i++) {
276         digitalWrite(relay_waterseed1, LOW);
277         digitalWrite(relay_waterseed2, HIGH);
```

```
278 }
279 delay(500);
280 stopSeed();
281 }
282
283 void seedtoright() {
284     // First sequence of relay commands
285     for (int i = 0; i < 100; i++) {
286         digitalWrite(relay_waterseed2, LOW);
287         digitalWrite(relay_waterseed1, HIGH);
288     }
289     delay(500);
290     stopSeed();
291 }
292
293 void stopSeed() {
294     digitalWrite(relay_waterseed1, HIGH);
295     digitalWrite(relay_waterseed2, HIGH);
296     delay(1000);
297 }
298
299 void water()
300 {
301     myServo1.write(90); // Rotate the first servo to 90 degrees
302     delay(2000); // Wait for 2 seconds
303     myServo1.write(0); // Rotate the first servo back to 0 degrees
304     delay(1000); // Wait for 1 second
305     // Pump control
306     Serial.println("pump");
307     digitalWrite(relayPin, LOW); // Turn the relay on (activates the
308         pump)
309     delay(9000); // Keep the pump on for 9 seconds
310     digitalWrite(relayPin, HIGH); // Turn the relay off (deactivates
311         the pump)
312     // Reset relays after the pump command
313     digitalWrite(relay_waterseed1, HIGH);
314     digitalWrite(relay_waterseed2, HIGH);
315     delay(2000);
316     // // Second sequence of relay commands
317     for(int i = 0; i < 100; i++) {
318         digitalWrite(relay_waterseed1, LOW);
```

```
317     digitalWrite(relay_waterseed2, HIGH);
318 }
319 delay(500);
320 // Final reset of relays
321 digitalWrite(relay_waterseed1, HIGH);
322 digitalWrite(relay_waterseed2, HIGH);
323
324 for(int i=0;i<100;i++){
325     digitalWrite(relay_waterseed1, HIGH);
326     digitalWrite(relay_waterseed2, LOW);
327 }
328 delay(500);
329
330 digitalWrite(relay_waterseed1, HIGH);
331 digitalWrite(relay_waterseed2, HIGH);
332 delay(1000);
333
334 myServo1.write(90); // Rotate the first servo to 90 degrees
335 delay(2000); // Wait for 3 seconds
336 myServo1.write(0); // Rotate the first servo back to 0 degrees
337 delay(1000); // Wait for 1 seconds
338 // //////////////////////////////////////PUMP
339 Serial.println("pump");
340 digitalWrite(relayPin, LOW); // Turn the relay on (activates the
    pump)
341 delay(9000); // Keep the pump on for 5 seconds
342 digitalWrite(relayPin, HIGH); // Turn the relay off (deactivates
    the pump)
343 }
```

Listing 3.1: Arduino Code for Farmer Robot

Chapter 4

Discussion

4.1 Mechanical Challenges

- **Vehicle Stability**

One of the significant mechanical challenges we encountered was maintaining the stability of the robot. Ensuring that the vehicle remained balanced and stable during operation was crucial for its effective performance and reliability. To address this issue, we consulted with several mechanics and experts who provided valuable insights and solutions. Through their guidance and iterative adjustments, we were able to resolve the stability issues and achieve the necessary balance for the robot's operation.

- **Rear Wheel Drive Force**

We faced a challenge with the drive force of the rear wheels. Initially, the wheels were connected to a single axle, which resulted in insufficient drive force for the robot. The power output was not meeting our expectations, impacting the robot's performance. To address this, we decided to separate the wheels, which allowed for individual drive forces. This adjustment significantly improved the robot's drive capabilities, providing the necessary power and performance for effective operation.

- **Pulley Wear**

We encountered a problem with the front wheel movement, which was driven by a stepper motor connected to a plastic pulley operating on a toothed gear. The plastic pulley wore out quickly due to the friction and stress from the gear mechanism. To resolve this issue, we replaced the plastic pulley with a metal one. The metal pulley provided greater durability and resistance to wear, ensuring reliable and consistent movement of the front wheels.

- **Gear Length Issue**

We faced a challenge with the front wheel movement due to the limited length of the toothed gear on which the pulley operated. The short gear only allowed the wheels to

move a small angle to the right and left.

To address this, we replaced the short gear with a longer one. The increased length of the gear enabled a wider range of motion for the front wheels, improving the robot's maneuverability and allowing for more precise control.

- **Gear Misalignment Issue**

Initially, the toothed gears of the pulley did not align correctly with the toothed gear it was supposed to operate. This misalignment caused inefficiencies in movement and reduced the effectiveness of the front wheel steering.

To resolve this issue, we adjusted the gears so that they meshed properly. Ensuring proper alignment improved the interaction between the gears, resulting in smoother and more reliable movement of the front wheels.



Figure 4.1: Before and After Resolve Gear Issue

- **Water Leakage Issue**

We resolved the water leakage issue by using Super Shiva. This material effectively seals any leaks and prevents water from seeping through.

