

FARMING AUTOMATION USING CABLE DRIVEN PARALLEL ROBOTS

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Mechatronics Engineering

by

Vigneshwar M (Reg. No. 38090021)

Naveen G (Reg. No. 38090015)



**DEPARTMENT OF MECHATRONICS ENGINEERING
SCHOOL OF MECHANICAL ENGINEERING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY - 2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF MECHATRONICS ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Vigneshwar M (38090021)**, **Naveen G (38090015)** who carried out the project entitled "Farming Automation using Cable Driven Parallel Robots" under my supervision from 01.11.2021 to 10.05.2022.

Internal Guide

Mr. V. JAYAPRAKASH, M.Tech., MIE., (Ph.D.)

Head of the Department

Dr. S. PRAKASH, M.E., Ph.D.,

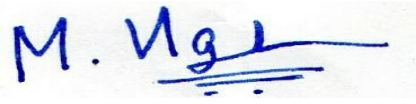
Submitted for Viva voce Examination held on 10.05.2022

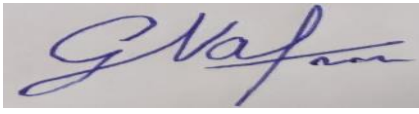
Internal Examiner

External Examiner

DECLARATION

I **Vigneshwar M (38090021)** and **Naveen G (38090015)** hereby declare that the Project Report entitled “Farming Automation using Cable Driven Parallel Robots” done by us under the guidance of **Mr. V. Jayaprakash, M.Tech., MIE., (Ph.D.)**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Mechatronics Engineering.

1) 

2) 

DATE: 10.05.2022

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my sincere thanks to **Dr. S. Prakash, M.E., Ph.D.**, Head of the Department, Dept. of Mechatronics Engineering and Dean of School of Mechanical Engineering, for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mr. V. Jayaprakash, M.Tech., MIE., (Ph.D.)**, for his valuable guidance, suggestions, and constant encouragement that paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Mechatronics Engineering who were helpful in many ways for the completion of the project.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	IV
	LIST OF FIGURES	V
	LIST OF TABLES.....	VI
	LIST OF SYMBOLS AND ABBEREVIATIONS	VII
1	INTRODUCTION	1
	1.1 PRECISION FARMING	1
	1.2 SYSTEM OF RICE INTENSIFICATION (SRI)	2
	1.2.1 Seeder	2
	1.2.2 Weeder	3
	1.2.3 Fertilizer Spreader	4
	1.3 SENSING CROP AND SOIL PROPERTIES, DISEASES, PESTS	5
	1.3.1 Sensing Soil Properties	5
	1.3.2 Sensing paddy disease and pest attack	6
	1.3.2.1 Image Processing	6
	1.3.2.2 Object Detection	6
	1.3.2.3 Machine learning with Tensorflow	6
	1.4 AUTOMATED LOCOMOTION ACROSS THE FIELD	7
	1.4.1 Unmanned Ground Vehicles (UGV)	7
	1.4.2 Unmanned Aerial Vehicles (UAV)	9
	1.4.3 Parallel Robots	9
	1.4.4 Cable Driven Parallel Robots (CDPR)	10
	1.5 ENVIRONMENTAL SPECIFICATIONS	11

	1.6 OBJECTIVE	12
2	LITERATURE SURVEY.....	13
	2.1 INTRODUCTION	13
	2.2 LITERATURE SURVEY	13
3	AIM AND SCOPE	19
	3.1 AIM OF THE PROJECT	19
	3.2 SCOPE OF THE PROJECT	19
4	MATERIALS AND EXPERIMENTAL METHOD.....	20
	4.1 METHODOLOGY	20
	4.2 DESIGN OBJECTIVE	21
	4.3 MATERIAL USED	21
	4.4 CABLE DRIVEN PLATFORM DESIGN	21
	4.4.1 Control algorithm for CDPR locomotion	22
	4.5 ISOLATION MECHANISM	24
	4.6 ATTACHMENT DESIGN	25
	4.6.1 Seeder	25
	4.6.2 Weeder	26
	4.6.3 Fertilizer Spreader	28
	4.6.4 Phenotyping	29
5	RESULTS AND DISCUSSION	31
	5.1 INTRODUCTION	31
	5.2 PLATFORM TRAJECTORY	31
	5.3 ATTACHMENT EFFICIENCY	34
	5.4 FURTHER DEVELOPMENT	35
6	SUMMARY AND CONCLUSION.....	36
	6.1 SUMMARY	36

6.2 CONCLUSION	36
REFERENCES	37
APPENDIX A	40
APPENDIX B	53

ABSTRACT

Agricultural automation is a field that integrates equipment, infotronics, and precision farming technologies to create viable solutions for challenges facing the food needs of the human race now and into the future. A vast majority of semi-automated solutions for agricultural processes are implemented as custom made attachments to tractors or smaller diesel powered devices. But data analytics of crop growth, disease, moisture content across the field are automated only at certain sophisticated farms. This project proposes an agricultural automation that combines precision farming mechanisms and phenotyping equipment with Cable Driven Parallel Robots. We have built a prototype of the proposed system to identify practical requirements to build such a system and to propose methods of improvement. The prototype is meant to automate Seeding, Weeding, Fertilizer spreading and Image acquisition for a paddy field. The agricultural practices were derived from SRI system. This choice made to cover the most cultivated crop in our state, Tamil Nadu, India.

LIST OF FIGURES

FIGURE No	TITLE	PAGE No.
4.1	Cable Driven Platform	22
4.2	Seeder Mechanism	26
4.3	Seeder Mechanism With CDPR	26
4.4	Isometric view of Seeder	27
4.5	Spiked Wheels Of Weeder	27
4.6	Vanes and Drum of Fertilizer Spreader	28
4.7	Fertilizer Spreader Mechanism With CDPR	28
4.8	Paddy Disease and Pest Identification	30
4.9	Disease Data JSON	30

LIST OF TABLES

TABLE No	TITLE	PAGE No.
5.1	Error Variation With Respect To Travel Distance	33
5.2	Error Variation With Respect To Z Axis	34
5.3	Error Variation With Respect To X Axis	34

LIST OF SYMBOLS AND ABBEREVIATIONS

SRI	System of Rice Intensification
TNAU	Tamil Nadu Agricultural University
CDPR	Cable Driven Parallel Robot
ML	Machine Learning
UGV	Unmanned Ground Vehicle
UAV	Unmanned Aerial Vehicle
RPM	Rotations per Minute
PWM	Pulse Width Modulation
IOT	Internet of Things
RADAR	Radio Detection and Ranging
LIDAR	Light Detection and Ranging
CCD	Charged Coupled Device
VTOL	Vertical Take Off and Landing
NPK	Nitrogen (N), Phosphorus (P), Potassium (K)
RHEA	Robot fleets for Highly Effective Agriculture
GPS	Global Positioning System
GIS	Global Information System
JSON	JavaScript Object Notation

CHAPTER 1

INTRODUCTION

1.1 PRECISION FARMING

Precision agriculture is a farm management philosophy that incorporates modern sensors and automated equipment to perform well informed agricultural actions [14]. The farm is logically divided into grids. In each grid, Variations in crop and soil properties are sensed continually while traversing the area. All this dense information is then interpolated to assign general values to the measured properties, for that particular grid. These grids' locations are tracked with the Geographical Positioning System (GPS). Interpolated values are mapped to the appropriate location in the Geographical Information System (GIS).

These values can then be compared to recommended values, using which, we can calculate the amount of fertilizer/water to be supplied to that place, provided that we know the concentration of nutrients that fertilizer contains. The variable supply of fertilizer is done by automated equipment, controlled by sophisticated control systems. This is expected to increase crop yield, its quality, thereby profits. It will also reduce the redundancy of fertilizer, thereby reducing the investment. Water usage can also be regulated for low rainfall situations.

Several tools for agricultural tasks had been proposed and used. These include Manual, Semi Automated and Fully automated equipment. Operations such as Tillage and Harvest are best done by large scale machinery attached to and operated through tractors. These processes allow flexible trajectory along the field and so it is convenient to deploy large vehicles like tractors. But the intermediate processes such as seeding, weeding, fertilizer spreading and data collection requires maintaining a constrained trajectory through the field. This calls for alternatives such as Manual operation, overhead cranes [3], small enough Unmanned Ground Vehicles (UGV) and in some cases drones.

1.2 SYSTEM OF RICE INTENSIFICATION (SRI)

As this project's target scope is paddy cultivation in the region of Tamil Nadu, we are closely following the recommendations from Tamil Nadu Agricultural University (TNAU). TNAU recommends System of Rice Intensification (SRI). SRI is an alternative farming methodology for increasing the productivity of irrigated or rain-fed rice cultivation by management of plants, soil, water and nutrients in favour of promoting greater root growth [9]. SRI is not a well-defined technology but a set of alternative practices rooting from Madagascar. Major steps endorsed by this methodology are: Square Planting (distance between rows is equal to distance between columns), Lower level of water puddling and Mechanical weeder usage. This method is also applicable for direct seeding. Conventionally seeds are sown in a smaller nursery. After 14 days, they are transplanted into rows across the field. But direct seeding suggests to bypass this transplanting phase and directly sow the seeds in rows all over the field. This is recorded to reduce the usage of seeds from conventional 80kg/hectare to around 15 kg/hectare. Also it reduces damage and loss of seedlings while transplanting. But it also requires frequent weeding and strict water management.

This system is observed to produce increased yields, using reduced seed requirements and less water puddling (40% reduction from conventional methods). Along with its advantages, SRI also comes with few limitations. Major setback is the requirement of high labour work (and subsequent investment) in the initial phases of cultivation, as levelling the soil and seeding has to meet specified requirements. But overall cost is reduced compared to conventional methods.

1.2.1 Seeder

Seeder mechanizes the process of sowing the seeds. Sowing a nursery for transplantation later is easily done by manual sowing. Contrary to sowing in a nursery, direct seeding requires the mechanism to ensure proper spacing between rows and columns. Direct seeding can be done for both dry and wet land conditions. For direct dry seeding, seed drills such as Turbo Happy Driller is used on coarse dry soil [23]. It consists of disc pairs that penetrate in soil, creating a furrow as the unit moves on

along with a tractor. The seeds from a hopper are dropped through a tube at regular intervals, in between the discs to be guided into the furrows [23].

Direct wet seeding is done on wet soil with a thin film of water over it. The field is ensured to be levelled. A Drum seeder mechanism is recommended by TNAU for this process [28]. The drum seeder has several drums, filled with seeds, mounted to a common shaft. Lugged wheels of diameter larger than the drums are fitted to both sides along with bars for dragging the unit. The drums are perforated with the regularly spaced holes on the surface, from where the seeds will fall. As the seeder is dragged over the field the drum rotates. Along with the rotations, the seeds fall on the surface of soil. This is known as Hill Dropping. The regular spacing of the holes ensures the distance between the seedlings is maintained. Number of seeds released is controlled by the hole size (9mm) [28].

As this project does not focus on developing standalone mechanisms but on incorporating all such mechanisms into a common platform, this project restricts itself with implementing only one seeder mechanism, for only one method and type of water maintenance - Direct Wet Seeding.

1.2.2 Weeder

Weeding is an important process of removal of unwanted plants in the field. TNAU. In wet seeded rice, recommendations from TNAU advises weeding to be done with mechanical weeders at 10, 20 and 30 Days after sowing. SRI system endorses the use of mechanical weeders instead of traditional manual weeding. Mechanical weeders are either manually operated like Cono Weeder or power operated with manual guidance. A square plantation allows the weeder to be operated in between rows and columns of the plants. General design objective of a weeder is to till the soil and disrupt the weeds in the path of movement.

Important design consideration of a cono weeder seems to be the weight of the unit and Draft force required by the mechanism. Draft force is the force that has to be applied to the unit, to overcome the soil resistance to be tilled and soil friction due to the weight of the weeder. Other considerations include the ergonomics of the design. Depth of tillage was maintained at 3cm for widely used Cono Weeder.

