

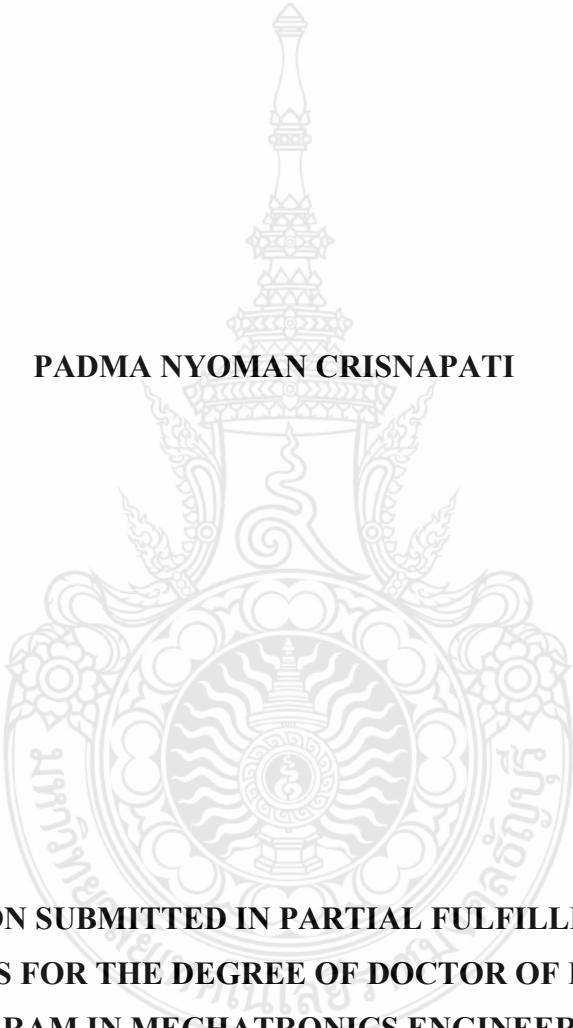
**AUTOMATION IN A TWO-WHEELED DRIVE HAND TRACTOR USING THE
EMBEDDED CONTROL SYSTEM**

PADMA NYOMAN CRISNAPATI

**A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF DOCTOR OF ENGINEERING
PROGRAM IN MECHATRONICS ENGINEERING
(INTERNATIONAL PROGRAM)
FACULTY OF TECHNICAL EDUCATION
RAJAMANGALA UNIVERSITY OF TECHNOLOGY THANYABURI
ACADEMIC YEAR 2022
COPYRIGHT OF RAJAMANGALA UNIVERSITY
OF TECHNOLOGY THANYABURI**

**AUTOMATION IN A TWO-WHEELED DRIVE HAND TRACTOR USING THE
EMBEDDED CONTROL SYSTEM**

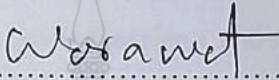
PADMA NYOMAN CRISNAPATI

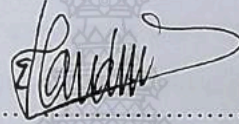


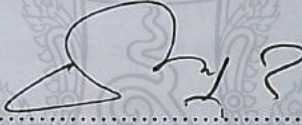
**A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF DOCTOR OF ENGINEERING
PROGRAM IN MECHATRONICS ENGINEERING
FACULTY OF TECHNICAL EDUCATION
RAJAMANGALA UNIVERSITY OF TECHNOLOGY THANYABURI
ACADEMIC YEAR 2022
COPYRIGHT OF RAJAMANGALA UNIVERSITY
OF TECHNOLOGY THANYABURI**

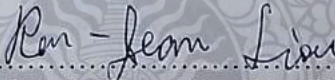
Dissertation Title Automation in a Two-Wheeled Drive Hand Tractor Using the Embedded Control System
Name-Surname Mr. Padma Nyoman Crisnapati
Program Mechatronics Engineering
Dissertation Advisor Associate Professor Dechrit Maneetham, D.Eng., Ph.D.
Academic Year 2022

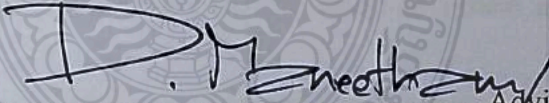
DISSERTATION COMMITTEE


..... Chairman
(Professor Worawat Sa-ngiamvibool, Ph.D.)



..... Committee
(Assistant Professor Evi Triandini, Ph.D.)


..... Committee
(Assistant Professor Petrus Sutiyasadi, D.Eng.)


..... Committee
(Associate Professor Ren Jean Liou, Ph.D.)


..... Advisor
(Associate Professor Dechrit Maneetham, D.Eng., Ph.D.)

Approved by the Faculty of Technical Education, Rajamangala University of Technology Thanyaburi in Partial Fulfillment of the Requirements for the Degree of Doctor of Engineering


..... Dean of Faculty of Technical Education
(Assistant Professor Arnon Niyomphol, M.S.Tech.Ed.)

Date 19 Month April Year 2023

Dissertation Title	Automation in a Two-Wheeled Drive Hand Tractor Using the Embedded Control System
Name-Surname	Mr. Padma Nyoman Crisnapati
Program	Mechatronics Engineering
Dissertation Advisor	Associate Professor Dechrit Maneetham, D.Eng, Ph.D
Academic Years	2022

ABSTRACT

The diminishing agricultural land is a problem that is being faced by developing countries like Indonesia. According to the Indonesian Central Statistics Agency, there was a significant reduction in agricultural land from 2014 to 2018. One reason is that the effort spent cultivating or ploughing the land is not worth the income obtained by farmers. This is what makes farmers decide to convert their agricultural land. This will be undeniably a serious problem for Indonesia's food security. Therefore, it is essential to combine conventional agriculture with technology in the form of a tractor that can operate automatically to help farmers ploughing the paddy fields. This dissertation presents the design, manufacture, and implementation of automation on a two-wheeled drive hand tractor. The five objectives of this dissertation were to: 1) design and create software, electrical and mechanical control system of a 2WD hand tractor, 2) collect the sensors data set of the 2WD hand tractor control system, 3) develop the path planning navigation using GIS and HTML, 4) develop and analyze GPS, accelerometer, gyroscope and compass data fusion system using the Kalman filter applied to the 2WD hand tractor, and 5) develop and analyze a Rice Field Sidewalk (RIFIS) detection system using image processing.

The methodology used to achieve these objectives included the collection of the tractor's behavior data that were controlled manually and remotely. The path-planning algorithm using the waypoint navigation method was used to complement the autonomous capabilities of the tractor. Several sensors were installed on the tractor (GPS, compass, and camera), and the data were collected using the MQTT Internet of Things protocol. Furthermore, it is necessary to apply a sensor fusion to overcome the noise generated by the sensor during data collection. The method applied both the Kalman filter

and the Butterworth filter. Meanwhile, the camera installed on the tractor produced video dataset recordings used as an input for the Rice Field Sidewalk detection process. The Mask-RCNN method was selected and tested as a detection algorithm.

In this study, the five primary objectives have been accomplished. The initial objective was to construct TROLLS: Tractor Controlling System, which combines software, electrical, and mechanical components to enable the remote tractor control. The initial prototype tested on the tractor Quick G-3000 was then reviewed. The study and assessment results were then applied to the Quick G-1000 tractor. The second objective was achieved during the field trial by recording the sensor and video data. This Rice Field Sidewalk (RIFIS) dataset is a compilation of GPS, compass, and camera sensor readings utilized to accomplish objectives four and five. In addition, the third objective involves the development of a path-planning platform based on Laravel and Google Maps, with the starting point, ending point, and puddler's distance serving as initial inputs. As an automation strategy for the tractor, the generated path results are deployed as an input for the waypoint navigation. The sensor readings are less steady based on the data acquired for the third objective; the Kalman filter eliminates the noise. For the fifth objective, an early investigation was done by utilizing Deep Learning approaches to detect RIFIS. Based on the research, an autonomous tractor control system has been created to help farmers in the process of ploughing fields.

Keywords: hand tractor, automation, control system, internet of things, kalman filter

Acknowledgments

First of all, I would like to express my sincere gratitude to E-Cube-I (E3I) Scholarships 2020, Rajamangala University of Technology Thanyaburi (RMUTT), for letting me be part of this incredible programme. I would also like to express my gratitude and appreciation to ITB STIKOM Bali for the financial support of this research. Further, I would like to thank my supervisor, Associate Professor Dr. Dechrit Maneetham for the thoughtful comments and recommendations on this dissertation, whose guidance, support, and encouragement have been invaluable throughout this study. I am also thankful to the Faculty of Technical Education RMUTT and all its member's staff for all the thoughtful guidance. To conclude, I cannot forget to thank my family and friends for all the unconditional support during this very intense academic year.

Padma Nyoman Crisnapati

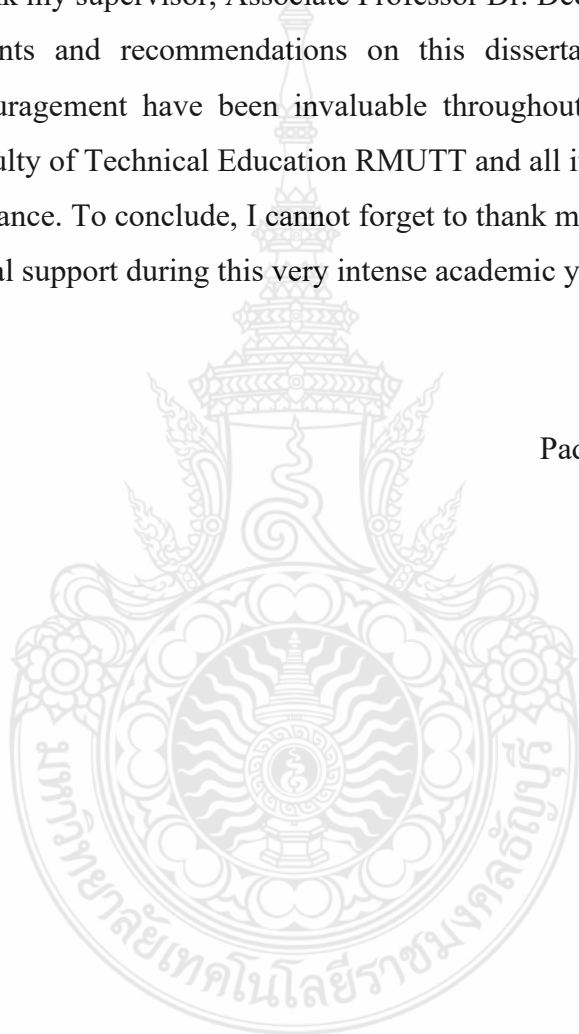


Table of Contents

	Page
ABSTRACT.....	(4)
Acknowledgments	(6)
Table of Contents.....	(7)
List of Figures	(9)
List of Tables	(12)
CHAPTER 1	13
INTRODUCTION	13
1.1 Background and Statement of the Problem.....	13
1.2 Purpose of the Study	19
1.3 Research Questions and Hypothesis	19
1.4 Theoretical Perspective	20
1.5 Delimitations and Limitations of the Study	20
1.6 Significance of the Study	20
CHAPTER 2	22
THEORIES AND LITERATURE REVIEW	22
2.1 Introduction.....	22
2.2 Literature Review.....	24
2.3 2WD Hand Tractor.....	27
2.4 Internet of Things.....	29
2.5 Embedded System.....	30
2.6 Data Fusion	31
2.7 Inertial Navigation System (INS).....	32
2.8 Global Positioning System (GPS).....	32
2.9 Compass Sensor	34
2.10 Camera as Sidewalk Detection.....	35
2.11 Kalman Filter Algorithm.....	35
2.12 Coverage Path Planning	36
2.13 GPS Waypoint Navigation.....	37
CHAPTER 3	38
RESEARCH METHODOLOGY	38