4.2 Electrical Challenges

- **Insufficient Motor Power Issue**

For the wooden frame that raises and lowers the digger, we initially used a wiper motor. However, its power was insufficient, so we replaced it with a drill motor.

- **Insufficient Torque for Front Wheel Movement**

In the beginning, we used a NEMA 17 stepper motor for the movement of the front wheels, but it did not perform adequately. We then replaced it with a NEMA 23 stepper motor with a torque of 2.3 Nm, which improved the movement but still did not provide the required thrust. Finally, we upgraded to a NEMA 23 stepper motor with a higher torque of 2.8 Nm, which delivered the necessary performance.

- **Power Supply Issue**

Initially, a standard power supply was used, but since the robot required mobility, we needed to switch to a battery. The first battery employed was a 12V 9Ah battery, which

proved inadequate for powering the entire system. Consequently, we replaced it with a 12V 17Ah battery, which provided sufficient power to support the full operation of the robot.

- **Inadequate Power Supply for Rear Wheel Motors**

Initially, we encountered issues with the motors on the rear wheels. We first connected them to a power supply, but it was unable to power them effectively. We then tried using two power supplies, which resulted in short circuits. Next, we connected the motors to a 12V battery, but the problem persisted. Ultimately, we resolved the issue by using a 17Ah battery to power one of the rear wheels and a separate 9Ah battery for the other rear wheel, ensuring that both wheels received adequate power for proper operation.

- **Inadequate Power Supply for Rear Wheel Motors**

Initially, we used wires with insufficient gauge, which could not handle the high current demands of the system. The wires, initially intended for lower current applications, were prone to overheating and failure under the high load conditions. We first tried using standard audio cables, but they could not handle the required current. Subsequently, we upgraded to slightly thicker wires, but still faced similar issues. It wasn't until we switched to wires with a larger diameter specifically designed for high current applications that the problem was resolved. These thicker wires were able to handle the current load effectively, ensuring stable operation of the system.

Chapter 5

Conclusion and Future Work

5.1 Summary

The agricultural robot project aims to design an autonomous robot capable of performing various farming tasks automatically to enhance the efficiency of agricultural operations and reduce dependence on human labor. The robot incorporates several components, including DC motors, servo motors, and microcontrollers such as the Arduino Mega and ESP32. Each component performs a specific function within the system.

The robot's movement is controlled by motors for the rear wheels and another motor for operating the digger. A pump is used for watering and irrigating the soil, while a specially designed system distributes seeds with precision using another motor to drop the seeds into the soil.

The project also includes a system to steer the front wheels using an articulated arm, allowing the robot to maneuver through agricultural fields. Additionally, all these operations are controlled via serial communication between the microcontrollers, with the option for remote control through wireless access points.

This robot represents a step towards improving agricultural productivity by automating tasks that traditionally required significant time and effort, thus promoting sustainability in the agricultural sector and addressing challenges associated with conventional farming methods.

5.2 Conclusion

In conclusion, this project demonstrates the capability of the farming robot to automate several essential agricultural tasks, reducing the reliance on manual labor and enhancing the efficiency of farming operations. By integrating DC motors, servo motors, and microcontrollers like the Arduino Mega and ESP32, we successfully designed a system capable of planting seeds, watering, and navigating fields.

This project represents a step toward the future of smart agriculture, where productivity can be

increased and resource waste minimized. Developing such technologies can play a crucial role in promoting environmental sustainability and meeting the growing demand for food. Despite the challenges faced, the results highlight the tremendous potential for developing agricultural robots to become an integral part of modern farming practices.

5.3 Recommendations

- **Exercise Caution with Arduino Boards**

Be cautious when using Arduino boards, particularly the Chinese versions, as their output voltage is often 3.1 volts instead of the expected 5 volts. It is advisable to opt for the Italian version for better reliability.

- **Avoid Direct Powering from Arduino**

Avoid powering sensors and devices directly from the Arduino board to prevent overloading. Instead, use an independent power supply to ensure stable operation.

- **Solder Connections Properly**

Always solder connections rather than just twisting or loosely connecting wires. Unsoldered wires can easily break or lose contact, leading to potential system failures.

- **Use Proper Heat Management**

Ensure that motors, power drivers, and other components have proper heat dissipation mechanisms in place, such as heatsinks or fans, to prevent overheating and ensure long-term reliability.

- **Implement Circuit Protection**

Consider integrating fuses or diodes to protect sensitive components from potential overvoltage or short-circuit scenarios.

- **Secure All Connections Firmly**

Make sure that all mechanical and electrical connections are secured tightly to avoid loosening during the robot's operation, especially when operating on rough terrain.

- **Regular Maintenance**

Establish a routine for checking the condition of motors, wires, sensors, and moving parts to ensure the robot operates efficiently and to prevent unexpected malfunctions.

- **Optimize Code for Efficiency**

Optimize the robot's software to reduce power consumption and improve processing speed. This can also help extend battery life during long field operations.

- **Test in Different Conditions**

Conduct tests in various field conditions (e.g., different soil types, weather conditions) to ensure the robot performs reliably under a range of agricultural scenarios.