A cono weeder is manually operated. This weeder has two cone shaped drums with serrated blades. The blades are serrated to reduce the surface area that contacts the soil. This reduces the soil resistance. A float is used to prevent the weeder from sinking into the mud. All of these are attached to an inclined bar/steering shaft. The bar is fitted with a handle bar on the top end, for steering. TNAU Cono weeders were recorded to require a draft force of 30 kgf (295 N) [18]. A modified version of the same was recorded to have a draft force 18 kgf (180 N) [18]. This is due to changing the profile of the blade and resizing the dimensions of float while keeping the surface area same. Best observed efficiency of these weeders were recorded as 87.96%.

An automated version of cono weeder driven by battery power also exists. This version was recorded to exhibit an efficiency of 95% in weeding [27].

A power rotary weeder has a cylindrical wheel attached with serrated blades on its outer surface. Powered by IC engines of around 3 KW - 6 KW capacity, these were recorded to require 20 kgf (200 N) of draft force on an average, per wheel [27]. There is a difference of power capacity as different attempts used different power due to varying numbers of rows affected at once.

1.2.3 Fertilizer Spreader

Fertilizers are granular particles that are traditionally spread around the field by hand. The same can be automated by a relatively simple mechanism known as centrifugal disc spreader [2]. It contains a hopper that supplies the fertilizer. A disc on which a controlled amount of fertilizer is dropped from the hopper, spins at a set speed. Due to this rotation, the fertilizer particles on the disc experience centrifugal force. They are thrown out of disc within a constant range, proportional to motor speed. The disk contains vanes that guide the fertilizer particles. The amount of fertilizer that is spread in a particular spot can be controlled approximately by controlling the speed in which the spreader unit moves away from the spot or by controlling the amount of fertilizer discharge from the hopper itself.

1.3 SENSING CROP AND SOIL PROPERTIES, DISEASES, PESTS

1.3.1 Sensing Soil Properties

Agricultural Sensors are used to Sense soil properties such as Moisture content, Nutrient concentration and Plant properties such as chlorophyll content, disease or pest attack. For paddy cultivation, 16 nutrients are found to be essential. These Nutrients are Carbon, Hydrogen, Oxygen, Iron, Manganese, Copper, Nitrogen, Phosphorus, Potassium, Zinc, Boron, Molybdenum, Calcium, Magnesium, Sulphur and Chlorine. Among these, Nitrogen, Phosphorus and Potassium are known as primary plant nutrients. Calcium, Magnesium and Sulphur as secondary nutrient elements. These are major nutrients while the other elements are micronutrients. This classification is drawn out of the amount of concentration required. The importance of all the nutrients are the same. Various sensors for measuring these nutrient concentrations were developed so that necessary amounts of fertilizer can be used on the soil. This project requires sensors that are capable of sensing in real time with good fidelity.

Capacitive Soil Moisture sensor is already in use. It is a capacitive sensor with fork-like geometry which will be the two conductors of the capacitor, while the soil acts as the dielectric separator [5]. The capacitance of the capacitor changes with the bulk volume of water in that soil. This relation is used to estimate the moisture content with an error less than 5%.

NPK sensors are used to measure the concentration of primary nutrients - Nitrogen (N), Phosphorous (P), and Potassium (K). There are various principles based on which these sensors can be made. A conductivity measurement technique uses capacitive platinum probes and soil as the dielectric. A.C. voltage is supplied through probes. The varying current gives varying conductivity which is used to estimate concentration of N, P and K concentrations. An Electrochemical method uses Ion Selective Electrodes (ISE) and Ion Selective Field Effect Transistor (ISFET) [8]. Ion selective membranes react with particular target ions in the soil, in a reaction similar to electrolysis. Current flow is measured and the proportional ion concentration is calculated. An optical method uses Spectroscopy to analyse variations in how incident

LASER / IR / UV radiation of varying frequencies gets absorbed. These are accurate but time consuming and costly.

1.3.2 Sensing paddy disease and pest attack

Object detection for agricultural monitoring using Machine Learning models and Image processing is an active area of development. A vast range of image processing techniques, different ML model architectures, labelled image datasets under different circumstances, ML frameworks are being used to develop ML models to detect and classify various paddy diseases and pest attacks.

1.3.2.1 Image Processing

The term Image processing means the use of digital computers to process digital images through an algorithm. End goals of such a process will be to enhance the image quality, modify the image, identify features and mask parts of importance. It is used in Machine Learning (ML) models to prepare the image to a suitable format for further Object detection.

1.3.2.2 Object Detection

Object detection is related to computer vision and image processing. It deals with detecting instances of semantic objects of predefined classes. This is done by the use of Machine Learning or Deep Learning with Neural networks. Machine Learning is the traditional approach to Artificial Intelligence, whereas Deep Learning is a relatively newer technology.

1.3.2.3 Machine learning with Tensorflow

Tensorflow is an open source platform for machine learning from the Google Brain team. It offers an ecosystem of tools and libraries that can be used to develop state-of-the-art ML models. Machine learning (ML) models are algorithms that analyze data, learn from it and make predictions on the learned insights. Training an ML model requires a set of data to be trained on and an objective function to fine tune the predictions based on calculated errors. For object detection, the flow of tasks in

training a ML model goes as follows. Large amounts of Training data (in this case, images) are collected in a common database. These images are segregated for training and validation. The parts of the image which are expected to be identified by the ML model are marked and labelled with a logical class name. The Tensorflow library is set to iterate over the labelled images to build a set of rules which will be used later to detect objects on it's own. This collection of rules and logic is known as the ML model. There are various types of model architectures that tensorflow can train and build from. These model architectures offer varying levels of speed of execution and accuracy. We can pick a model type for our use case from the tensorflow model zoo. The trained model is validated against a part of the labelled data. An objective function calculates the error in the predictions of the model. Rules of the model are adjusted to minimize this error as much as possible. Once the model attains a satisfactory level of confidence on predictions, the model graph is freezed and converted to different formats to be incorporated into external projects. ".tfod" is a tensorflow object detection format to which these models can be saved. For these models to work on embedded systems, models need to ensure faster execution and lightweight. "tflite" is a light weighted format to store ML models. It is meant to be deployed onto mobile phones and IOT devices. Choosing right architectures such as Mobile Net can ensure faster, nearly real time detection with good accuracy. Machine Learning algorithms are essentially optimization algorithms. The accuracy of predictions made by such a model grows with the size of the dataset it is trained on.

1.4 AUTOMATED LOCOMOTION ACROSS THE FIELD

1.4.1 Unmanned Ground Vehicles (UGV)

Unmanned ground vehicles are autonomous vehicles that operate while in contact with ground. It is capable of automatically navigating through the field, responding to its surroundings and executing desired operations. UGVs are designed especially to be compact enough to navigate through rows of crops without disrupting them. Design of such UGVs typically starts with tractor design as a reference point. Some notable attempts at UGV for weed control and targeted spraying are BoniRob, AgBot II, Autonome Roboter, Hortibot, Kongskilde Robotti, RIPPA, Spray robot - a

narrower body design, etc [15]. These robots spray herbicides that selectively inhibit weed growth. Stability due to ground contact is the main advantage of UGV. For achieving the stability and to be compatible with the row structure of plants, an UGV has to have a minimum of four legs. The body of the vehicle has to stay above the maximum height of crop, to avoid disturbance. As the crop reaches harvesting stage, due to increased height clearance requirement of the body, the center of gravity of the vehicle increases. To ensure stability of the vehicle, the wheelbase has to be increased. This increases system size and the power consumption.

All of these robots are equipped with machine vision systems. Various image sensors such as Charge Coupled Device (CCD) sensor, Near InfraRed (NIR) Imaging sensor, etc [15]. Digital Images thus captured are processed by ML models to detect plants and separately detect weeds, fruits, diseases and pest attacks.

Field scouting and data collection robots are used for data acquisition. Several successful attempts had been made with different form factor and variety sensors. Trimbot 2020, Wall-Ye, Ladybird, MARS, SMP S4, VineRobot, HV-100 Nursery bot, VinBot, Mantis, GRAPE, OSCAR field survey robot, Husky UGV are some of notable field scouting and data acquisition UGVs [15]. Common functionalities of these systems include 3D Mapping for the field, Rendering point clouds, thermal imaging, Object detection, Soil property sensing. Sensors such as RADAR, LiDAR, panospheric, stereovision, thermal cameras and other soil sensors.

Systems such as the SWEEPER robot [26] are equipped with serial manipulators and grippers for harvesting in horticulture fields (fruits and plants cultivation). Along with Sweeper, other harvesting UGVs include Harvey, CROPS, Energid citrus picking system, Citrus robot, DogTooth, Shibuya Seiki, UR5 apple robot [15].

A relatively newer approach is to use a fleet of smaller robots throughout the farm [13]. These smaller robots can traverse through individual rows and are less disruptive to the plants. Smaller payload capacity of these robots makes it necessary for multiple robots to be engaged in a single operation. An example of this type of robot is Tertill - a solar powered compact robot.

1.4.2 Unmanned Aerial Vehicles (UAV)

Unmanned Aerial vehicles are autonomous vehicles that operate without ground contact. Drones are commonly used UAV systems. Common UAV platform configurations are recognised to be Fixed Wing, Helicopters, Muticoptors (Drones) [4]. In the agricultural context, usage of Multicopters, especially micro drones of payload varying from 10kg to 30kg has been steadily increasing [6]. A number of review works on these subjects identify the best applications of drones in agriculture as Seeding, Fertilizer and Pesticide spreading, Soil sampling, Image data acquisition and Crop monitoring purposes. Microdrones with vertical take-off and landing (VTOL) [4] are used throughout the project RHEA [21,22] and project Flourish [20] for image acquisition for aerial maps. These drones were incorporated with a GPS module for navigation and mapping the collected data to location. Stable drones with a payload of 10kg to 30 kg are available in prices ranging anywhere from 2 lakh INR to 8 lakh INR. Drones offer functionalities that UGVs are incapable of, but at comparable cost to that of tractors. Drones are designed to be battery operated, with a common flight time of 30 minutes. This requires open charging stations to be installed across the field for the drone to recharge itself. An important part of a UAV is a gimbal system between the actual drone platform and the attachment. This effectively isolates any imbalance in the orientation of the drone platform and keeps the attachment in the correct orientation. This is important in order to ensure the expected operation is actually realized. Drones cause the air beneath it to spin, which can be disruptive if they operate closer to the crops. This makes them unsuitable for operations such as wedding and seeding in proper distancing, which requires the drone to fly low.

1.4.3 Parallel Robots

A very common type of robotic manipulators is Serial manipulator which is a open kinematic chain with actuators to control the joint variables. The joint variables together affect the end effector's position and orientation. Serial manipulators come with its weaknesses such as small payload to robot weight ratio, poor positioning accuracy, lower speed and acceleration. Another lesser common, but more accurate

type of manipulator is Parallel manipulator. Parallel manipulators have independent kinematic chains connected to the end effector, forming a closed kinematic chain. This structure redistributes the load closer to the individual actuators. Due to this, the links can be comparatively lighter owing to lower stiffness requirements. Positioning errors are also brought down as the individual joint errors are not stacking upon each other. Higher velocity and acceleration are also possible. The Gough platform uses a parallel manipulator set up to change the position and orientation of a hexagonal platform. There are 6 links with linear actuators. These joints are connected to the platform using a spherical joint. Some applications of the Gough platform are flight simulation training, positioning in assembly of micro devices, surgery robots, etc. Delta robot from ABB robotics is an industrial Pick and Place robot with high positional accuracy at higher accelerations such as 100 m/s^2 .