3.1	Theoretical Framework	38
3.2	System Requirement	39
3.3	Hardware Design, Assembly, and Testing.....	41
3.4	Software Design and Deployment	42
3.5	Procedure of the Data Collection	44
3.6	Hardware Software Integration and Testing	45
3.7	Path Planning Related Works.....	46
CHAPTER 4		57
RESEARCH RESULTS		57
4.1	Mechanical Design and Development	57
4.2	Electrical Driver Unit Design.....	61
4.3	Software Development.....	62
4.4	TROLLS: Tractor Controlling System.....	67
4.5	Path Planning	74
4.6	RIFIS: Rice Field Sidewalk Detection Using Machine Learning.....	82
CHAPTER 5		94
CONCLUSION AND RECOMMENDATIONS		94
List of Bibliography.....		97
APPENDICES		115
APPENDIX A		116
PYTHON PATH PLANNING SOURCE CODE		116
APPENDIX B		125
PYTHON GENERATE GRID MAP MATPLOTLIB.....		125
APPENDIX C		132
MASK-RCNN GOOGLE COLAB SOURCE CODE		132
APPENDIX D		136
INTERNATIONAL PUBLICATIONS		136
Biography.....		138

CHAPTER 1

INTRODUCTION

1.1 Background and Statement of the Problem

Smart farming has become a new trend in agricultural development. Technologies based on GPS (Global Positioning System), Sensor Fusion, IMU (Inertial Measurement Unit), and Computer Vision are widely used in modern agriculture (Catania et al., 2020; Hadas et al., 2019; Hasheminasab et al., 2020; Kragh et al., 2017). Paddy fields are one of the sub-farms that provide staple food (Syuhada et al., 2020) this place makes it possible to combine traditional agricultural models with modern technology.

Generally, paddy fields are used for rice cultivation, but several steps must be taken before the rice planting process. This stage consists of field ploughing, cultivating rice, treating the land, pests and preventing disease, and harvesting. Each of these stages has an important purpose, and if not carried out, the process after will not run optimally. Among them, one of the most important processes is the ploughing.

Ploughing is one of the greatest energy consumers in agriculture (Namdari et al., 2011). This is an activity of cultivating the land by rotating the soil so that the soil becomes smooth and easy to plant. The process consists of two processes (loosening and refining the soil). These two processes take a long time because, in this process, the soil must be soft so that it will be easy to carry out planting rice. The length of time it takes to complete the tillage process is determined by several factors, one of which is the structure and texture of the soil (S. Li et al., 2021). The process of ploughing the soil uses a machine called a tractor engine. Various types of tractors exist and are currently being used in different parts of agriculture sectors, both two-wheeled and four-wheeled tractors (Moinfar et al., 2020).

Farmers in some Indonesian provinces primarily use two-wheel drive (2WD) hand tractors. This tractor has a control in the shape of the letter "V," with clutch handlebars at the end that is used to turn the tractor left or right. This type of tractor uses

gasoline. However, the control process is still manual or uses human power/operators (Syuhada et al., 2020). When seen from the area of agricultural land to be ploughed, this process can tire the farmer operators of controlling the tractor for a long time, particularly in scorching weather. These problems can make farmers feel tired quickly during the tillage process and make tractor operation inefficient.

Several studies in control systems have been conducted on two and four-wheeled tractors to cope with these problems. A multi-modal dataset was compiled utilizing a variety of cameras and sensors to detect static and moving objects to serve as the foundation for the development of automated agricultural vehicles (Kragh et al., 2017). Research on Precise Point Positioning (PPP) in Precision Agriculture (PA) was also conducted utilizing GNSS technology on big four-wheel tractors. (Guo et al., 2018) (H. Wang & Noguchi, 2019a) (Binh et al., 2019) (Alipour et al., 2019). Several of these studies have resulted in the development of a four-wheel tractor control system at a relatively high cost to Indonesian farmers (Fabbri et al., 2017)(Shyrokau et al., 2018)(Fang et al., 2017)(Kassaeiyan et al., 2020). Furthermore, numerous studies have been conducted to optimize the tractor's mechanical performance (J. Han et al., 2017; Janulevičius et al., 2018; Shafaei et al., 2019, 2020a, 2021; H. Wang & Noguchi, 2018; Xiao et al., 2018), embedded control systems (Das et al., 2020; Ding et al., 2021; Gupta et al., 2019; Javad & Saeid, 2021; Yin et al., 2020; Zhou et al., 2020), and software (Ospina & Noguchi, 2020; Shafaei et al., 2018, 2020b; Soylu & Çarman, 2021; C. Wu et al., 2019), enabling it to operate and be controlled by a distance. These comprehensive studies are focused on powerful four-wheeled tractors.

This study presents a novel design, implementation, and testing on the low-cost Trolls (Tractor Controlling System) platform for a small two-wheel-drive walk-behind hand tractor (Quick G3000 and G1000). Based on our knowledge, this is the first time a remote-control system has been used to operate this tractor from a certain distance. The control command was sent via Bluetooth, and the HC-05 module was utilized as the receiver in Arduino. This platform was tested in the Quick G3000 and G1000 tractors. These tractors were selected considering that many farmers in Indonesia use this tractor. This research has two main contributions; the first is the novel design and implementation

of the mechanics, electronics, and software of the G3000 and G1000 tractors. The system prototype was implemented in G3000, and the commercialized ready final product was implemented in G1000.

With this platform, farmers can control their tractors remotely using smartphones to plough rice fields without having to heat up and experience fatigue. The maximum control distance is 30 meters due to Bluetooth's signal range limitation. The second contribution is the tractor's mathematical model, which simulates the tractor movement using two comparison controllers, Pure Pursuit Control (PPC) and Supervisory Logic Control (SLC). This model creates tractor behaviour and movement simulations in the Matlab application. A pure pursuit controller algorithm moves the tractor model simulation autonomously from one coordinate to another. A pure pursuit controller was chosen because using a high-speed tractor in the rice field tillage process is unnecessary; this algorithm is solid and easy to implement (Kapsalis et al., 2021; Rains et al., 2014). The researcher also uses a supervisory logic controller to compare (Mebarki et al., 2015; Tomera, 2016). The simulation results are made using Matlab/Simulink, and some experiments are presented and discussed. As a preliminary study, the researcher collects data from GPS and compass sensors with the Internet of Things technology in the field-testing process. Because the data obtained has a lot of noise and is less stable, filtering is needed. The researcher compared two filters (Kalman and Butterworth Low Pass) to find the best possible one. In addition to developing tractor control systems, this research focuses on adapting the autonomous mobile robots concept to tractors.

Mobile robots have been effectively used to perform vital unmanned operations in various situations during the last few decades, including military, industrial, security, and agricultural. (Castillejo et al., 2020). Mobile robots are increasingly being used in modern agriculture because the number of farmers is decreasing, necessitating more effective farming practices through agricultural mechanization. Agriculture has been mechanized in most field operations, including tillage, transplantation, agrochemical application, harvesting, and drying. While most agriculture operations are being explored for automation, Indonesian farmers' tillage methods are not yet automated.

Path Planning (Almoaili & Kurdi, 2020) is a critical problem that must be solved before a mobile robot may navigate and explore autonomously; this kind of robot is usually known as an Unmanned Ground Vehicle (UGV). The UGV can search for pathways based on their start and finish points, the surrounding environment, and specified parameters. With path planning, the UGV can save time and significantly reduce the UGV's wear and tear and the associated costs. It is an intriguing subject of study regarding the critical significance of path planning for UGVs.

UGV applications have exploded in popularity during the last five years (Bonadies & Gadsden, 2019). Survey missions for two-dimensional (2D) coverage have demonstrated exceptional performance among diverse uses. For instance, two-dimensional and three-dimensional mapping (Asadi et al., 2020), search & rescue (Qi et al., 2021), in combination with an Unmanned Aerial Vehicle for disaster & emergency management (RADMANESH et al., 2021), or precision agriculture (Fotio Tiotsop et al., 2020). The survey mission performed by UGV on agriculture is generally separated into two stages: i) the preparation stage, during which the vehicle is selected, the embedded system configured, and the path is planned; and ii) the execution stage, during which the vehicle operates autonomously and gathers data. Path planning must be completed to fully automate the operation, which is described as calculating paths for the robot to traverse the Region of Interest (ROI). (Galceran & Carreras, 2013). The problem's complexity has been determined as NP-Hard. (Arkin et al., 2000), However, assuming that traveling in a straight line is the quickest and most efficient way to traverse the entire landscape, a sophisticated solution involves reducing the problem and computing the path with the least tracing, e.g., (Huang, 2001) and (Santos et al., 2020). While several of these studies have produced significant results, they do not account for the path factors associated with the user's start and finish points. The puddler widths can be changed because the UGV is configured as a Walk Behind Hand Tractor. The path planning must account for additional criteria, such as interval tillage line distance from the user. Taking these three factors into account will affect the accuracy of land management while utilizing various models of tractors; this requires a path design that considers the mission's distance interval and the mission's starting and ending sites.

Due to limited field resources, efficient path computation is required; this can be accomplished by moving the Tractor in a straight line, forming a back-and-forth path (BFP) (Vasquez-Gomez et al., 2020). This study analyses convex polygon boustrophedon routes, a fast algorithm for determining BFP coverage on an ROI, considering the distance interval, start, and finish points (Figure 1.1). The decomposition of Boustrophedon cells is a widely used technique for coverage. The cells of the boustrophedon are filled in a simple backward and forward motion. Once each cell is closed, the entire environment is sealed (Choset & Pignon, 1998). As a result, the scope is narrowed to identify the whole path through the graph, reflecting the boustrophedon decomposition's cell proximity connection. This strategy is well-suited for plowing missions with defined start and finishes points and coverage.

This path planning is a continuation of the previous work, TROLLS: Tractor Controlling System for Walk-Behind Hand Tractors (Crisnapati et al., 2023), presented in its first version. The initial version can control the Tractor remotely without automation. This article presents novel results in an open-source web-based platform and Google Maps for planning rectangular polygon paths that consider distance intervals mission start and finish points. This path planning is the first step towards an autonomously operating tractor engine validated in field trials using the Tractor G1000 manufactured by Quick. The plowing mission was carried out as a validation of the success of the path-planning platform. The Tractor used in the field test is equipped with an embedded system platform that allows the Tractor to move autonomously based on the path planning platform's waypoints.