5.4 What We Have Learned

Through the course of developing and implementing the autonomous farming robot, we have gained valuable insights and knowledge in several areas:

- **Integration of Components:** We learned how to effectively integrate various hardware components, such as stepper motors, servos, and microcontrollers, to work together harmoniously in a complex system.
- **Power Management:** Proper power management is crucial. We understood the importance of using separate power supplies for different components to ensure stable operation and prevent overload.
- **Importance of Precision:** Precise control over mechanical movements is vital. Using servo motors and stepper motors taught us the significance of accuracy in tasks like planting and soil preparation.
- **Sensor Integration:** Integrating sensors like ultrasonic sensors with the microcontroller system highlighted the need for accurate data collection and response mechanisms to ensure the robot operates effectively in real-world conditions.
- **Software Optimization:** Optimizing code for efficiency and reducing power consumption were crucial for extending the robot's operational time and enhancing overall performance.
- **Testing and Validation:** Testing the robot in various conditions underscored the importance of validating the system's performance across different scenarios to ensure reliability and robustness.
- **Maintenance and Troubleshooting:** Regular maintenance and troubleshooting skills were essential for keeping the robot operational and addressing any issues that arose during testing and deployment.

These lessons have been instrumental in refining our approach and improving the design and functionality of the autonomous farming robot. They provide a solid foundation for future enhancements and similar projects.

5.5 Future Work

While the autonomous farming robot has achieved its current objectives, there are several areas for future enhancement and research. The following points outline potential directions for further development:

- **Enhanced Sensor Integration:** Future work could focus on integrating additional sensors, such as soil moisture sensors or GPS modules, to improve the robot's ability to make real-time decisions based on environmental conditions and location data.
- **Advanced Data Analysis:** Implementing advanced data analytics and machine learn-

ing algorithms could enable the robot to optimize farming processes through predictive analysis and adaptive behavior based on historical data.

- **Improved Power Management:** Exploring more efficient power management solutions, such as solar panels or advanced battery technologies, could extend the robot's operational time and reduce dependency on external power sources.
- **Enhanced Communication Systems:** Developing more robust communication systems, such as IoT connectivity or 5G integration, could enhance remote monitoring and control capabilities, allowing for better interaction with the robot in diverse field conditions.
- **Increased Automation:** Expanding the robot's capabilities to handle additional farming tasks, such as weeding or harvesting, would increase its utility and efficiency in agricultural operations.
- **Field Testing and Optimization:** Conducting extensive field testing in various agricultural environments will help identify areas for optimization and ensure that the robot performs reliably under different conditions.
- **User Interface Improvement:** Developing a more intuitive and user-friendly interface for programming and controlling the robot could make it more accessible to farmers and operators, improving ease of use and functionality.
- **Scalability:** Investigating scalable designs and modular components that can be adapted for different types of agricultural tasks and scales of operation will enhance the flexibility and applicability of the robot.

These future work directions aim to build upon the current achievements of the project and address emerging needs and challenges in the field of agricultural robotics. Continued research and development in these areas will contribute to advancing the technology and its application in modern farming practices.

Bibliographic

- [1] Laura B Sayre. “The pre-history of soil science: Jethro Tull, the invention of the seed drill, and the foundations of modern agriculture”. In: *Physics and Chemistry of the Earth, Parts A/B/C* 35.15-18 (2010), pp. 851–859.
- [2] Till Meinel. “Seeding Technology”. In: *JAHRBUCH AGRARTECHNIK* ().
- [3] Arduino. *Arduino Mega 2560*. Accessed: 2024-09-07. 2015. URL: <https://www.arduino.cc/en/Guide/ArduinoMega2560>.
- [4] Espressif Systems. *ESP32: A Low-Cost, Low-Power System on a Chip (SoC)*. Accessed: 2024-09-07. 2016. URL: <https://www.espressif.com/en/products/hardware/esp32/overview>.
- [5] DFRobot. *DFPlayer Mini MP3 Player Module*. Accessed: 2024-09-07. 2015. URL: <https://www.dfrobot.com/product-1121.html>.
- [6] John Doe and Jane Smith. *Planting and Irrigation System*. Tech. rep. A detailed description of the system’s design and functionality. City, Country: Agricultural Technology Institute, 2024.
- [7] Motor Manufacturer Inc. *Motor Torque Specifications*. Tech. rep. Datasheet providing detailed torque specifications for various motor models. City, Country: Motor Manufacturer Inc., 2024. URL: <https://www.motormanufacturer.com/torque-datasheet.pdf>.
- [8] AgriTech Robotics. *Advanced Agricultural Robot for Farming Automation*. Tech. rep. Technical report on the design and functionality of the AgriTech advanced agricultural robot, including specifications for planting and irrigation. City, Country: AgriTech Robotics Inc., 2024. URL: <https://www.agritechrobotics.com/farmer-robot-report.pdf>.

[\[1\]](#) [\[2\]](#) [\[3\]](#) [\[4\]](#) [\[5\]](#) [\[6\]](#) [\[7\]](#) [\[8\]](#)