1.4.4 Cable Driven Parallel Robots (CDPR)

CDPR is a parallel robot in which rigid kinematic chains are replaced with cables. This allows for larger working areas. The end effectors position and orientation is controlled by the length of the cables. The setup can be easily transported and installed at required locations. Cables such as Kevlar cables shall have high tensile strength while having less weight per unit length. Due to their flexible nature, interference and joint limit problems are overcome. They are comparatively light weighted and easy to attach to the end effector. First of its kind was a robot called RoboCrane, developed by NIST. It had all 6 degrees of freedom. It was used for paint stripping of airplanes. Another use case was lifting shipping containers, proposed by the Automated All Weather Cargo Transfer System (AACTS) project and the U.S Navy. A very noticeable use of CDPR is in SpiderCams [25] used in sports stadiums for over the head video coverage. SpiderCam is the company behind this product. Their system follows a general component structure that is closer to this project. It consists of a CDPR platform, a motorized gimbal mechanism, to which the Camera (attachment) is attached. The oscillations, positional and orientation inaccuracies in the platform are separated from the attachment through the gimbal. The entire system is remotely controlled from a dashboard.

CDPR comes with its own complications too. The use of cables converts this into a nonlinear kinematic problem [1]. This is due its flexibility and its nature to sag [19]. This also brings upon oscillation before reaching a steady state. Several implementations of CDPRs overcome oscillations up to a satisfactory extent with the use of redundant cables.

Attempts had been made to use CDPRs in plant phenotyping systems. Swiss Federal Institute of Technology in partnership with SpiderCam company, developed a CDPR that carries a platform with phenotyping sensors and equipment [12]. It is an over-constrained system that has 3 degrees of freedom supported by 8 cables. The use of 8 cables were said to prevent pitching of the system toward the center of the field. Pitching is observed due to the uneven distribution of load and change in direction of cable tension due to cable sag.

1.5 ENVIRONMENTAL SPECIFICATIONS

This project discusses incorporating attachable precision farming automations and data acquisition systems into Cable Driven Parallel Robots (CDPR) for automation of paddy cultivation. The project demonstrates the viability of the proposed system by implementing it for a particular type of farm, soil, crop and seeding mechanism. This choice was made to cover as much percentile of agricultural land as possible in Tamil Nadu. Paddy occupies the highest hectares of agricultural land (17,21,265 ha as of 2018-2019 period) than any other crop cultivated in Tamil Nadu [24]. Hence, paddy is chosen. The system assumes direct wet seeding. Soil types include Red Loamy sand (53%), Clay (22%) and Sandy soil (18%) [24]. It can be seen that Majority of the soil types have good water retaining capacity and so good rigidity. Rigid soil has higher soil resistance to tillage. This requires a higher draft force for weeding than other types of soils.

1.6 OBJECTIVE

The objective of this project is to incorporate precision farming mechanisms into CDPR, evaluate their viability and adaptability and lay a basic component structure to expand further on this system.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

Methods of Rice Cultivation and various attempts to automate outdoor farming processes had been reviewed. Generally the operations are carried with Unmanned Ground Vehicles while the surveillance, data collection are done with Unmanned Aerial Vehicles. Swarm of robots connected to a common Internet of Things seems to be adapted in various forms.

2.2 LITERATURE SURVEY

Amir Moradi, (2013), (1), This thesis presents kinematic equations and failure analysis of a 3 DOF cable driven parallel robot system. Jacobian relation between cable length, angle and platform velocities was derived.

Artur Przywara, (2015), (2), This is a study based on centrifugal disc spreader for fertilizer spreading. It discusses the effect of factors such as fertilizer properties, feed rate, disc rotation speed, feed position of fertilizer on the disc, etc., on the spatial distribution variability of the fertilizer. We inferred that spatial distribution relies more on the vane design and the feed position on the disc.

Chetan Dwarkani M et.al., (2015), (3), A overhead crane based automation was proposed in this paper. The system is meant to be a permanent fixture. The land is populated with soil and plant sensors, which regularly update the field variable to a central system. Irrigation and fertilizer spraying equipment mounted on an overhead crane is operated based on the sensed data. Variable amounts of water and chemicals are sprayed over localized areas marked using GPS. The structure can only have a rectangular workspace and it is permanent once it is laid out.

Dileep M R, (2020), (4), A study on different types of UAV, UAV based agricultural automations, Companies that manufacture these automations. Comparisons are drawn between automations that are based on different types of UAV. General design recommendations were laid down for such a system, which were observed to be common in different systems.

H. Eller, et.al., (1995), (5), A capacitive sensor setup was proposed and tested for measuring soil moisture content. The system uses 32 MHz A.C. voltage supplied to 2 fork-like conductor probes that are dipped into soil. While the soil acts as dielectric in the capacitor, the system measures the permittivity of the soil which is influenced by the moisture content in the soil. Sensed values were found to be satisfactorily accurate with errors less than 5%.

Francesco Marinello, et.al., (2016), (6), This paper analyses technical sheets of over 250 models available for providing an understanding of the applicability of UAV in agricultural automation. It describes different types of UAVs, lists and compares the general components of UAV based automations and their performance statistics.

James Lowenberg-DeBoer, (2019), (7), This study is a review of economic aspects of research on mechatronic automation and robotics for agricultural processes. Budgetary evaluations of the systems proposed after 1990 were re-evaluated to find out the budgets were partial and assumed a lot of expenditure as already made. Bases on these, conclusions were drawn on whether systems were actually profitable or not along with recommendations for budgetary evaluation.

Jianhan Lin, et. al., (2007), (8), This paper discusses Electrochemical methods of measuring N, P, K content in soil. Working and efficiency of Ion Selective Electrodes and Ion Selective Field Effect Transistor combination were presented.

H. De Laulanié, (2011), (9) Observations and Records of System of Rice Intensification (SRI) practices from Madagascar dating back to 1993. This paper was refined and reviewed over years. This paper serves as a standard reference for this methodology all over the world.

Laxmi C, et. at., (2017), (10) This paper reviews various Major Nutrients (N, P, K) sensing methods. Conductive Measurement technique, Electrochemical Method and Optical Method to measure N, P, K concentration in soil were discussed. Their time and cost requirements were also discussed.

S. Meivel, (2020), (11), This paper discusses the general workflow of image acquisition, processing and mapping it to a GIS system with the use of GPS. Normalized difference Vegetation Index (NDVI), Near-InfraRed band (NIR) sensors and Thermal Camera were used for image capture. Various irrigation properties of the farm were identified with a range of Indexes maps, deduced from the captured images. Indexes were - VCI for drought map, TVI for age range, SAVI, PVI for soil lines map from which leaf area index can be deduced, SBI for greenness/yellowness map. Entire process was demonstrated using python. These Indexes were proposed to be fed to deep learning neural networks to predict management strategies.

Norbert Kirchgessner et.al., (2016), (12), This paper discusses the usage of cable driven platforms for phenotyping in the ETH project. Payload was 12 kg, operating over 2-5 meters above the canopy.

Pablo Gonzalez-de-Santos, (2016), (13), The RHEA fleet of compact UGVs and UAVs for variable herbicide spraying was discussed in detail. Various tooling strategies for spraying were discussed such as: Patch sprayer (to cut down herbicide usage upto 75%), Canopy Sprayer (to cut down usage upto 50%) and Mechanical/Thermal tools. Weed patches are identified using the images acquired by UAV drones. The UGVs take care of spraying the herbicide. All the robots are connected to a central system via a Linux-based wireless router that implements TCP/UDP API, ZigBee pro interface, ITU GPRS, 802.11b/g, 802.11a interfaces. Weed detection with a 90% accuracy target was done with a system that contains both color and NIR images to get an unique 4 channel image.

Rahul Tripathi., et al. (2013), (14), This paper defined Precision agriculture for an Indian context. Discussed the motives, objectives, and components involved in developing such a system. Such a precise farming system is expected to have: sensor probes that can measure crop and soil properties/disease attacks; GPS system for

location mapping; a GIS system that will collect all this sensed data, interpolate them for localized divisions of farm and store them. Based on this information, automated equipment can release variable rates of fertilizer/water/insecticides across the field.

Redmond Ramin Shamshiri et. al., (2018), (15), A detailed review of application of UGVs in automation of agricultural processes. It listed various UGV robots for three purposes: weed control and targeted spraying robots, Field scouting and data collection robots and Harvesting robots. A general pattern was deduced from the functionalities of the listed robots. Various form factors, sizes, manipulators, Machine vision and Artificial Intelligence systems were reviewed.

Rendong Nan et.al., (2011), (16), This paper discusses the design FAST project from Japan. It is a 500 meter aperture spherical radio telescope. The section of interest for this project is the cable suspended cabin above the reflective dish. The cable was suspended and controlled using 6 cables. Power was supplied to the cabin via power cable from one of the poles, loosely coupled with the drive cable.

Selvakumar Manickam, (2021), (17), A system for smart agriculture using drones connected to an IOT network was discussed. The drones were set to perform soil sampling, seed broadcasting, pesticide/fertilizer spraying and crop monitoring. A Zigbee type network was used for connection. The project was not fabricated/simulated and tested, but it built upon previously tested systems that showed some improvement in yield.

Er. H.B.Shakya, et. al., (2016), (18) This paper proposes a modified form of TNAU Cono Weeder which has reduced draft force requirements. This paper presents performance evaluation results of experiments conducted following RNAM test code and ISI code: 7927-197. It was recorded that TNAU cono weeder demands a mean draft of 29.12 kgf whereas the modified version showed reduced mean draft of 17.94 kgf. Weeding efficiency varied was 77.41% and 87.77% respectively.

Tuong Phuoc Tho, et.al., (2021), (19) This paper discusses the catenary and straight line based equations for calculating cable sag. Then, a method based on ANFIS (adaptive neuro-fuzzy inference system) architecture for predicting the cable

sag based on perceived force distribution. The algorithm produced results with small enough errors to be incorporated into the CDPR control algorithm.

Flourish Project (2015-2018), (20), The Flourish project was funded by the European Community's Horizon 2020 programme. Its objective was to develop new methods of sustainable farming that will increase the yield while decreasing chemical inputs. This required constant monitoring and automated action. Their project proposed a combination of UGV and UAV automations connected together to centralized servers. Over 50 research papers were published as a part of this project on various sub topics such as crop/field feature detection using CNN, Weed map detection using deep neural networks, Image acquisition, positioning system, UGV based automation, 3D mapping of field using UAVs, phenotyping sensors, etc.,

RHEA Project (2012 - 2015), (22), RHEA stands for Robotics and Associated High Technologies and Equipment for agriculture. This project was funded by the European Commission to design, develop and test automatic robotic systems for effective weed management of various crops including cereals. It aims at decreasing the chemical inputs by 75%. The project successfully demonstrated a final system in 2015, which consisted of a fleet of small heterogeneous robots - both grounded and aerial, equipped with advanced sensors, end effectors and decision control algorithms.

Rice Knowledge Bank - International Rice Research Institute, (23), A detailed collection of paddy cultivation guides, different methods, automations were made available on this website. Methods were particularly focussed on low cost solutions applicable for Asian and African regions.

Tamil Nadu Agricultural University - Agritech Portal, (27), This portal serves as a repository of recommendations from TNAU for Tamil Nadu farmers. Information on a variety of crops were provided including paddy. This website served as our source on System of Rice Cultivation (SRI) Cultivation, Seeder and Weeder Automations, Statistics of crop distribution across Tamil Nadu. Soil and Land Statistics of Tamil Nadu agricultural land are accessible from the Tamil Nadu Agricultural department.

TNAU Paddy drum seeder specification document, (28), A Detailed specification of well known TNAU drum seeder. Seed metering is done with orifice diameter to 9mm. Spacing is set to hill plant the seeds at a distance of 20mm on all sides.

CHAPTER 3

AIM AND SCOPE

3.1 AIM OF THE PROJECT

The Aim of this project is to evaluate the viability of a system that combines Cable Driven Parallel Robots (CDPR) and Precision farming mechanisms for automation of Seeding, Weeding and Fertilizer spraying.