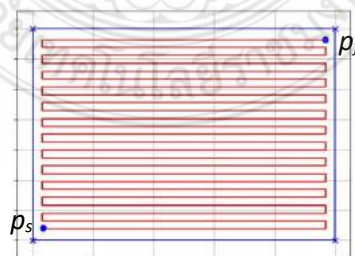


Figure 1.1. Path Planning Pattern for Tractor Movement

As additional preliminary research, a rice field sidewalk detection system is also proposed to overcome the expensive Lidar sensor. The rice field sidewalk line is a thin boundary that becomes the boundary of a rice field in Indonesia. This part assists in isolating the observed rice field region, which can ultimately be used as a computational reference for image processing for tractor automation, particularly in the plowing process. Consequently, rice field sidewalk identification is a key function in agricultural computer applications for tractor (Jeon et al., 2021; Rondelli et al., 2022) navigation, UGV (Cutulle & Maja, 2021; Singh et al., 2020), monitoring (Quaglia et al., 2019), object detection (L. Wang et al., 2019), tracking (de Simone et al., 2018), distance calculation (Zhao et al., 2022), collision avoidance (Mammarella et al., 2020, 2021), and path planning (Zoto et al., 2020a). Rice field sidewalk detection is a challenging task. A rice field scene's abundance of elements contributes to its complexity. Strongly linear foreground or background objects and environmental variables are prominently featured. Grass, soil, puddles, clouds, paddy field structures, and background landscapes are strong sources of linear features. Rice field sidewalk (RIFIS) partial occlusion is possible because the horizon line may not traverse the entire width of the image, and its visibility is localized to a small section or region of the image.

This scenario presents an additional difficulty for RIFIS detection methods based on projection-based computer vision, as they seek the presence of linear features in an image by employing edge detection methods and linear transformation. Variable illumination, grass, puddles, and the resemblance between the rice field region and the sidewalk present an additional obstacle for the RIFIS detection algorithm. Depending on the level of gloss and glare of the water surface in the rice fields, there may be a slight color variation between the sidewalk and the rice fields. Moreover, atmospheric conditions can alter the hue of puddles. The current scenario presents a difficulty for the RIFIS detection approach, which attempts to distinguish sidewalks from rice fields through image processing.

For testing and performance evaluation, the method that seeks to address the problem of RIFIS identification requires collecting benchmark image data of rice fields. The dataset is the sole benchmark for evaluating the robustness of a procedure.

Researchers have offered numerous datasets of rice field imaging; however, limitations to seedling(Yang et al., 2021), disease (Kiratiratanapruk et al., 2020; Nguyen et al., 2021; Yakkundimath et al., 2022), height(Lee et al., 2018), varieties(Qadri et al., 2021), growth (Chang et al., 2021; Ramadhani et al., 2020), and pests (Dadashzadeh et al., 2020) rather than rice, the absence of background objects, low-resolution photos, and the lack of a RIFIS in this collection leave room for development. This research offered a RIFIS image dataset that satisfied the requirement by including distinct RIFIS characteristics in ploughing fields using hand tractors. The dataset primarily focused on computer vision and deep-learning-based RIFIS detection techniques. The entire dataset comprised 18 videos, 3723 high-definition RGB images (1920×1080 pixels), and 970 labeled images. These images combined nineteen distinct characteristics for testing and evaluating the RIFIS detection algorithm. As an evaluation of the developed RIFIS dataset, Mask R-CNN was used as validation. This Mask R-CNN model was used because of its popularity in detecting various objects (Blok et al., 2022; S. Wang et al., 2021; Warden & Situnayake, 2019; Yu et al., 2019). According to researcher knowledge, no other publicly available dataset currently contains images of these RIFISs.

1.2 Purpose of the Study

An architecture for the Automation Embedded Control System was made for a 2WD Hand Tractor as an Unmanned Agricultural Vehicle. Data collection (GPS, Accelerometer, Gyroscope, Compass, and Camera) on a 2WD hand tractor will be gathered, and the automation portion will be handled by the Kalman Filter algorithm.

1.3 Research Questions and Hypothesis

Several experiments will be carried out to answer the following questions:

1. How to design and create software, electrical and mechanical of a 2WD hand tractor control system?
2. How to collect the sensors data set of the 2WD Hand Tractor Control System?
3. How to develop Path Planning navigation using GIS and HTML?

4. How to develop and analyze GPS, Accelerometer, Gyroscope, and Compass data fusion system using Kalman Filter applied to the 2WD hand tractor?
5. How to develop and analyze a rice field sidewalk detection system using Image Processing?

1.4 Theoretical Perspective

Several interconnected fundamental theories are employed as a foundation in this study; below are those theories:

1. The Fundamental of GPS Waypoint Navigation.
2. The Fundamental of Accelerometer, Gyroscope, and Compass Sensor.
3. The Fundamental of Image Processing for Rice Field Sidewalks Detection.
4. The Fundamental of the Kalman Filter Algorithm.

1.5 Delimitations and Limitations of the Study

Several scopes have been used in this research to limit the work to make it more specific and focused. The scope of what will be done is just as follows:

1. The automatic control system process runs based on data from sensors embedded in the tractor.
2. The Kalman Filter algorithm is used to reduce the noise caused by the noise of sensor readings.
3. The Quick G-1000 and G-3000 tractors are used in this research. G-3000 will be used for prototyping, while G-1000 will be used for the final stage.
4. The path planning algorithm applied in this study only covers an area classified as a convex region of interest.
5. Video captured with a camera is used as preliminary research for rice field sidewalk detection and is not used to make tractor movement decisions.

1.6 Significance of the Study

Ploughing paddy fields is a laborious process usually done by hand, with humans acting as tractor operators. The main purpose of this research is to develop an innovation in increasing the efficiency of time and labor in operating the Quick G-1000

and G-3000 type tractor, a legendary tractor popularly used by farmers in Indonesia. Farmers' paddy field ploughing will become more effective and efficient with this system.



CHAPTER 2

THEORIES AND LITERATURE REVIEW

2.1 Introduction

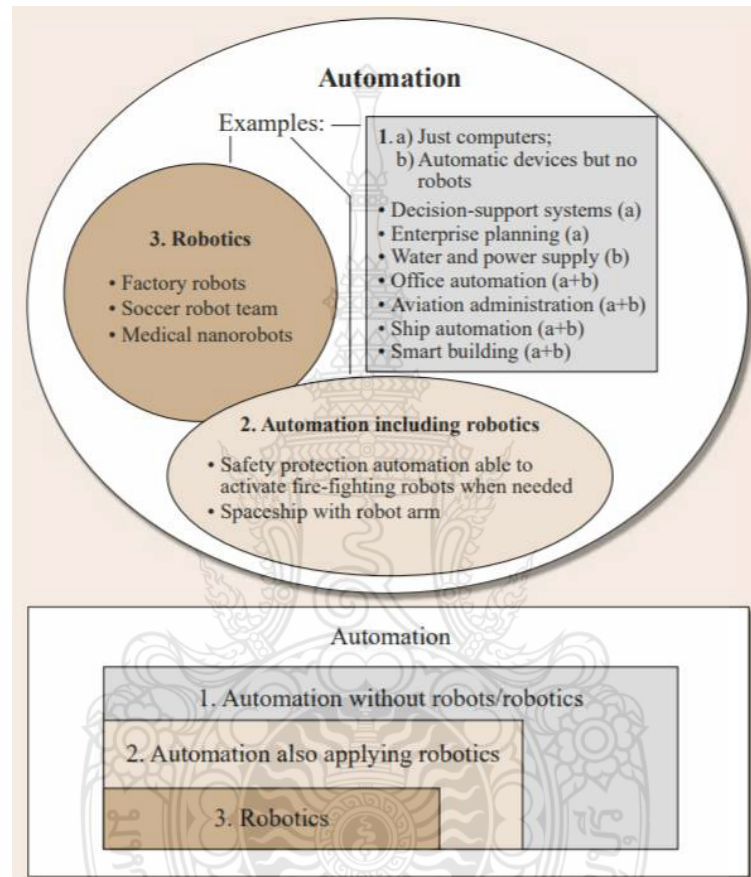


Figure 2.1 Relation Between Robotics and Automation

(SOURCE: Springer Handbook of Automation, Springer, 2009)

Automation is a general term for a platform that can operate independently without human interaction. Automatos in Greek is the origin of the word of this term, meaning acting with its own will or spontaneously. The platform involves machines, electronic devices, and systems created by humans to carry out a series of activities. Automation is also closely related to other terms, such as mechanization, cybernetics, artificial intelligence, and robotics (Figure 2.1). Some industrial fields have used

automation in facilitating jobs, such as factories, elevators, smart homes, smart cities, and agriculture.

Currently, the agricultural sector has entered a new era related to the processes that must be done, ranging from soil preparation, sowing, manure, fertilizers, irrigations, protection from pests, harvesting, and storage. Each of these processes requires a lot of effort, especially soil preparation, especially land piracy. Therefore, automation is needed in these processes, especially land piracy.

This research will create an agricultural automation platform, especially in the process of land piracy. Before starting the discussion on technical and implementation requires a study of basic theories and a review of the literature from previous research. This Chapter will be discussed several topics relevant to research, including the following:

1. Literature Review
2. 2WD Hand Tractor
3. Internet of Things
4. Embedded System
5. Data Fusion
6. Inertial Navigation System (INS)
7. Global Positioning System (GPS)
8. Compass Sensor
9. Camera as Sidewalk Detection
10. Kalman Filter Algorithm
11. Coverage Path Planning
12. GPS Waypoint Navigation

2.2 Literature Review

In this research, some of these technologies will be combined and produce a 2WD tractor that can run automatically in rice fields. Developing countries like Indonesia are dealing with the issue of dwindling agricultural land. According to the Indonesian Central Statistics Agency, there was a significant reduction in agricultural land from 2014 to 2018 (Adiyaksa & Nugroho Djojomartono, 2020; Ardli Swardana, 2020). One reason is that the effort spent cultivating or ploughing the land is not worth it to the income obtained by farmers (Wulandari et al., 2017). This is what makes farmers decide to convert their agricultural land. Of course, this will be a serious problem for Indonesia's food security. Therefore, combining conventional agriculture with technology in the form of a tractor that can operate automatically to help farmers plow the paddy fields is needed. Advanced technologies such as embedded systems, robots, sensors, GPS, and image processing profoundly influence how modern agriculture works. This progress is more efficient, safer, profitable, and environmentally friendly (Roldán et al., 2018). Several studies on precision agriculture, positioning, navigation using GPS and several sensors have been carried out on large four-wheel tractors. The recording platform was developed using a combination of several sensors to read environmental conditions such as lidar, cameras with various resolutions, IMU, and GNSS (Global Navigation Satellite System) (Kragh et al., 2017).



Figure 2.2 Platform for Recording
(SOURCE: FieldSAFE: Dataset for obstacle detection in agriculture, 2017)

To improve the accuracy of tractor position detection, research has been conducted on multi-GNSS receivers on large four-wheel tractors (Guo et al., 2018).

Several other studies using four-wheel tractors have also been developed and tested (Binh et al., 2019; H. Wang & Noguchi, 2019b).