To assemble a mini prototype of the proposed system and record the advantages, capability, limitations, sources of errors of such a system.

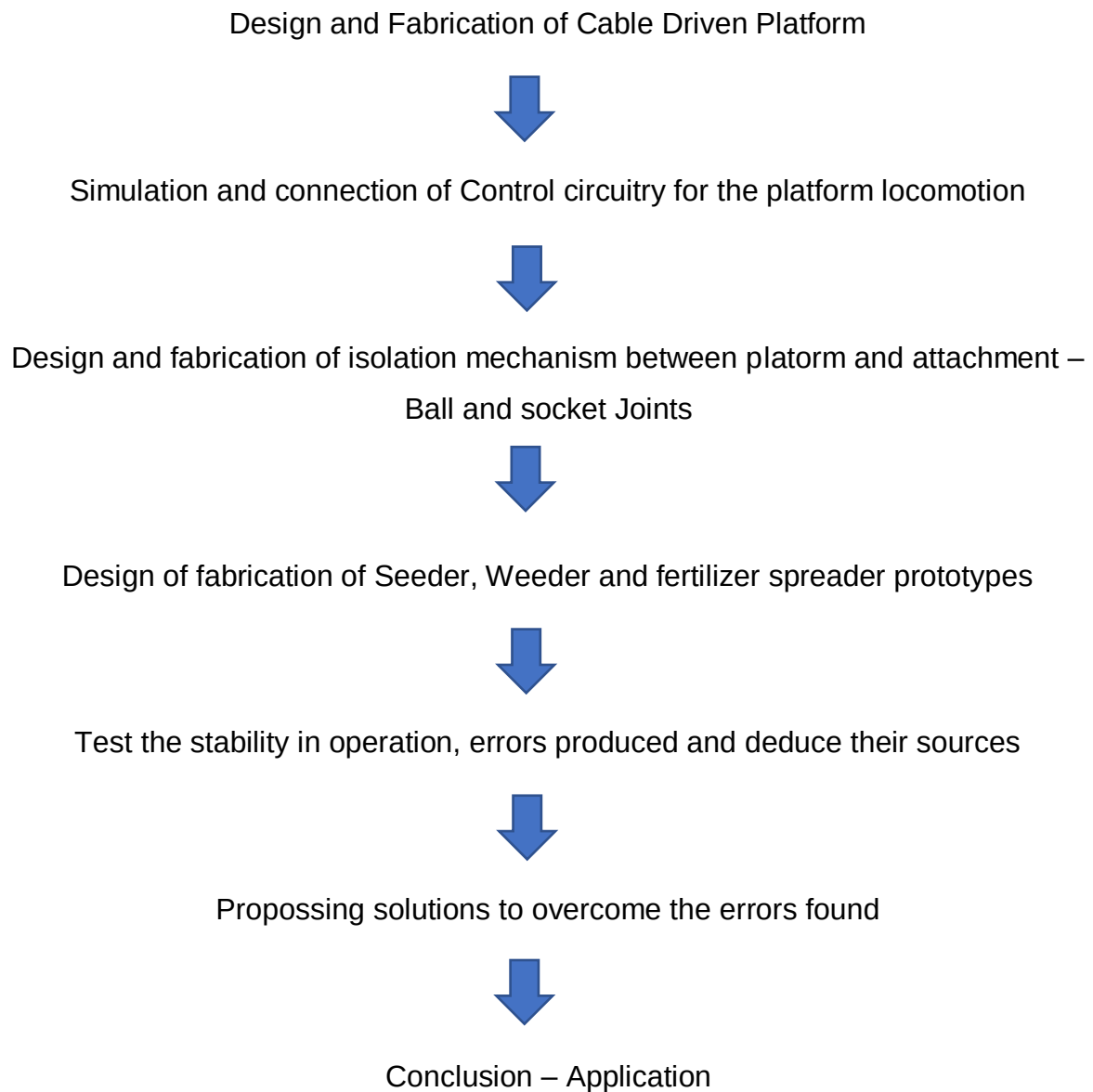
3.2 SCOPE OF THE PROJECT

The project contributes to the agricultural automation research. CDPR systems can be well adapted for vast outdoor areas. Quadrilateral field areas can also be easily operated upon using CDPRs. This prototype architecture is kept open to newer implementation of subsystems, so that the finding about sources of errors in control and operation stays at least partially relevant, for future development.

CHAPTER 4

MATERIALS AND EXPERIMENTAL METHOD

4.1 METHODOLOGY



4.2 DESIGN OBJECTIVE

A prototype is made as a part of this project. This project tends to keep the system "Implementation agnostic". Meaning that the individual implementation of the components mentioned above can vary depending upon the problem constraints. But they all are expected to cohesively come together in the same architecture specified. The system architecture itself is treated as abstract as possible. We have implemented each component with the simplest implementation we could make. Upon scaling up, these mechanisms may be replaced with sophisticated ones which are better suited for that context. We have also mentioned what mechanisms could be better suited for general purposes at each section.

4.3 MATERIAL USED

Polylactic Acid (PLA) material manufactured by Fila-Tech is used for 3D printing all the components for this project. This PLA has a tensile strength of 55 MPa, and density of 1.24 g/cm³. The maximum dimensions that can be printed in the 3D printer that is available at our disposal proved to be a constraint in our prototype designs.

4.4 CABLE DRIVEN PLATFORM DESIGN

The cable driven platform is based on a square frame of dimensions 19cm×19cm (due to the maximum size of the 3D printer we used). Provisions for mounting motor, gears and cable reel are made on each of 4 corners. The reel is set up to release the cable along the diagonal direction of the frame.

Four 775 motors were used. The motors operate at a rated voltage of 12V and draw a non-load current of 20A DC. In specified input, the motor is rated to produce 6400 RPM and a torque up to 1 kg cm. The motors are attached to the cable reel through a worm gear setup. This ensures the cable does not release from the reel when the platform is still or when the motors are not engaged. The worm gear has a gear ratio of 40:5. Arduino mega 2560 is used as a controller. The board is mounted on the middle of the frame. Each motor is connected to Arduino via BTS7906 H-Bridge motor driver. The H bridge has 2 BTS7906 chips, one for right rotation (releasing the

cable) and the other for left rotation (retracting the cable). Speed of the motors can be controlled by the using pulse width modulation (PWM) of power MOSFET within the chip. We can directly access the 16 bit timers within Arduino's ATMEGA processor to unlock a 2^{16} possible values of duty cycle. This will help to fine tune the speed of the motors.

Power is supplied from external step down convertors via wires as found in the CDPR of the FAST telescope project [15]. Power source provides 12V and 20 A. This could also be replaced with batteries. It is beneficial to leave the battery on ground rather than including it with a platform. This will cut down the necessity to halt the operation for recharging. The power cable could be loosely coupled together with one of controlling cables.

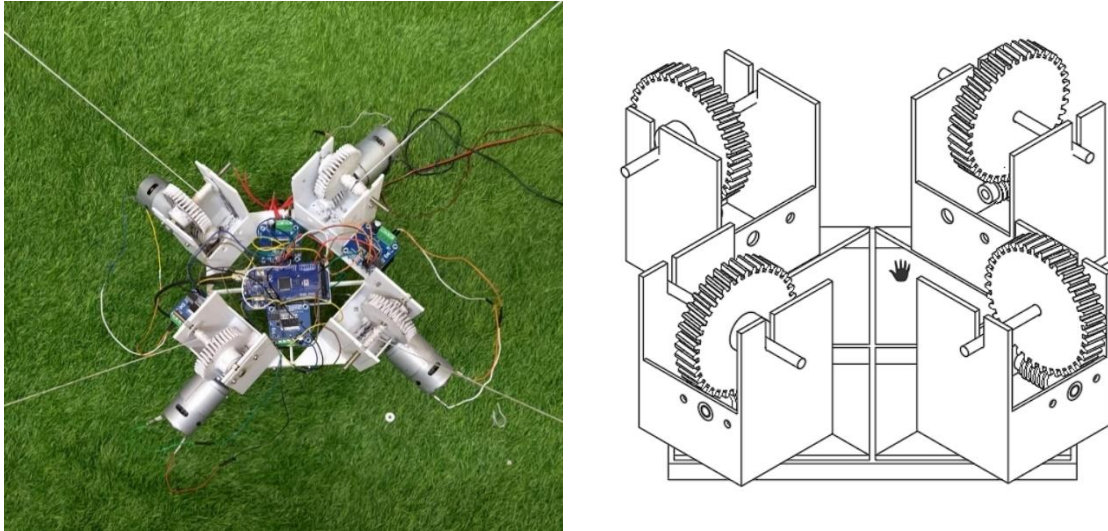


Fig:4.1 : Cable Driven Platform

4.4.1 Control algorithm for CDPR locomotion

The control algorithm assumes some preset values such as:

1. Current position coordinates (start coordinates are set at the starting)
2. Current wire lengths (starting wire lengths are set at the starting)
3. the maximum velocity of the platform
4. Area dimensions and limits in the x, y and z directions
5. The coordinate system itself

6. Cable length released per rotation of the motor
7. Maximum RPM of the motors
8. Distance between the CDPR frame center and the cable release points

The algorithm takes the target coordinates as input and sets the motor speed, motor engage time and motor spin direction as output. Motor speed is set using Pulse Width Modulation (PWM). Pulse width modulation restricts the amount of time output is high in a given time period. Such a pulsating signal will appear to be a overall output voltage lesser than or equal the input voltage. The pulse width is what we will be setting to control the amount of voltage supplied to the motors, thereby controlling the motors' speed. Pulse width is maintained by timers. For our purposes, we will use the internal timers of the ATMEGA 2560 processor in ARDUINO MEGA Board. There will be 4 timers, each for each motor. Timers are 16 bit timers, meaning that we will have 65536 possible values to set the pulse width of the motor. With all this, the algorithm works mentioned below.

Note: The limits of workspace are mentioned as farthest and nearest with respect to the origin.

1. Initiate the timers of ATmega processor.
2. If current coordinates is within its x axis limits, proceed with the operation
3. If the current Y coordinate is at its limits, change the state of forwardFlag
4. Based on the forwardFlag, If current Y coordinate is at its farthest limit, set the target Y coordinate towards the nearest limit in 0.5m steps.
 - Else if the Y coordinate is at its nearest limit, set the target Y coordinate towards the farthest limit in 0.5m steps.
5. Calculate the target wire lengths for the given target coordinates using Pythagoras theorem.
 - For each wire, the target coordinates are modified with the frame parameters (a and b), so that the wire lengths are calculated until the wire release points instead of up to center of the frame.
6. Calculate the distance to be travelled between the current and target coordinates using the distance formula.
7. Calculate the Cable driven platform travel time by dividing the distance to be travelled by the maximum platform velocity.

8. For each wire, Calculate the length of cable to released or retracted by subtracting the target wire lengths from the current wire lengths.
9. For each wire, based on the cable length difference, calculate the no of rotations of motor and real, required to release are retract the cable.
 - Divide the cable length difference by cable length released per rotation
10. For each wire, calculate the required RPM of the motor
 - divide the number of rotations by platform travel time in milliseconds.
11. For each wire, calculate the PWM value.
 - Divide the maximum RPM of the motor with the calculated RPM. Multiply it by 65535. Round this value to floor to get a permissible PWM value.
12. For each wire, calculate the actual motor RPM from the calculated PWM value.
 - Divide the PWM value by 65535. Multiply the result with 60000.
13. For each wire, calculate the time for which the motor need to be engaged (kept on), using the actual motor RPM maintained.
 - Divide the no. of rotations by actual RPM. Multiply the result by 60000 to get milliseconds.
14. Set the PWM wave to motor speed control pin.
15. Based on the sign of cable length difference, set the appropriate motor pin HIGH to ensure the correct direction of rotation.
 - Record the actuation time.
16. A loop checks the difference between actuation time and current time and turns of appropriate motors when the time difference crosses the respective calculated engagement time. Loop ends when all the motors are set OFF.