Figure 2.3 Multi GNSS receivers

(SOURCE: Multi-GNSS precise point positioning for precision agriculture, 2017)

However, based on researcher knowledge, the data set and platform available today only on large tractors with four wheels. Some of the sensors used are quite expensive to purchase by farmers in Indonesia. Therefore, this research presents the data set, design, manufacture, and implementation of automation on a two-wheel drive small hand tractor. GPS, accelerometer, gyroscope, compass, and camera will be used as input data and collected. Meanwhile, the automation process will be done by fusing GPS and IMU using the Kalman filter algorithm. Kalman filter was chosen because it has several advantages, such as light memory usage with good accuracy, making it suitable for real-time problems and embedded systems (Ahmadi Jeyed & Ghaffari, 2019; X. Han et al., 2017; H. Kumar & Pimparkar, 2018; T. Wu & Hung, 2017). The camera data are also collected to detect objects around the tractor and detection of rice field sidewalks using an image processing algorithm. All data from these sensors will be processed on an ESP32. This microcontroller is used as the main processing tool of the tractor embedded control system.

Table 2.1 Table Literature Review

Source	Research	Type of Article/ Theory/ Research/ etc.	Similarities	Differences
M. F. Kragh (2017)	FieldSAFE: Dataset for obstacle detection in agriculture (Figure 2.2)	This article describes a multi-modal dataset for agricultural obstacle identification utilizing a large four-wheel tractor.	Collecting data about navigational in agriculture.	Data collection was carried out on two different types of tractors. This study did not use a two-wheel tractor.
J. Guo et al. (2018)	Multi-GNSS precise point positioning for precision agriculture (Figure 2.3)	Through a series of studies on four-wheel tractors, the accuracy of GNSS RTK (Realtime Kinematic) was determined.	Measuring the level of accuracy of the navigation system.	This study uses tools and technologies that are quite expensive for Indonesian farmers.
H. Wang and N. Noguchi (2019)	Navigation of a robot tractor using the centimeter level augmentation information via Quasi-Zenith Satellite System	This article aims to describe tractor control using Centimetre Level Augmentation Service (CLAS) technology obtained from a commercial Quasi-Zenith Satellite System (QZSS).	Evaluate the tractor's navigation control system.	This research does not use an open-source path planning platform and cameras as sidewalk detectors in rice fields.
X. Han et al. (2017)	Development of a low-cost GPS/INS integrated system for tractor automatic navigation	Three Garmin GPS 19x HVS were utilized to create a low-cost navigation system using a triangulation method.	Developing a low-cost auto-navigation system for tractors	The research is currently in the prototype stage and has not been implemented directly on tractors; sidewalk recognition has not been accomplished through cameras, nor has a produce path planning system been utilized.

2.3 2WD Hand Tractor

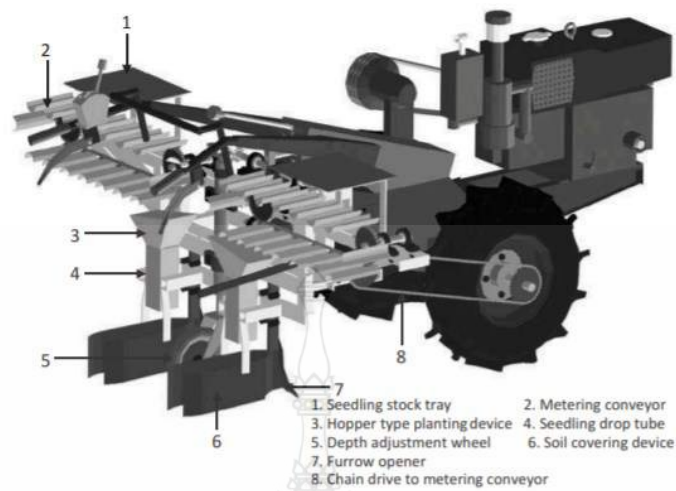


Figure 2.4 Three-Dimensional Model of Walk-Behind-Type Hand Tractor
(SOURCE: Hand-Fed Vegetable Transplanter for Use with a Walk-Behind-Type Hand Tractor, 2018)

Two-wheel tractors, often referred to as power tillers, single-axel tractors, walk-behind and hand tractors have a versatile design that can be used in various land conditions such as rice fields and soil, terracing, and gardens (Dihingia et al., 2018; Negrete, 2020). This tractor functions as a land hijacker, controlling rotors and rakes, making trenches, and even transporting (Figure 2.4). This device is an evolution of the concept of a wheeled carrier pushed by animals (Nwakaire et al., 2018).



Figure 2.5 Two-Wheeled Tractor Mexican Design
(SOURCE: Analysis of the current situation of two wheels tractors in Mexico, 2020)

A hand tractor is an agricultural machine used to cultivate the land and other agricultural work with an earth-moving tool installed on the back of the machine. This machine has high efficiency compared to bullock power [65, 66], because ground turning and cutting can be done simultaneously (Figure 2.5). This machine is a multipurpose machine because it can also function as a driving force for other tools such as water pumps, processing tools, trailers, and others. As a field ploughing machine, the tractor must be equipped with soil processing equipment, such as a chop plow, rake or rotary plow. To know a tractor as a ground processing machine, it is necessary to understand the working principles and requirements of working conditions, equipment, and their use. Hand tractors with a V-shaped frame are the most widely used machines in the paddy field ploughing process in Indonesia (Figure 2.6) (Dewangan & Tewari, 2009; Lakitan et al., 2019; Shiotsu et al., 2015).



Figure 2.6 V-Shaped Hand Tractor Machine (Quick G-1000 Type)

2.4 Internet of Things

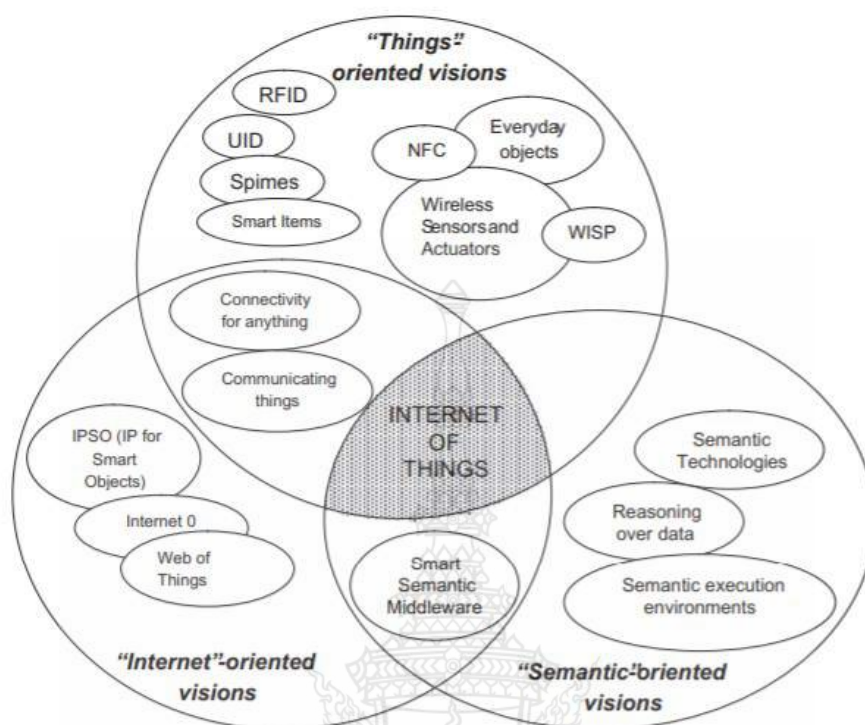


Figure 2.7 Internet of Things
(SOURCE: The Internet of Things: A survey, 2020)

The Internet of Things (IoT) is a collection of embedded technology that includes wired and wireless communication, sensors and actuators, and physical things that are linked to the Internet (Figure 2.7) (Atzori et al., 2010; Cecchinell et al., 2014). The Internet of Things (IoT) and the Unmanned Agricultural Vehicle (UAV) are two emerging agricultural technologies that are converting old farming techniques into a new era of precision agriculture (Boursianis et al., 2020). Over the past few years, agriculture has experienced its fourth revolution (Agri-food 4.0) by integrating Information and Communication Technology (ICT) in traditional agricultural practices (Lezoche et al., 2020). Technologies such as Remote Sensing, the Internet of Things (IoT), Artificial Intelligence (AI), and Unmanned Agricultural Vehicles (UAV), are developing rapidly and enabling innovations in agriculture. (Mahbub, 2020; Sharma et al., 2020).

With the presence of IoT in the agricultural environment, several physical parameters can be measured in real-time and increase aquaculture production (Nukala et

al., 2016). This technology is needed by various markets that require wireless data transmission. The sensing and communication platforms used in unmanned agricultural vehicles are a novelty with exciting potential in precision agriculture (Zhang & Kovacs, 2012). Several technologies, such as smart sensors, Low Power Wide Area Network (LPWAN), Long Distance Wide Area Access Network (LoRaWAN), Internet of Things (IoT), and others, are currently being used in precision agricultural data acquisition, data processing, evaluation and implementation. (Balafoutis et al., 2020).

2.5 Embedded System

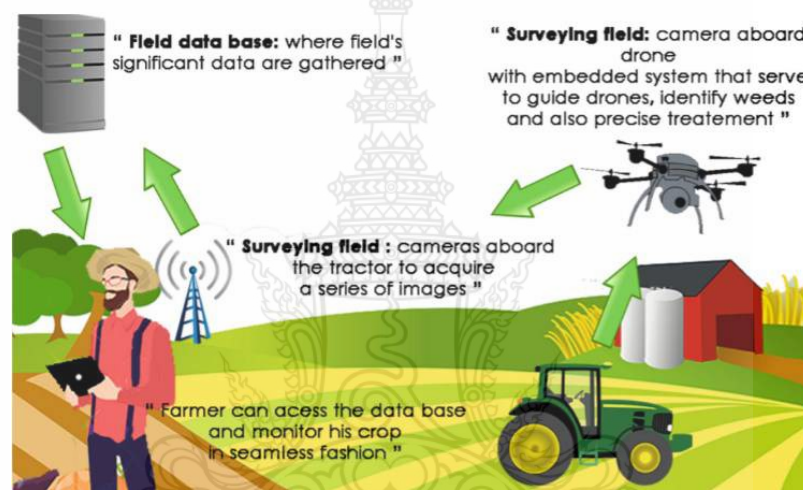


Figure 2.8 Illustration of IoT technologies for smart agriculture
(SOURCE: IoT-empowered smart agriculture: A real-time light-weight embedded segmentation system, 2017)

When the embedded system and automation concept meets agriculture, farming will be upgraded to the next level and become Smart Agriculture or Precision Agriculture (Figure 2.8) (Abouzahir et al., 2017). The application of smart agriculture is very easy to do by using an embedded system based on a microcomputer. An embedded system can simultaneously measure surrounding environmental conditions and control some of the actuators as a response (Lakhwani et al., 2019). Data collection and control were handled using a TTGO T-Call Esp32 sim8001 to find ways to make experimentation more accessible and portable. The TTGO T-Call ESP32 Sim8001 is a tiny device that supports the MQTT data handling and retrieval protocol. This device includes a 3G / 4G internet

module with a nano sim card, so it does not require additional access modules (Journal, 2021).