4.5 ISOLATION MECHANISM

A CDPR platform is more stable in its position than its orientation. To avoid the cable driven platforms' oscillations propagate to the attachments, motorized gimbal is expected for isolation between platform orientation and attachments. But, to simplify the control process, we are using a ball and socket joint which is expected to give 2 DOF. The joint is left to suspend itself by its own weight, perpendicular to the ground, for some time after the platform stops moving. After a predefined time, assuming the oscillations have settled, the ball is locked in its place. The predefined time for our setup was identified to be approximately 2 seconds.

4.6 ATTACHMENT DESIGN

4.6.1 Seeder

Initially we were planning to incorporate the drum seeder mechanism that is fully developed. In drum seeder, the load was supported by the wheels on either side by direct ground contact. But, the presence of the CDPR platform eliminates the need for such ground contact to carry the load. Instead the setup can be entirely suspended from the platform. Our design consists of a hopper, a vertical drum with hole on the bottom face and a disc attached to a motor.

The hopper will contain the seeds. Small, regular amount of seeds are dropped from the hopper into the vertical drum. The drum has a hole in its bottom face, through which the seeds are dropped to the ground. Hole's size controls the seed metering (number of seeds planted at a particular spot). Hole diameter is taken as the same from the drum seeder, i.e., 9mm.

A disc, attached to a motor is installed closely with the drum's bottom face. The seed metering hole is covered by a disc for the most part. A sector of the disc has an opening. This is a provision for opening the drum's hole and letting the seeds drop. The disc is rotated by it's motor. When the open sector of the disc and the seed metering hole coincides seeds are dropped. The speed of CDPR platform and the RPM of the disc are setup to synchronize the release of seeds at constant intervals of 25cm and the disc is rotated to open the seed metering hole for hill planting. The motor is again operated after a certain time interval to close the hole. This time period can be set to ensure the number of seeds sown with a tight deviancy. The CDPR drops the seeds from a height of 20cm. The motor and the disc are connected via pinion and gears with a gear ratio of 9:1. The following equations govern the rpm of the disc:

$$\text{Time of release (min)} = (\text{platform velocity (m/min)} / \text{distance (m)})$$

$$\text{Rpm of the motor} = (1 / \text{time of release}) * \text{gear ratio}$$

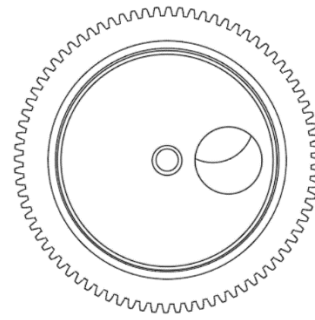
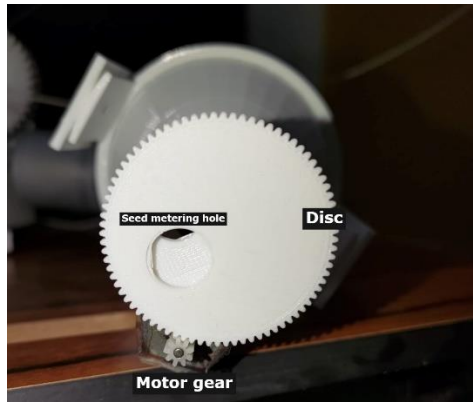


Fig:4.2 : Seeder Mechanism

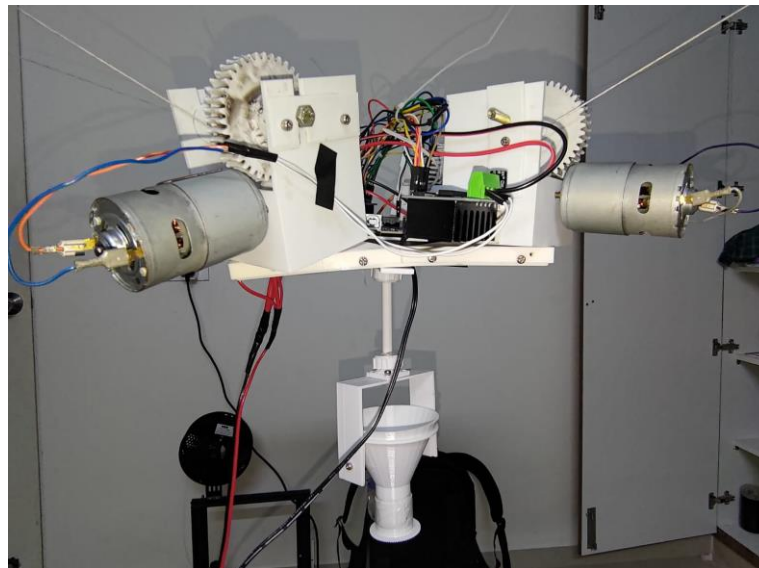


Fig:4.3 : Seeder Mechanism with CDPR

4.6.2 Weeder

The weeding mechanism consists of a cylindrical body with spikes on its outer surface. The prototype cylinder has a diameter of 6.3 cm and width 10.5 cm. The spikes are 3 cm in height. When the wheel rotates, the spikes till the soil, disrupting the weeds. The weeder is operated in-between the rows. Conventional weeders require the attachment to have ground contact to support their load. The friction from the system's normal forces and inclined application of manual force in those weeders contribute significantly to the draft force required. Here, the CDPR platform carries the

load eliminating the load's contribution to friction. This leaves only the soil resistance to be handled by the weeding motor. Lower torque and power can be used.

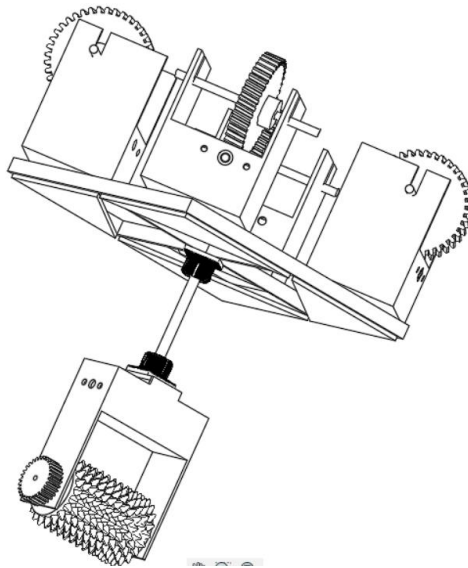


Fig:4.4 : Isometric View of Weeder



Fig:4.5 : Spiked wheels of Weeder

4.6.3 Fertilizer Spreader

A centrifugal fertilizer spreader contains a hopper, a motorized disc with guiding vanes. Fertilizer is dropped from the hopper on the disc. The motorized disc is a rotating platform. This rotation causes centrifugal force on the fertilizer particles. Because of which they are scattered out of the disc onto the field. Unlike its tractor based counterpart, full 360 degrees is scattered with the fertilizer. Uniform scattering and range of scatter depends upon the fertilizer's frictional properties, profile of the vanes and importantly motor speed. Variable amounts can be scattered at localized zones by increasing or decreasing the overall CDPR traversal speed. The prototype consists of a disc of diameter 4cm.

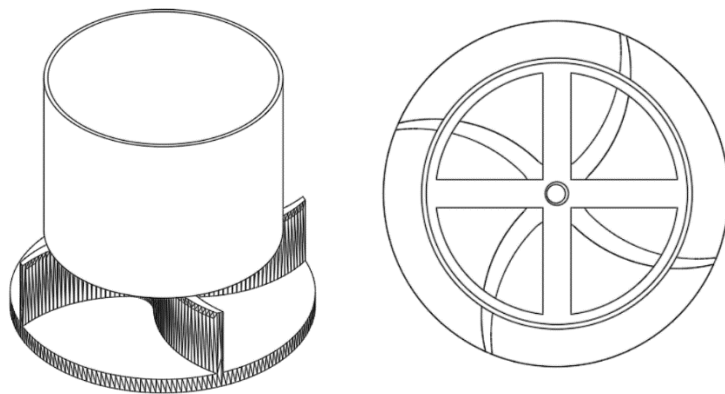


Fig:4.6 : Vanes and Drum of Fertilizer Spreader

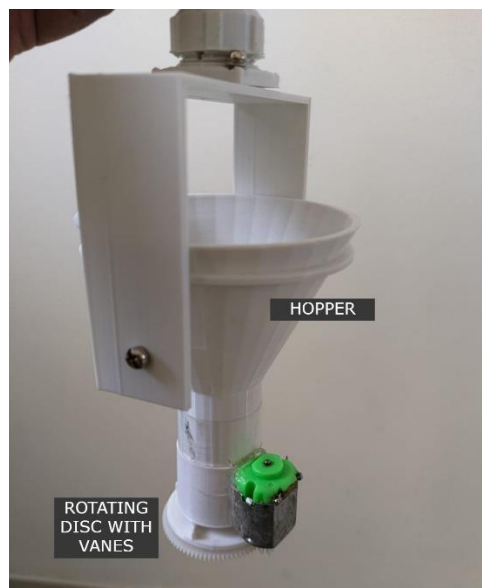


FIG:4.7 : Fertilizer Spreader Mechanism with CDPR

4.6.4 Phenotyping

Phenotyping is the process of sensing various crop and field properties. Fixing a particular phenotyping system is rather constraining for a prototype. Different farm strategies need different plant and soil sensor requirements. Literature review shows that Image based ML models for detecting localised maps of drought, disease and pest attacks are in being actively researched in the context of phenotyping. Image acquisition system is done with UAV such as drones. It is not needed for our system to have its own unique ML model. We can make use of open source projects on the same topic. For demonstrating the incorporation of UAV capabilities, this project uses an image processing and object detection system for detecting paddy diseases and pest attack. This part of the project requires a camera, good processor, memory and GPS. Instead of purchasing these components separately, we will be reusing a mobile phone for demonstration purposes. This cut down the cost for forging this prototype and also opened up possibilities to incorporate dedicated open source projects into this project. We have adapted an open source paddy disease and pest attack detection application - PalayLab, into this project. The application is android native and uses tensorflow for it's ML model creation. The tflite ML model of the application is used for our own purposes. On startup, the app starts the camera and calls the ML model for predictions for that image. The ML model returns a list of prediction values and their confidence. These predictions are written to a JSON format file in the external storage. A location change interceptor lets us to capture the image and store the predictions whenever the GPS location changes. GPS location is used to map these prediction data to a part of the field. The file in external storage acts as the GIS system. Data stored for each location has latitude, longitude and a disease list. The individual disease stats have the disease name and the confidence in its detection. This information can be easily parsed with many of the available JSON parsers namely the Jackson Project.

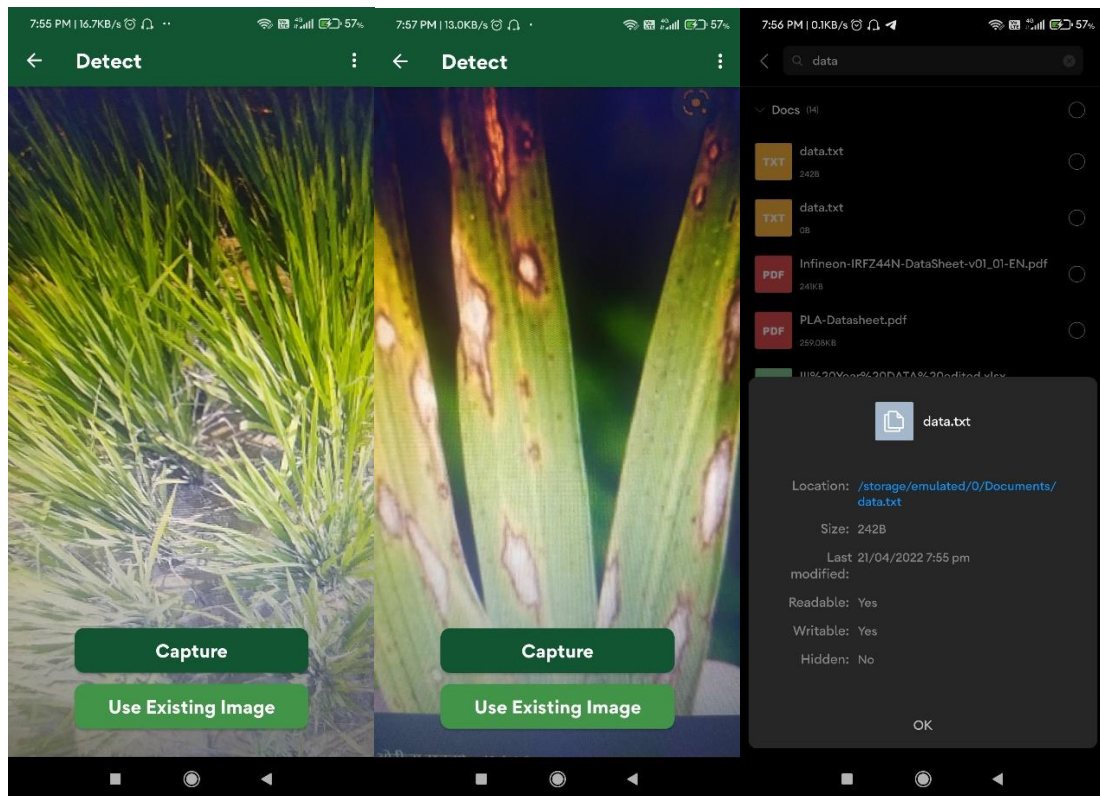


Fig:4.8 : Paddy Disease and Pest Identification

```
data.txt - Notepad
{"diseases":{"Black Bug":0.0,"Far Bug":3.325,"Green Leafhopper":0.0,"Sheath Blight":0.0,"Army Worm":0.0,"Bacterial Leaf Blight":0.0,"Golden Apple Snail":15.15,"Blast":58.066666666666666,"Tungro":16.633333333333333},"latitude":13.1150599,"longitude":80.2511246,"sampleCount":5}
```

Fig:4.9 : Disease Data JSON

CHAPTER 5

RESULTS AND DISCUSSION

5.1 INTRODUCTION

The following results are obtained from simulations and test code mentioned in Appendix B. The results were confirmed by running the cable driven platform in a space of 2.7m x 3m x 1.2m (X Y Z lengths). The platform operated within 0.5m from the edges of the space, at a speed of 5m per min. The trajectory followed by the platform was straight line traversals in Y axis from 0.5m to 2.5m and a 0.125m traversal in X axis to the next row.

5.2 PLATFORM TRAJECTORY

Trajectory of the CDPR platform is a vital component to ensure the expected operation of the attachments. The set trajectory traverses the field row by row from one end to other with a hop to next row when it reaches the field limit. This path can be broken down into straight lines that can be focused on one at a time. Positional accuracy of the platform is controlled by the accuracy of the cable lengths. These cable lengths are calculated to be released or retracted completely at the end of every single line traversal. Ideally, for a given traversal, the number of rotations required from each motor to release or retract the required cable lengths are all divided by one common travel time, from which the RPM of each motor is obtained. The motors are engaged for this common time period with calculated RPM. This ensures that the cable lengths are done releasing or retracting at the same end time. But, due to limitations in controller hardware, the PWM duty cycle values that are available are only discrete. This is because the Timers from processor is a digital timer. So we need to round off the calculated RPM to closest discrete RPM that is possible. Due to this, the time for which each motor will be engaged changes, compromising the synchronization of the cable length at the end of the trajectory. This introduces deviation of platform position at the end point. The deviations last for a time interval equal to the difference between

maximum and minimum motor engagement time. So, a key factor here is to reduce the difference between motor engagement times.

The arduino mega 2560 board provided 8 bit PWM with its analogWrite function. 8 bits offered 256 possible values, dividing 12V into 256 discrete voltage levels. This had increased difference between ideal RPM and the actual rounded off RPM. The maximum time difference between motor engagement time showed to be an average of 14 seconds to traverse 5m in a 6m×6mspace. This created significant trajectory deviations at the end point. So, we directly accessed the 16 bit Timers in ATMEGA 2560 Processor in Arduino Mega. The 16 bits unlocked 65536 possible voltage levels. The gap between ideal and actual RPM values narrowed, which brought down the maximum time difference between motor engagement times to lesser than 10 milliseconds. The trajectory deviations at the end of the trajectory was greatly reduced by this.

For a line single trajectory, the platform is also expected to pass through a number of collinear intermediate positions between start and end points. These points are traversed in a constant speed. But as per the observations and simulation, to achieve a linear speed of platform, the motor RPMs (or cable speeds) were needed to be accelerated (or decelerated) constantly. Maintaining the motors at constant RPM produced a curved trajectory. Maximum deviation occurred at the middle the trajectory. It was also observed that the maximum deviation is proportional to the travel distance. The following table provides a subset of the results from tests that were ran. Maximum error of cable length in each division is noted to represent the overall increase or decrease in error of all four cables. The maximum error can be seen to increase in middle of each trajectory. And the errors can also seen to increase as the travel distance increases. These tests simulations are run in a space of 64m x 64m (approximately operates in a acre area). The trajectory is set along the Y axis, whereas the X and Z dimensions are kept at 32m and 3m. The platform moves with a velocity of 5m per minute.

Table 5.1 : Error Variation with respect to travel distance

Division	Set 1 Distance = 0.5 m	Set 2 Distance = 5 m	Set 3 Distance = 60 m
0 (current position)	0	0	0
1	0.00062	0.0617	4.94665
2	0.00093	0.0923	6.29353
3	0.00093	0.0920	5.58145
4	0.00062	0.0611	4.42892
5 (target position)	0	0	0

Maintaining proper acceleration requires parallel threads to run for each motor for efficient control. Instead, we have implemented the algorithm to divide the travel distance into steps of 50cm or lesser at a given time to keep the trajectory deviations and errors minimal. This brought down the deviations per step to lesser than 2cm.

It is to be noted that despite the deviations, the error in the end position was minimal. This error was also observed from simulations to be minimized to the order of 10^{-5} m, when the travel distances are kept short. The trajectory control alone ensured most expected operation from each mechanism. The error in cable length decreases with the increase in depth at which the cable driven platform is suspended. The following table lists the maximum error in cable length for in each traversal. The travel distance is set to be 0.5m, from 0.5m to 1m in Y axis. The X coordinate is maintained at 0.5m. Only Z coordinate is changed for each iteration and the maximum error in cable length is noted.

Table 5.2 : Error variation with respect to Z axis

Depth of operation (Z axis in m)	Maximum Error in cable length (in m)
1	0.01723
5	0.00582
10	0.00297

For the same operation, the errors were also observed to vary within the different parts of the field. A traversal of 0.5m to 2.5m in Y axis at a depth of 0.6m produces different errors for different X coordinates. The test is set to run in a space of dimensions, 3m x 3m. The error is high around the edge of the field and lowest at the middle of the field. The following table lists various X coordinates and the maximum error produced in cable lengths.

Table 5.3 : Error variation with respect to X axis

X coordinate in m	Maximum Error in cable length (in m)
0.5	0.1306
1	0.1382
1.5	0.1392
2	0.1382
2.5	0.1306

5.3 ATTACHMENT EFFICIENCY

The attachments have exhibited expected behavior. But, the expected accuracy was not attained due to insufficient isolation mechanism. But the mechanisms proved to be working and are capable of good efficiency upon further improvement.

5.4 FURTHER DEVELOPMENT

For a real application, the system has to be scaled up. Following that, the material has to be changed to a much sturdy material. The control circuitry is not bound to undergo much changes in terms of general architecture. Motors with greater torque has to be used. Thus, the motor drives has to be updated to capable Power MOSFET. A feedback mechanism can be incorporated using rotary encoders. This will be helpful in the initial setup and reducing errors. An important component that requires development is the isolation mechanism. An ideal solution would be a motorized gimbal mechanism. The seeding and fertilizer attachments require bigger hoppers and discs. Weeder mechanism requires bigger wheels. A more capable system would have two wheels, one behind the other, spinning in the same direction. This will disrupt the sand in between them in both the directions, eliminating the need for to and fro motion followed in a manual weeder operation. Phenotyping attachment is still left ambiguous as different situations require different sensors. Image acquisition system, lightweight ML model and considerably powerful process embedded within the mechanism itself instead of Internet of Things will be a much more compact approach.

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 SUMMARY

In this project, we designed and evaluated a farming automation system based on Cable Driven Parallel Robot. The system was divided into sections namely, Cable driven platform, control circuitry, isolation mechanism, Seeder attachment, Weeder attachment, Fertilizer spreader attachment and phenotyping system. Control algorithm was developed and tested for errors between expected values and the values the algorithm produce. Each section was designed, 3D printed, assembled and tested to conform the findings of the test. All the findings were reported and scope for further development was proposed.

6.2 CONCLUSION

A functional Farming automation was developed to automate seeding, weeding, fertilizer spreading and phenotyping. The designs exhibited expected behavior. Test results show that the control of Cable driven platform shows good performance even without the feedback mechanism. But the isolation mechanism didn't completely suppress the attachment oscillations. It can be concluded that the prototype still requires further development in the areas of stability against oscillation and feedback. Upon further development, the system shows promise to be a capable and low cost alternative to conventional automations.

REFERENCES

1. Amir Moradi, (2013), Stiffness Analysis of Cable-Driven Parallel Robots, PhD Thesis submitted to Queen's University.
2. Artur Przywara, (2015), The impact of structural and operational parameters of the centrifugal disc spreader on the spatial distribution of fertilizer, Science Direct
3. Chetan Dwarkani M, Ganesh Ram R, Jagannathan S, R. Priyatharshini, (2015), Smart Farming System Using Sensors for Agricultural Task Automation, IEEE.
4. Dileep M R, Navaneeth A V, Savita Ullagaddi, Ajit Danti, (2020), A Study and Analysis on Various Types of Agricultural Drones and its Applications, IEEE.
5. H. Eller, A. Denoth, (1995), A Capacitive soil moisture sensor, Journal of Hydrology, Pages 137-146
6. Francesco Marinello, Andrea Pezzuolo, Alessandro Chiumenti, Luigi Sartori, (2016), Technical analysis of unmanned aerial vehicles (drones) for agricultural applications, Latvia University of Agriculture.
7. James Lowenberg-DeBoer, Iona Yuelu Huang, Vasileios Grigoriadis, Simon Blackmore, (2019), Economics of robots and automation in field crop production, Springer.
8. Jianhan Lin, Maohua Wang, Miao Zhang, Yane Zhang, Li Chen, (2007), Electrochemical Sensors for Soil Nutrient Detection: Opportunity and Challenge, Computer and Computing Technologies in Agriculture, Volume II, Pages 1349 - 1353, Springer
9. H. De Laulanié, (2011), Intensive Rice Farming in Madagascar, Tropicultura
10. Laxmi C. Gavade, A.D. Bhoi, (2017), N, P, K Detection & Control for Agriculture Applications using PIC Controller: A Review, International Journal of Engineering Research & Technology (IJERT)
11. S. Meivel, S. Maheswari, (2020), Remote Sensing Analysis of Agricultural Drone, Springer
12. Norbert Kirchgessner, Frank Liebisch, Kang Yu, Johannes Pfeifer, Michael Friedli, Andreas Hund, Achim Walter, (2016), The ETH field phenotyping platform FIP: a cable-suspended multi-sensor system, CSIRO.
13. Pablo Gonzalez-de-Santos, Angela Ribeiro, Cesar Fernandez-Quintanilla, Francisca Lopez-Granados, Michael Brandstötter, Slobodanka Tomic, Stefania Pedrazzi, Andrea Peruzzi, Gonzalo Pajares, George Kaplanis, Manuel Perez-Ruiz,