2.6 Data Fusion

Data Fusion is an extraordinary concept, which can be found in almost any part of modern technology that involves any kind of sensing or automation such as a smartphone or a car. This allows for extraordinary engineering achievements and their use to increase dramatically over time with new automation technologies such as self-driving cars. Overall, data fusion combines many data sources to create a picture that is more accurate, full, and resilient than any single data source. More specifically, the researcher will look at the most common use of data fusion: sensor fusion or multi-sensor fusion. This is the process by which researchers can take data from sensors and other sources of information and turn it into useful information (Bar-Shalom et al., 2011; Bleiholder & Naumann, 2009; Castanedo, 2013).

Data fusion from inertial sensors is the most common type used to get an orientation and location estimation of objects by integrating accelerometers and gyroscopes. However, this sensor has a weakness in the form of bias and sensor deviation, this problem is prone to occur in low-cost Microelectromechanical System (MEMS) gyroscope sensors. Some studies also combine the inertial sensor with a magnetic sensor. In addition, in some applications, these sensors are often combined with GPS (El-Sheimy & Youssef, 2020; Jain & Kanhangad, 2018; Ludwig & Jiménez, 2018; Sarkka et al., 2017).

2.7 Inertial Navigation System (INS)

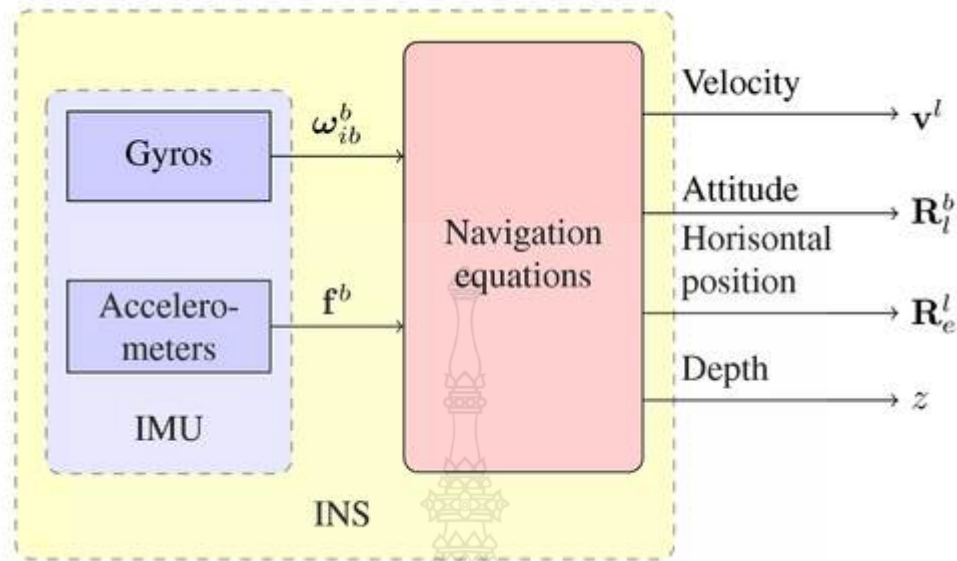


Figure 2.9 Inertial navigation system (INS)

An INS is a self-contained navigation system that may provide data on an object's speed, position, and orientation. A three-degree-of-freedom (DOF) accelerometer and a three-degree-of-freedom (DOF) gyroscope are usually used for these navigation systems to measure angular velocity and linear acceleration, respectively (Figure 2.9). These systems can usually only provide accurate solutions for a short period of time. Because the position is double integrated, any mistakes in the acceleration measurement are also integrated, resulting in a bias in the velocity estimation and aberrations of the INS's position estimate. In addition, when doing this integration, the INS code needs to use the approximated angular position of the accelerometer. The angular position is often tracked using the gyroscope sensor's angular rate integration. This also introduces an unknown bias into the integration process for determining the unit position (Berrabah & Baudoin, 2011). Therefore, it is necessary to integrate INS and GPS to get more accurate results.

2.8 Global Positioning System (GPS)

GPS technology was developed in the 1970s and is still evolving. In terms of accuracy, availability, and stability, significant improvements have indeed been

implemented. Standard Positioning Service (SPS) and Precise Positioning Service (PPS) are indeed the two key services provided by this technology (Figure 2.10). SPS is a free service that billions of worldwide civic and commercial users can utilize immediately. At the same time, PPS is a form of encrypted service that is only intended for military and government purposes. The application of SPS allows users to enjoy services through various devices such as smart watches, gadgets and smartphones, mostly utilizing signals from GPS (Kaplan & Hegarty, 2017).

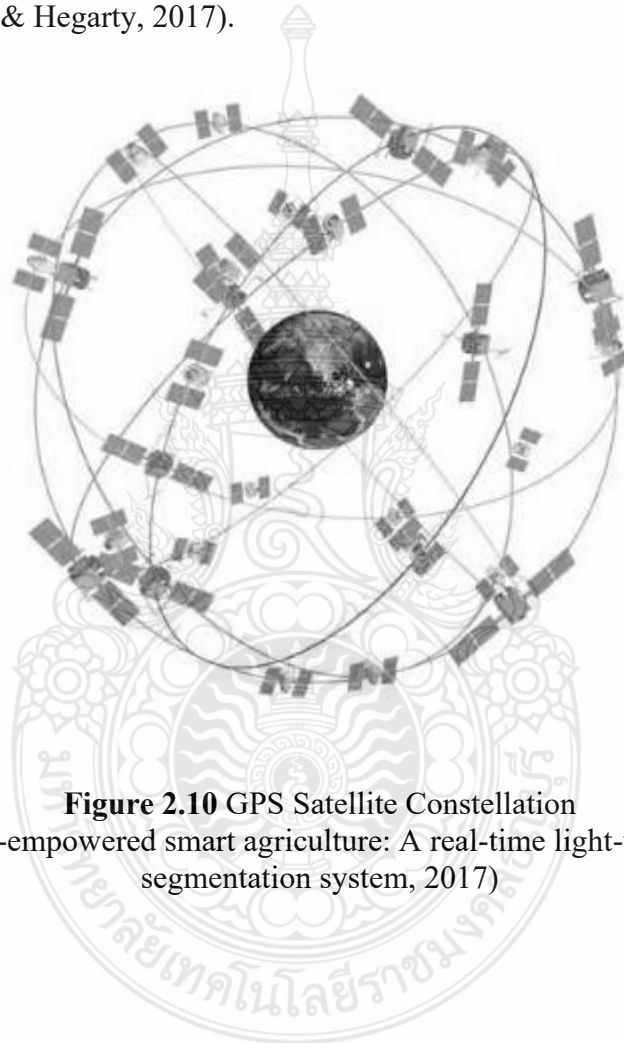


Figure 2.10 GPS Satellite Constellation
(SOURCE: IoT-empowered smart agriculture: A real-time light-weight embedded segmentation system, 2017)

2.9 Compass Sensor

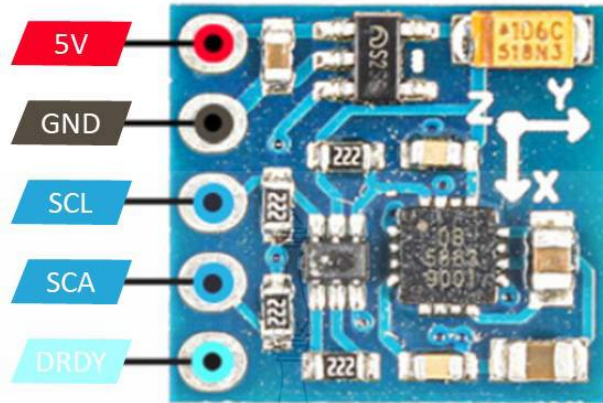


Figure 2.11 Compass Sensor GY-271 HMC5883L

Determining the direction of the magnetic has become one of the oldest methods to detect heading and orientation. When aligned with the Earth's magnet, the magnetic needle and its position are utilized as information on a thing's direction to estimate the magnet's north direction. MEMS is a technology used to manufacture electric magnetometer sensors by transferring magnetic fields into electrical signals. This technology uses three orthogonally installed magnetometers (Figure 2.11), in which there is also an analog-to-digital converter to digitize the magnetometer's electrical signal. In addition, there is also data roll and pitch angle from the compass sensor on the digital magnetic compass (Figure 2.12).(Livada et al., 2019)

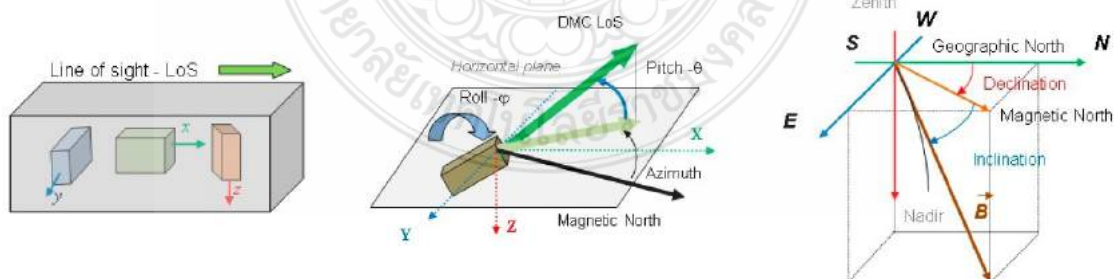


Figure 2.12 Compass Sensor Basic Configuration
(SOURCE: Digital magnetic compass integration with a stationary, land-based electro-optical multi-sensor surveillance system, 2019)

2.10 Camera as Sidewalk Detection

Some research on sidewalk feet on the highway has been done to make it easier for pedestrians, especially the blind to navigate. (Cohen et al., 2020; Jinghong et al., 2018). But in Indonesia sidewalk is not only available on the highway, there is also a sidewalk in rice fields to make it easier for farmers to navigate. Sidewalk of rice fields has different characteristics from other sidewalks, especially in its colour combination (Figure 2.13).



Figure 2.13 Rice Field Sidewalk

2.11 Kalman Filter Algorithm

Kalman Filter is a method used to combine complex data into simpler problems (Figure 2.14). This method is widely used in performing data fusion and estimation. This allows users to do and build things that were impossible before. This method is generally applied directly in complex dynamic systems, such as those used in guidance, navigation and control of cars, ships, planes, and spacecraft. Kalman Filters are widely used in robotics and manufacturing and are applicable to almost all time series analysis such as those used in signalling, economics, stock market predictions, finance and many more. The Kalman Filter was chosen after its inventor, Rudolph Emil Kalman, who was born in Budapest in 1930. This method is generally applied directly in complex dynamic systems, such as those used in navigation. (Anitha, 2018; Kitiashvili, 2019; Togashi et al., 2018).