- Constantino Valero, Jaime del Cerro, Marco Vieri, Gilles Rabatel, Benoit Debilde, (2016), *Fleets of robots for environmentally-safe pest control in agriculture*, Springer.
14. Rahul Tripathi*, Mohammad Shahid, A.K.Nayak, R.Raja, B.B.Panda, Sangita Mohanty, K.Thilgham and Anjani Kumar, (2013), "Precision Agriculture in India: Opportunities and Challenges", Central Rice Research Institute.
 15. Redmond Ramin Shamshiri, Cornelia Weltzien, Ibrahim A. Hameed, Ian J. Yule, Tony E. Grift, Siva K. Balasundram, Lenka Pitonakova, Desa Ahmad, Girish Chowdhary, (2018), *Research and development in agricultural robotics: A perspective of digital farming*, International Journal of Agricultural and Biological Engineering (IJABE)
 16. Rendong Nan, Di Li, Chengjin jin, Quiming Wang, Lichun Zhu, Wenbai zhu, Haiyan Zhang, Youling Yue, Lei Quian, (2011), *The Five-Hundred-Meter Aperture Spherical Radio Telescope*, International Journal of Modern Physics, Vol.20, page 989-1024.
 17. Selvakumar Manickam, (2021), *A Drone-based IoT Approach to Agriculture Automation and Increase Farm Yield*, SSRN
 18. Er. H.B.Shakya, Er.M.R.Parmar, Er. M.T.Kumpavat and Dr. R.Swarnkar, (2016), *Development and Performance Evaluation of Manually Operated Cono-Weeder for Paddy Crop*, International Refereed Journal of Engineering and Science (IRJES)
 19. Tuong Phuoc Tho, Nguyen Truong Thinh, (2021), *Analysis and Evaluation of CDPR Cable Sagging Based on ANFIS*, Hindawi.
 20. Flourish Project, funded by the European Community's Horizon 2020 programme - <http://flourish-project.eu/documents>
 21. Proceedings of the first International Workshop on Robotics and Associated High Technologies and Equipment for Agriculture, RHEA-2011
 22. RHEA project, funded by European Commission - <http://www.rhea-project.eu/Summary.php>
 23. Rice Knowledge Bank - International Rice Research Organization - <http://www.knowledgebank.irri.org/>
 24. Soil and Land classification statistics from Tamil Nadu Agriculture Department - <https://www.tn.gov.in/deptst/agriculture.pdf>
 25. SpiderCam - <https://www.spidercam.tv/>
 26. SWEEPER pepper harvesting robot - <http://www.sweeper-robot.eu/>

27. Tamil Nadu Agricultural University AgriTech Portal - https://agritech.tnau.ac.in/agriculture/agri_cropproduction_cereals_rice.html
28. TNAU Paddy drum seeder specification document - <https://tnau.ac.in/abd/wp-content/uploads/sites/50/2022/03/Model-Technology-Document.pdf>

APPENDIX A

The following is the C code to control the CDPR platform's movement. The algorithm this code follows is mentioned in the section 4.4.1.

```
//Code sketch by Vigneshwar M

//a - right rotation

//b - left rotation

#define mot1a 22

#define mot1b 23

#define mot2a 24

#define mot2b 25

#define mot3a 26

#define mot3b 27

#define mot4a 28

#define mot4b 29

#define atchEn 10

#define atchA 30

#define atchB 31


const uint16_t MAX_PWM_VALUE = 65535U;


/*-----*/
```

```

/*box state variable*/

float boxdim[3] = {2.70,3,1.2};


/*pod design variables*/

float a = 0.085;

float b = 0.07;


/*spiderbot state variables in 'm'*/

float curCoord[3] ={0.5,0.5,0.6};

float curWireLen[4] = {3.0528,6.1903,8.2049,6.1903};

float mprot = 1/6; /*meters per rotation, cable released from a 1cm dia reel*/

bool forwardFlag = true, opsFlag = false;


/*standard maximum pod velocity in 'm/s'*/

/*estimated reel dia = 0.01m;

circumference/rope released per rotation = 1/6 m

maxvel = 5m/min*/

float maxVel = ((float)5)/60;


/*target state variables in 'm'*/

float targCoord[3] = {0.5,0.5,0.6};

float targWireLen[4];

```

```

float travelDist = 0;

unsigned long travelTime = 0;


/*control variables*/

float maxMotorSpeed = 6400;

int motSpeed[4];

unsigned long engTime[4]; /*motor engage time*/


/*-----*/

/**
 * func - hyp
 * input - three float values as three coordinates
 * return - on float value as distance from local ref point (0,0,0)
 * uses pythagorian theorem
 */

float hyp(float a, float b, float c){

    float result = sqrt( pow(a,2) + pow(b,2) + pow(c,2));

    return result;

}

void initPWM() {

    noInterrupts();

```

```
TCCR1A = 1 << WGM11 | 1 << COM1A1 | 1 << COM1B1; // set on top, clear OC on  
compare match
```

```
TCCR1B = 1 << CS10 | 1 << WGM12 | 1 << WGM13; // clk/1, mode 14 fast PWM
```

```
TCCR3A = 1 << WGM31 | 1 << COM3A1 | 1 << COM3B1 | 1 << COM3C1; // set on top,  
clear OC on compare match
```

```
TCCR3B = 1 << CS30 | 1 << WGM32 | 1 << WGM33; // clk/1, mode 14 fast PWM
```

```
TCCR4A = 1 << WGM41 | 1 << COM4A1 | 1 << COM4B1 | 1 << COM4C1; // set on top,  
clear OC on compare match
```

```
TCCR4B = 1 << CS40 | 1 << WGM42 | 1 << WGM43; // clk/1, mode 14 fast PWM
```

```
ICR1 = MAX_PWM_VALUE;
```

```
ICR3 = MAX_PWM_VALUE;
```

```
ICR4 = MAX_PWM_VALUE;
```

```
interrupts();
```

```
pinMode(12,OUTPUT);
```

```
pinMode(11,OUTPUT);
```

```
pinMode(5,OUTPUT);
```

```
pinMode(3,OUTPUT);
```

```
pinMode(2,OUTPUT);
```

```
pinMode(8,OUTPUT);
```

```
pinMode(7,OUTPUT);
```

```
pinMode(6,OUTPUT);
```

```

}

/* Func - setTarg

* Input - target coordinates

* Sets target wire lengths

* Sets travel Distance between curr and targer points

* Sets travel time

*/

void setCur(float tCoord[3]){

    curWireLen[0] = hyp((tCoord[0]-a),(tCoord[1]-b),tCoord[2]);

    curWireLen[1] = hyp((boxdim[0]-tCoord[0]-a),(tCoord[1]-b),tCoord[2]);

    curWireLen[2]      =      hyp((boxdim[0]-tCoord[0]-a),(boxdim[1]-tCoord[1]-
b),tCoord[2]);

    curWireLen[3] = hyp((tCoord[0]-a),(boxdim[1]-tCoord[1]-b),tCoord[2]);

}

void setTarg(float targCoord[3]){

    targWireLen[0] = hyp((targCoord[0]-a),(targCoord[1]-b),targCoord[2]);

    targWireLen[1]      =      hyp((boxdim[0]-targCoord[0]-a),(targCoord[1]-
b),targCoord[2]);

    targWireLen[2]      =      hyp((boxdim[0]-targCoord[0]-a),(boxdim[1]-targCoord[1]-
b),targCoord[2]);

    targWireLen[3]      =      hyp((targCoord[0]-a),(boxdim[1]-targCoord[1]-
b),targCoord[2]);

```

```

        travelDist      =      hyp(      (targCoord[0]-curCoord[0]),(targCoord[1]-
curCoord[1]),(targCoord[2]-curCoord[2]) );

        travelTime = (travelDist / maxVel)*1000;

}

/* Func - setMotor

* Input = motor index

* sets motor speed based on travel time

* sets motor engagement time based on motor speed

*/

int setMotor(int motInd){

    float noofrotations = (targWireLen[motInd] - curWireLen[motInd]) / mprot;

    float rpm = (float)(noofrotations/((double)travelTime/60000));

    motSpeed[motInd] = (int)((rpm/maxMotorSpeed)*65535);

    if(motSpeed[motInd] == 0){

        motSpeed[motInd] = 1;

    }

    rpm = ((float)motSpeed[motInd]/65535)*maxMotorSpeed;

    engTime[motInd] = (unsigned long) ( (noofrotations/rpm)*60000 );

}

```

```

/* Apply the mot speed to the speed control pins of motor drive

* ensure all motor pins are low

* make an int array of drive pins that need to be high based on the sign of mot
speed

*/

void engDriver(){

    OCR1B = (uint16_t)abs(motSpeed[0]);

    OCR1A = (uint16_t)abs(motSpeed[0]);

    OCR4C = (uint16_t)abs(motSpeed[1]);

    OCR4B = (uint16_t)abs(motSpeed[1]);

    OCR4A = (uint16_t)abs(motSpeed[2]);

    OCR3A = (uint16_t)abs(motSpeed[2]);

    OCR3C = (uint16_t)abs(motSpeed[3]);

    OCR3B = (uint16_t)abs(motSpeed[3]);


    byte motHighPins[4] = {mot1a,mot2a,mot3a,mot4a};

    if(motSpeed[0]<0){

        motHighPins[0] = mot1b;

    }

    if(motSpeed[1]<0){

        motHighPins[1] = mot2b;

    }

```

```

    if(motSpeed[2]<0){

        motHighPins[2] = mot3b;

    }

    if(motSpeed[3]<0){

        motHighPins[3] = mot4b;

    }

    if(opsFlag){

        startOperation();

    }

    engLoop(motHighPins);

}

/* Func - engLoop

* operates the motors for calculated engTime

* keeps track of time using millis() func

*

*/

void engLoop(byte motHighPins[]){

    bool mA = true, mB = true, mC = true, mD = true;

    digitalWrite(motHighPins[0],HIGH);

    digitalWrite(motHighPins[1],HIGH);

    digitalWrite(motHighPins[2],HIGH);

    digitalWrite(motHighPins[3],HIGH);

```

```

unsigned long startTime = millis();

while(true){

    if(mA && ( (millis() - startTime) >= engTime[0])){

        digitalWrite(motHighPins[0],LOW);

        mA = false;

    }

    if(mB && ((millis() - startTime)>= engTime[1])){

        digitalWrite(motHighPins[1],LOW);

        mB = false;

    }

    if(mC && ( (millis() - startTime) >= engTime[2])){

        digitalWrite(motHighPins[2],LOW);

        mC = false;

    }

    if(mD && ( (millis() - startTime) >= engTime[3])){

        digitalWrite(motHighPins[3],LOW);

        mD = false;

    }

    if((!mA) && (!mB) && (!mC) && (!mD)){

        break;

    }

}

}

```

```

void setup() {

    Serial.begin(9600);

    pinMode(mot1a,OUTPUT);

    pinMode(mot1b,OUTPUT);

    pinMode(mot2a,OUTPUT);

    pinMode(mot2b,OUTPUT);

    pinMode(mot3a,OUTPUT);

    pinMode(mot3b,OUTPUT);

    pinMode(mot4a,OUTPUT);

    pinMode(mot4b,OUTPUT);

    pinMode(atchEn,OUTPUT);

    pinMode(atchA,OUTPUT);

    pinMode(atchB,OUTPUT);

    initPWM();

    setCur(curCoord);

}

```

```

void loop(){

    if(curCoord[0]<=boxdim[0]){

        if(curCoord[1] >= boxdim[1]-0.5){