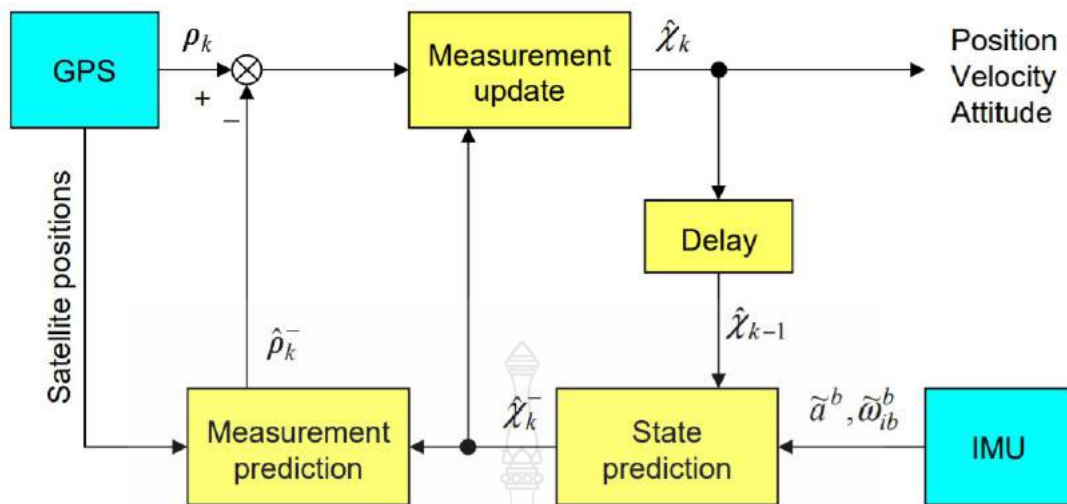


Figure 2.14 Kalman filter for the GPS/INS integration
 (SOURCE: Robust Navigation @ FOI View project Collaborative GPS/INS Navigation in Urban Environment, 2004)

2.12 Coverage Path Planning

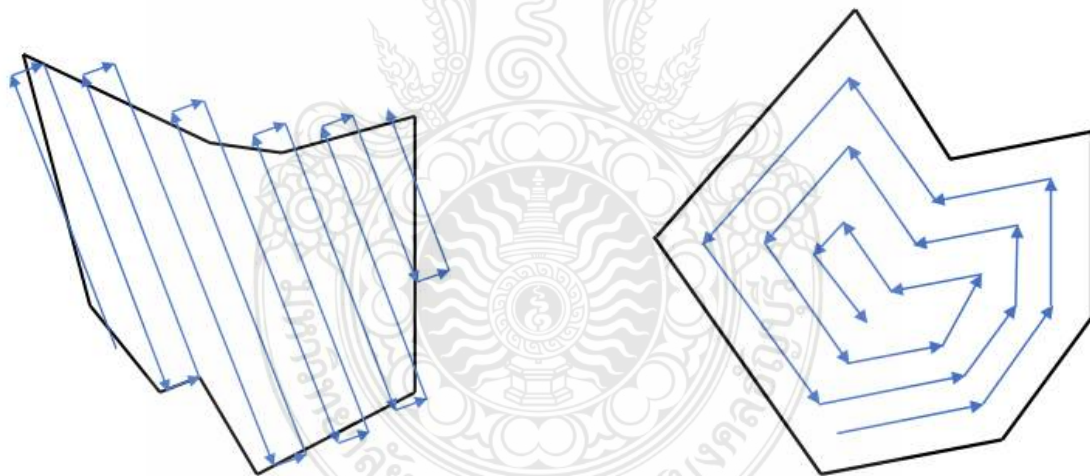


Figure 2.15 Simple Flight Pattern in Polygonal Areas

Coverage path planning (CPP) is a method used to find the route of a Region of Interest (ROI) (Figure 2.15). This method is widely used to create a flight route for Unmanned Aerial Vehicles (UAV) in several application domains namely smart farming, surveillance, 3D mapping, cleaning and tracking (Roldán, 2018)(Wang, 2019)(Han, 2017). CPP can also be applied as a guide for tractors in navigating from one point to

another in one rice field. The difference in its application in the field, UAVs can navigate freely even though they have passed the ROI while the tractor cannot pass the predetermined limit. This method will be applied to a Geographical Information System (GIS) with HTML and JavaScript frameworks.

2.13 GPS Waypoint Navigation

GPS technology is closely related to the waypoint. A waypoint is a set of position points/coordinates that have been stored (Figure 2.16). This point is a destination or intermediate point that must be passed. Waypoint has a standard form in its storage, namely with longitude and latitude formats. When one or more waypoints must be passed, this is referred to as a route or path. The sequence of points will significantly impact the route/path of the robot's journey. Several studies have also been conducted and produce waypoint navigation products in various fields (Wu, 2017) (Ahmadi,2019).

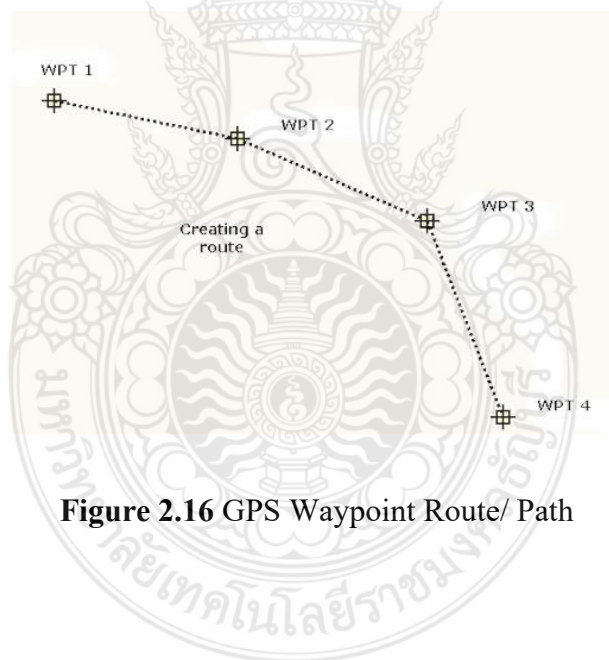


Figure 2.16 GPS Waypoint Route/ Path

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Theoretical Framework

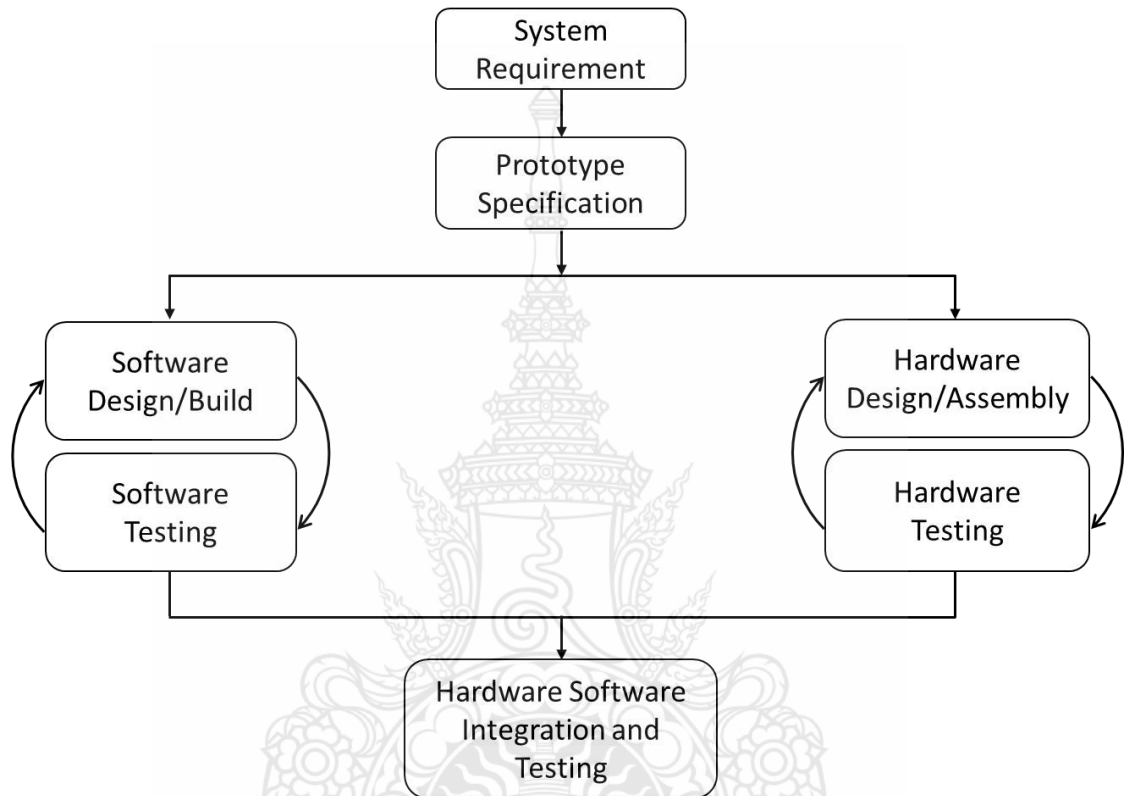


Figure 3.1. Methodology Diagram

Smart farming is the crown jewel of the technological advancement of agriculture. The literature review has given several investigations on rather expensive, high-end technology. Therefore, research is required to explicitly address Indonesian farmers' demands for low-cost technology compatible with the instruments they employ (two-wheel hand tractors). This research involves the integration of hardware and software based on system specification requirements. In the first phase, the system requirements are determined and then continued at the purchasing of all hardware components according to their specifications. The prototype specification phase included hardware prototype design and development. This prototype is field tested using the

Tractor type of quick G 3000 and G 1000. TTGO T-Call ESP32 Sim8001 is selected with several sensors connected to it. While waiting for the components to arrive, the prototype software is implemented using the Arduino IDE software interface. The software's coding is then tested on a series of hardware devices. Furthermore, hardware and software testing are carried out repeatedly. Finally, software and hardware integration will be performed, hardware components will be assembled on the board, and all experimental results will be recorded.

3.2 System Requirement



Figure 3.2. (a) Quick G3000 (b) Quick G1000

The first stage of this research is the system requirement. At this stage, an analysis will be carried out about the needs of the software and also the needs of the hardware (electronic and mechanical components) to be used. The tractors used as tests are Quick G3000 and G1000 tractors. Farmers in Indonesia widely use both tractors, and they are controlled in the same manner. The main distinction between the two tractors is in their weight, frame design, and gear box preparation. G1000 tractors are still widely used because they have a lighter weight when compared to the G3000 type.

Table 3.1 Quick G3000 Tractor Specification

Brand/Model	QUICK / G 3000 ZEVA	
Speed	1 Forward	
Transmission System	Combination (Gear-Chain)	
Gear Case	Casting Dual Part System	
Main Clutch	V-Belt (2 pcs) & Tension Pulley	
Steering Clutch	Dog Clutch (4 pcs, large)	
Lubricant – SAE 90-140 Oil	5.5 Litre	
Dimension with Cage Wheel or Tyre	Length (mm)	2725 / 2725
	Width (mm)	1130 / 865
	Height (mm)	1430 / 1360
	Weight without engine (kg)	214.8 / 165.2 *
	Weight with engine (kg)	depends on the engine
Capacity (8.5 HP and single plough)	Paddy field (hr/Ha)	± 10.28 **
	Dry field (hr/Ha)	± 9.53 **

Table 3.2 Quick G1000 Tractor Specification

Model	QUICK / G 1000 BOXER	
No. of speed	1 forward (two-way pulley)	
Transmission	Combination (Gear-Chain)	
Gear Case	Casting Dual Part System	
Main clutch	V-Belt (2 pcs) & Tension Pulley	
Steering Clutch	Dog Clutch (4 pcs, large)	
Lubricant (Viscosity Grade)	5.5 Litre (SAE 90-140 oil)	
Dimensions (Cage Wheel / Tyre)	Length (mm)	2750 / 2750
	Wide (mm)	1130 / 860
	Height (mm)	1410 / 1275
	Weight with Kubota RD 85 DI – 2S engine (kg)	292.8/ 253.8 *
Capacity (using 8.5 HP engine and single plow)	Paddy Field (hr/Ha)	± 10.46 **
	Dry land (hr/Ha)	± 9.90 **

3.3 Hardware Design, Assembly, and Testing

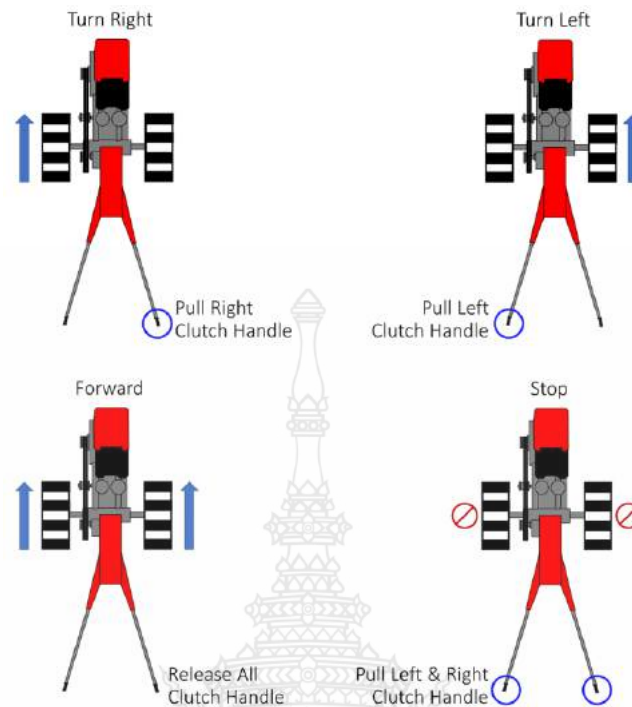


Figure 3.3. Tractor Clutch Handle

At this stage, several modifications were made to the Quick 2WD Tractor manufacturer with the G3000 and G1000 types. Modifications were made to five parts of the tractor, i.e. Head of Handle Bar, Steering Linkage, Tension Handle and Main Pipe. Each part will be given a mechanical design and an additional actuator so the tractor movement can be controlled remotely before becoming automatic. The G3000 and G1000 tractors have a similar movement control system. Both use two wheels to manoeuvre over paddy fields. Each wheel is driven by pulling two Clutch Handles independently or simultaneously. A pulled Clutch Handle will move the steering linkage, shifting rod and shifting lever. If the left Clutch Handle is pulled, the tractor will turn left and vice versa. But if the operator does not pull both of them, then the tractor will run straight and if both are pulled, then the tractor will stop. Figure 3.3 illustrates the direction of the tractor movement.

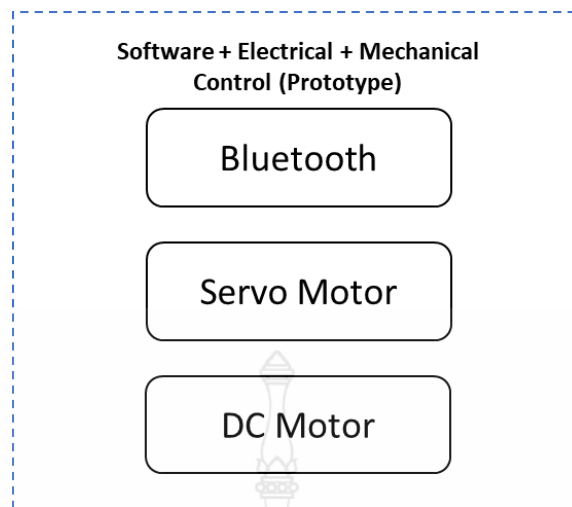


Figure 3.4 Hardware Controller Design

The G3000 and G1000 tractors have a similar movement control system. Both use two wheels to manoeuvre over paddy fields. Each wheel is steered by individually or simultaneously dragging two Clutch Handles. A pulled Clutch Handle will move the steering linkage, shifting rod and shifting lever. If the left Clutch Handle is pulled, the tractor will turn left and vice versa. But if the operator does not pull both of them, then the tractor will run straight and if both are pulled, then the tractor will stop.

3.4 Software Design and Deployment

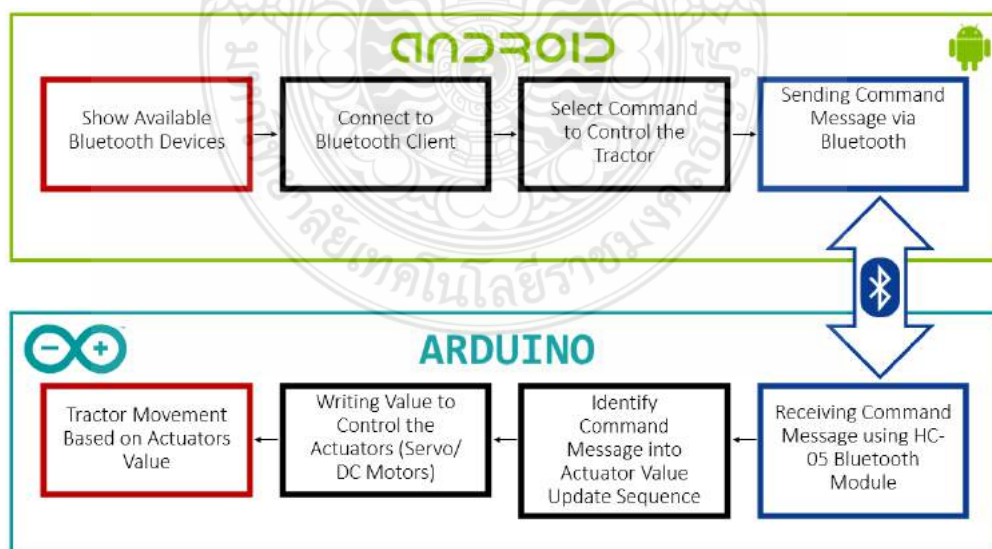


Figure 3.5 Android- Arduino Application

This phase consisted of two parts: design and development of the software embedded onto the Arduino microcontroller as the slave and Android software as the master. As a master, the Android software will send commands in the form of a message to Arduino. Software embedded onto the Arduino functioned as the slave to receive the message sent by the Android software through Bluetooth communication. The message received was then identified to obtain a set of sequence commands to do the update value towards a number of actuators such as DC or servo motors. Once the message was identified, the command was issued to write the value to each actuator later on, resulting in tractor mobility based on the values executed by the software. Android Software acts as the master of message sender to Arduino. This software shows the list of Bluetooth available surrounding and then connects them. Once they are connected, users can select the commands available. The selected commands are then sent in the form of messages through the Bluetooth communication.

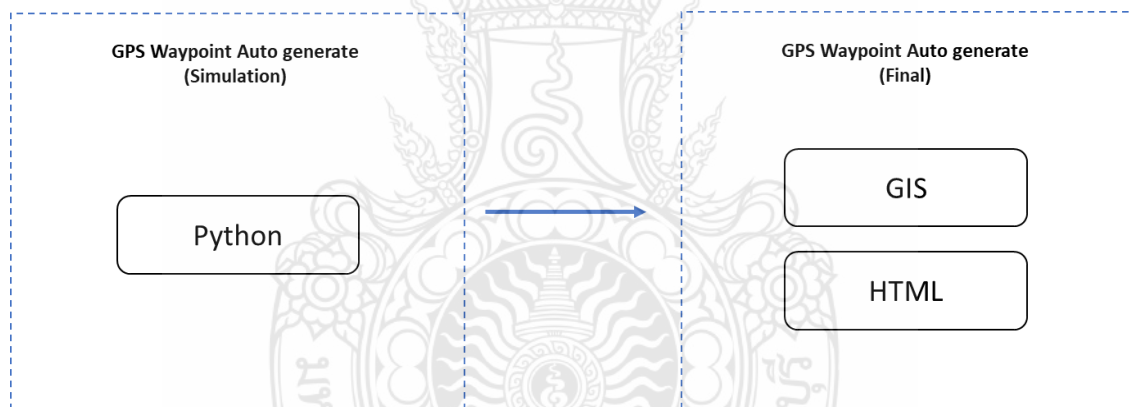


Figure 3.6 GPS Waypoint Design

After the tractor is modified and can be controlled, then a navigation system is developed for the user. Users will be able to define the boundary points of their farmland. Based on these points, a zig zag waypoint will be generated as a path to be followed by the tractor. The initial simulation of the tractor will be developed using Python (Appendix 1 and Appendix 2) and then implemented into HTML so that it can be integrated into an IoT web-based system. The application interface design of the Region of Interest application can be seen in Figure 3.7, while the application interface design of the path planning generator can be seen in Figure 3.8.

Area Coordinate Input Page

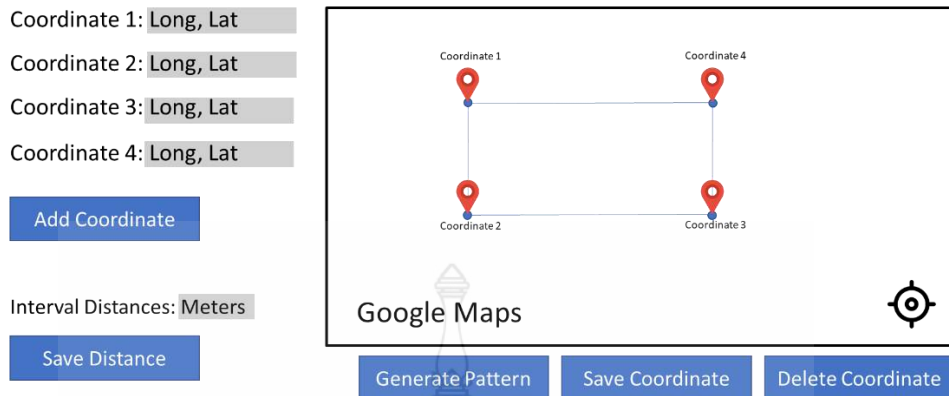


Figure 3.7 HTML Java Script Region of Interest Application Design

Generated Pattern Page

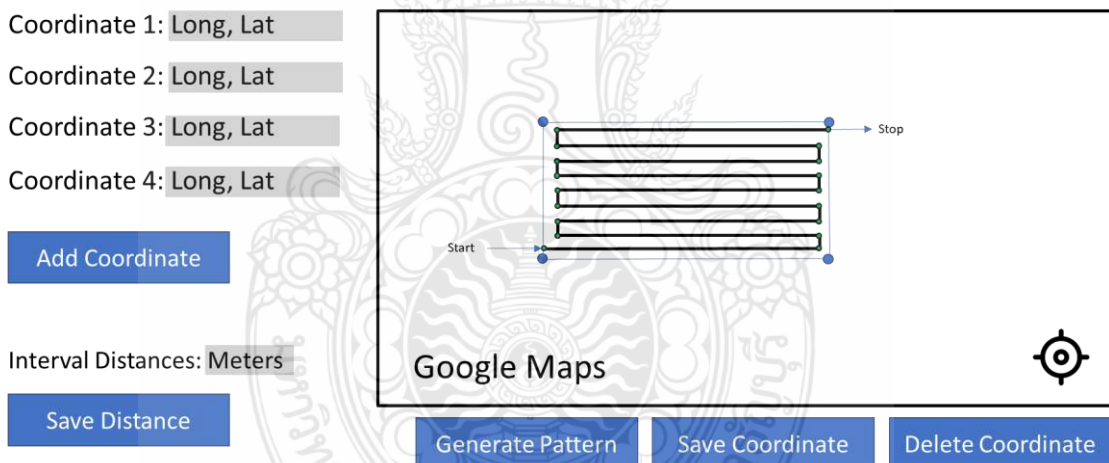


Figure 3.8 HTML Java Script Path Planning Generator Application Design

3.5 Procedure of the Data Collection

At this stage, an IoT-based system for retrieving and collecting environmental data is being developed. Data is collected using a variety of sensors, including GPS, accelerometer, gyroscope, magnetometer, GPS, and camera. Instead of using a tractor, GPS is used to retrieve location data based on longitude and latitude. The tractor's position