```

```

forwardFlag = false;

opsFlag = false;

stopOperation();

targCoord[0] = curCoord[0] + 0.125;

motion();

curCoord[0] = targCoord[0];

curCoord[1] = targCoord[1];

curCoord[2] = targCoord[2];

curWireLen[0] = targWireLen[0];

curWireLen[1] = targWireLen[1];

curWireLen[2] = targWireLen[2];

curWireLen[3] = targWireLen[3];

}else if(curCoord[1] <= 0.5){

    forwardFlag = true;

    opsFlag = false;

    stopOperation();

    targCoord[0] = curCoord[0] + 0.125;

    motion();

    curCoord[0] = targCoord[0];

    curCoord[1] = targCoord[1];

    curCoord[2] = targCoord[2];

    curWireLen[0] = targWireLen[0];

    curWireLen[1] = targWireLen[1];

```

```

        curWireLen[2] = targWireLen[2];

        curWireLen[3] = targWireLen[3];
    }

    if(forwardFlag){

        targCoord[1] = curCoord[1] + 0.5;

    }else{

        targCoord[1] = curCoord[1] - 0.5;

    }

    opsFlag = true;

    motion();

    stopOperation();

    curCoord[0] = targCoord[0];

    curCoord[1] = targCoord[1];

    curCoord[2] = targCoord[2];

    curWireLen[0] = targWireLen[0];

    curWireLen[1] = targWireLen[1];

    curWireLen[2] = targWireLen[2];

    curWireLen[3] = targWireLen[3];

}

}

```

```
void motion() {  
  
    setTarg(targCoord);  
  
    setMotor(0);  
  
    setMotor(1);  
  
    setMotor(2);  
  
    setMotor(3);  
  
    engDriver();  
  
}
```

```
void startOperation(){  
  
    analogWrite(atchEn,150);  
  
    digitalWrite(atchA,HIGH);  
  
    digitalWrite(atchB,LOW);  
  
}
```

```
void stopOperation(){  
  
    analogWrite(atchEn,0);  
  
    digitalWrite(atchA,LOW);  
  
    digitalWrite(atchB,LOW);  
  
}
```

APPENDIX B

The following is the C++ code logs the outputs of the controller code to the console and compare them with expected true values. The same program is run for a different sets of field dimensions, travel distances and starting and end points within a field. Parts of the controller program is first run to set up the target Wire lengths, motor engagement time. Controller program operates the motors in constant RPM. Hence, the wire lengths at fixed intervals of time is calculated up to the minimum motor engagement time. As the motor RPM is constant, this is equivalent to dividing the cable lengths by a common divisor. The code also divides the travel distance into positions, at the same time intervals time in which wire lengths are calculated. Thus, the travel distance is divided into equidistant collinear points, since the velocity is constant. The cable lengths required to reach each of these positions are calculated. These expected wire lengths are then compared to the actual wire lengths at the same time. The results are printed to the terminal using simple “cout” statements.

Note: Here the abbreviation wl means wire length. The array wl holds the expected wire lengths for the current position. The arrays wIA, wIB, wIC, wID holds the all the actual wire lengths of each motor respectively.

```
#include <iostream>

#include <cmath>

#include <algorithm>

using namespace std;

/*box state variable*/

float boxdim[3] = {6,6,4};

/*pod design variables*/

float a = 0.1;
```

```

float b = 0.1;

/*spiderbot state variables in 'm'*/

float curCoord[3] ={0.5,0.5,3};

float curWireLen[4] = {3.0528,6.1903,8.2049,6.1903};

float mprot = 0.00167; /*meters per rotation, cable released from a 1cm dia reel*/

/*standard maximum pod velocity in 'm/s'*/

/*estimated reel dia = 0.01m;

circumference/rope released per rotation = 0.0314 m

maxvel = 5m/s*/

float maxVel = ((float)5)/60;

/*target state variables in 'm'*/

float targCoord[3] = {0.5,1,3};

float targWireLen[4];

float travelDist = 0;

unsigned long travelTime = 0;

/*control variables*/

float maxMotorSpeed = 6400;

int motSpeed[4];

unsigned long engTime[4]; /*motor engage time*/

```

```

/*-----*/

/**

* func - hyp

* input - three float values as three coordinates

* return - on float value as distance from local ref point (0,0,0)

* uses pythagorian theorem

*/

float hyp(float a, float b, float c){

    float result = sqrt( pow(a,2) + pow(b,2) + pow(c,2));

    return result;

}

void setTarg(float targCoord[3]){

    targWireLen[0] = hyp((targCoord[0]-a),(targCoord[1]-b),targCoord[2]);

    targWireLen[1]      =      hyp((boxdim[0]-targCoord[0]-a),(targCoord[1]-
b),targCoord[2]);

    targWireLen[2]      =      hyp((boxdim[0]-targCoord[0]-a),(boxdim[1]-targCoord[1]-
b),targCoord[2]);

    targWireLen[3]      =      hyp((targCoord[0]-a),(boxdim[1]-targCoord[1]-
b),targCoord[2]);

    travelDist      =      hyp(      (targCoord[0]-curCoord[0]),(targCoord[1]-
curCoord[1]),(targCoord[2]-curCoord[2]) );

```

```

        travelTime = (travelDist / maxVel)*1000;

    }

void WLCalc(float pose[3], float wl[4]){

    wl[0] = hyp((pose[0]-a),(pose[1]-b),pose[2]);

    wl[1] = hyp((boxdim[0]-pose[0]-a),(pose[1]-b),pose[2]);

    wl[2] = hyp((boxdim[0]-pose[0]-a),(boxdim[1]-pose[1]-b),pose[2]);

    wl[3] = hyp((pose[0]-a),(boxdim[1]-pose[1]-b),pose[2]);

}

void setMotor(){

    for(int motInd=0;motInd<4;motInd++){

        float noofrotations = (targWireLen[motInd] - curWireLen[motInd]) / mprot;

        float rpm = (float)(noofrotations/((double)travelTime/60000));

        motSpeed[motInd] = (int)((rpm/maxMotorSpeed)*65535);

        if(motSpeed[motInd] == 0){

            motSpeed[motInd] = 1;

        }

        rpm = ((float)motSpeed[motInd]/65535)*maxMotorSpeed;

        engTime[motInd] = (unsigned long) ( (noofrotations/rpm)*60000 );

    }

}

```

```

int main()

{

    setTarg(targCoord);

    setMotor();

    cout<<"\ntargCoord  curCoord\n";

    for(int i=0;i<3;i++){

        cout<<targCoord[i]<<" "<<curCoord[i]<<"\n";

    }

    cout<<"\ntargWireLen  curwireLen\n";

    for(int i=0;i<4;i++){

        cout<<targWireLen[i]<<" "<<curWireLen[i]<<"\n";

    }

    cout<<"\nmotspeed  engTime\n";

    for(int i=0;i<4;i++){

        cout<<motSpeed[i]<<" "<<engTime[i]<<"\n";

    }

    cout<<"\ntravelDist  travelTime\n"<<travelDist<<" "<<travelTime<<"\n";

    int div = 10;

    auto positions = new float[div+1][3];

    float *wlA = new float[div+1];

    wlA[0] = curWireLen[0];

```

```

float *w1B = new float[div+1];

w1B[0] = curWireLen[1];

float *w1C = new float[div+1];

w1C[0] = curWireLen[2];

float *w1D = new float[div+1];

w1D[0] = curWireLen[3];

float travelUnitx = 0, travelUnitz = 0;

float travelUnity = travelDist/(float)div;

positions[0][0] = 0.5;

positions[0][1] = 0.5;

positions[0][2] = 3;

float error = 0;

cout<<"\npositions x y z\n";

for(int i =1;i<=div;i++){

    positions[i][0] = positions[i-1][0] + travelUnitx;

    positions[i][1] = positions[i-1][1] + travelUnity;

    positions[i][2] = positions[i-1][2] + travelUnitz;

    cout<<positions[i][0]<<" "<<positions[i][1]<<" "<<positions[i][2]<<"\n";

}

float wireunitA = (targWireLen[0] - curWireLen[0]) ;

float wireunitB = (targWireLen[1] - curWireLen[1]) ;

float wireunitC = (targWireLen[2] - curWireLen[2]) ;

```

```

float wireunitD = (targWireLen[3] - curWireLen[3]) ;

cout<<"\nwire unit\n";

cout<<wireunitA<<" "<<wireunitB<<" "<<wireunitC<<" "<<wireunitD<<"\n";

unsigned                long                minWTime                =
min(min(engTime[0],engTime[1]),min(engTime[2],engTime[3]));

cout<<"\nmintime "<<minWTime<<"\n";

cout<<"\nw1A\n"<<w1A[0]<<"\n";

for(int i=1;i<=div;i++){

    w1A[i] = w1A[i-1] + ((wireunitA/engTime[0]) * (minWTime/div));

    cout<<w1A[i]<<"      "<<w1A[i-1]<<"      "<<((wireunitA/engTime[0])      *
(minWTime/div))<<"\n";

}

cout<<"\nw1B\n"<<w1B[0]<<"\n";

for(int i=1;i<=div;i++){

    w1B[i] = w1B[i-1] + ((wireunitB/engTime[1]) * (minWTime/div));

    cout<<w1B[i]<<"      "<<w1B[i-1]<<"      "<<((wireunitB/engTime[1])      *
(minWTime/div))<<"\n";

}

cout<<"\nw1C\n"<<w1C[0]<<"\n";

for(int i=1;i<=div;i++){

    w1C[i] = w1C[i-1] + ((wireunitC/engTime[2]) * (minWTime/div));

    cout<<w1C[i]<<"      "<<w1C[i-1]<<"      "<<((wireunitC/engTime[2])      *
(minWTime/div))<<"\n";

}

```

```

cout<<"\nwLD\n"<<wLD[0]<<"\n";

for(int i=1;i<=div;i++){

    wLD[i] = wLD[i-1] + ((wireunitD/engTime[3]) * (minWTime/div));

    cout<<wLD[i]<<"      "<<wLD[i-1]<<"      "<<((wireunitD/engTime[3])      *
(minWTime/div))<<"\n";

}

float maxerr = 0;

cout<<"\nwLcal - wL\n";

for(int i=0;i<=div;i++){

    float wL[4];

    WLCalc(positions[i],wL);

    cout<<wLA[i]<<"-"<<wL[0]<<"      "<<wLB[i]<<"-"<<wL[1]<<"      "<<wLC[i]<<"-"
"<<wL[2]<<" "<<wLD[i]<<"-"<<wL[3]<<" "<<"\n";

    float error = max(max((wLA[i]-wL[0]) , (wLB[i]-wL[1])) , max((wLC[i]-wL[2])
, (wLD[i]-wL[3])));

    cout<<"errors a b c d "<<(wLA[i]-wL[0])<<" "<<(wLB[i]-wL[1])<<" "<<(wLC[i]-
wL[2])<<" "<<(wLD[i]-wL[3])<<"\n";

    if(maxerr<error){

        maxerr = error;

    }

    cout<<i<<". error "<<error<<"\n\n";

}

```

```
cout<<"\nmaxerr " <<maxerr<<"\n";

return 0;

}
```

Batch 38 20.04_Plag

ORIGINALITY REPORT

3%

SIMILARITY INDEX

2%

INTERNET SOURCES

1%

PUBLICATIONS

1%

STUDENT PAPERS

PRIMARY SOURCES

1

"Cable-Driven Parallel Robots", Springer Science and Business Media LLC, 2018

Publication

<1%

2

Climate Resilient Agriculture for Ensuring Food Security, 2015.

Publication

<1%

3

www.ijitee.org

Internet Source

<1%

4

Submitted to City University

Student Paper

<1%

5

Alberto Pretto, Stephanie Aravecchia, Wolfram Burgard, Nived Chebrolu et al. "Building an Aerial-Ground Robotics System for Precision Farming: An Adaptable Solution", IEEE Robotics & Automation Magazine, 2020

Publication

<1%

6

Parminder Singh, Avinash Kaur, Anand Nayyar. "Role of Internet of Things and image processing for the development of agriculture robots", Elsevier BV, 2020

Publication

<1%