and direction are estimated using accelerometer, gyroscope, and magnetometer sensors. The data will be processed using the Kalman Filter data fusion algorithm. An preliminary research was conducted in this study to detect sidewalks from rice fields using camera images captured. The results of this camera vision detection are not used to make tractor movement decisions.

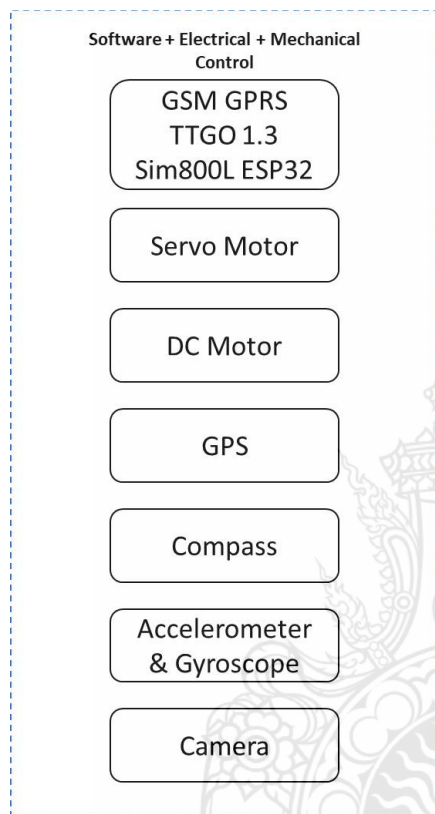


Figure 3.9 Data Collection Using IoT

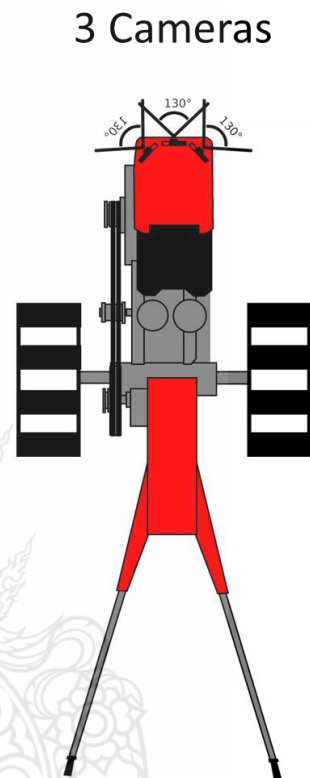


Figure 3.10 Cameras Position

3.6 Hardware Software Integration and Testing

The last stage of this research is testing the automatic control system that has been developed. The test will be carried out using a miniature tractor that has the same movement as the G1000 tractor type. Path planning applications, waypoints, electronic circuits, and actuators will be tested as part of the integration. Testing will be done in an open area.

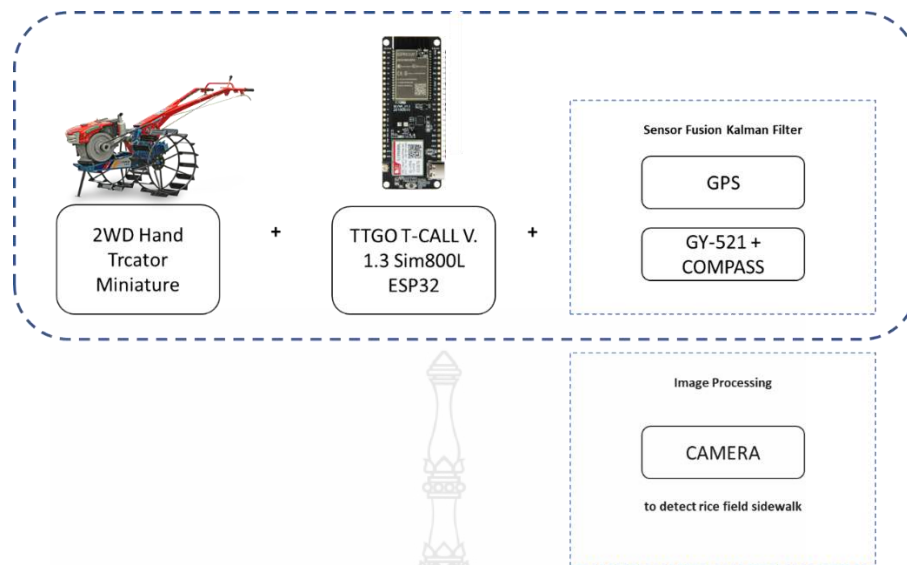


Figure 3.11 System Overview

3.7 Path Planning Related Works

Over the last few decades, path planning problems have been addressed extensively; nevertheless, the available literature focuses primarily on UGV and UAV difficulties. There are a variety of strategies and algorithms that can be employed with ground robots; in general, path planning based on ROI can be classified as solid or border representations. NP-Hard problems are used as a base to path planning with solid-representation approaches; it is emphasized on interiors rather than regions (Arkin, Fekete, and Mitchell 2000). Generally, the interior of an environment is represented as a grid of binary values or probabilities (Moravec & Elfes, 1985). This method is also known as the grid-based method (Galceran and Carreras 2013). This resolution-complete approach takes a significant amount of computer power for high-resolution maps. The grid-based process makes use of a neural network in which the region is the input (Zhu et al., 2019), divides into triangular cells, and in conjunction with a region growth algorithm, generates a path (Asadi et al., 2020), utilizing a Turing machine (Song & Gupta, 2018), or by using a genetic algorithm (Kapanoglu et al., 2012). In addition to solid representation, the boundary representation approach defines the region of interest (ROI) depending on the geometry of the polygon. The polygonal model is adopted in this

study. There are still numerous literature evaluations on alternative path planning approaches (Galceran & Carreras, 2013).

One of the most extensively used polygonal representations is Boustrophedon Cellular Decomposition (BCD); this method provides a path planning solution for traversing the polynomial world (Choset & Pignon, 1998). BCD generates a line using the idea of cells and then explores each cell using this manner, except that the robot's direction of movement is always the same. Huang (Huang, 2001) developed another coverage approach for mine operating robots that uses alternate orientations perpendicular to the ROI range instead of polygons. However, this method ignores the trip distance between the starting point and the ROI. Numerous studies have been conducted on the covering of non-convex terrain. The fundamental difficulty in this strategy is to acquire a near-optimal ROI partition and then map the coverage path to visit each division sequentially (An et al., 2020); in some circumstances, other restrictions like energy consumption are incorporated into the optimization process (Wei & Isler, 2018). In this paper, researcher focus exclusively on the convex region.

Path planning with UGVs has been generally approached from the standpoint of land-based mobile robotics but with novel functionalities. The majority of approaches adhered to Huang's optimality criteria (Huang, 2001), where the ideal path is the one with the fewest paths. (Y. Li et al., 2011) provides techniques for calculating the area of convex polygons. Convex covering area method is presented in research (Santos et al., 2020). Numerous researchers have also used path planning in agriculture, including UGV, to determine the ground features of greenhouses (Ruiz-Larrea et al., 2016) by utilizing the BFP technique and a differential robot.

Further (Ohi et al., 2018) created a robot for precision pollination in a greenhouse and implemented it on a differential robot. Path planning for an unmanned ground vehicle in conjunction with aerial images utilizing the A* search method in graphs with gradients Optimization of the descent to smooth the trajectory (Zoto et al., 2020b). Large tractor machines are used; this work optimizes the harvest area of a combined harvester robot for wheat or rice using convex and concave polygon fields(Rahman et al., 2019).

In short, the prior approach ignored the dynamic size of the tractor puddler, limiting its application to a single vehicle, and the approach was limited to establishing the beginning point and the Tractor with huge dimensions. As a result, the initial approximation is inadequate. A more accurate path can be obtained by combining the dynamic starting inputs for the interval distance, start point, and finish point and then applying these to the brief walk behind the Tractor. Abbreviations and symbols used in the article are listed in Table 3.3.

Table 3.3. Abbreviations and symbols are used in this article.

R^2	two-dimensional (2D) tractor work area	planar	$\phi 1$	latitude of the initial point
(x, y)	within the work area, a point is a location		$\phi 2$	latitude of the final point
L	line as a linear combination of two points		$\Delta\phi$	$\phi 2 - \phi 1$
a, b, c, d	four points within a Region of Interest (ROI)		λ	longitude
Q	the region of interest		$\lambda 1$	Longitude of Initial Point
V	a collection of points on a plane (vertices)		$\lambda 2$	Longitude of Final Point
E	a group of edges		$\Delta\lambda$	$\lambda 2 - \lambda 1$
$A(Q)$	polygon area		<i>earth's radius</i>	6,371km
I_x	tillage footprint length		d	distance between two points
I_y	tillage footprint width		ps	starting point
p	tractor position		pf	finish point
$p1, p2$	two points on the coverage area of the Puddler		$p0 \dots pn$	knot edge path
$C(T)$	the coverage area of a tillage line		dx	distance between tillage line
ϕ	latitude		δ	declination angle
dst	distance between two parallel lines			

3.7.1. General Definition and Notation

The Tractor in this study moves in a two-dimensional planar plane, where L is a linear combination (Figure 4.38), as in the formula (1). The distance between point a and line L is the perpendicular distance between c and a point on L (2). The distance between parallel lines $L1$ and $L2$ is denoted by (3). Equation (4) shows a line segment as a horizontal line linking two points, c and d . (Preparata & Shamos, 2012)

$$[c, d] = L(a) = (1-a)c + d \rightarrow a \in R \quad (1)$$

$$dst(a, L) = dst(a, b) \rightarrow b \in L, [c, d] \perp L \quad (2)$$

$$dst(c \in LI, LI) \quad (3)$$

$$\overline{cd} = T(a) = (1-a)c + ad \rightarrow 0 \leq a \leq 1 \quad (4)$$

Region of Interest (ROI)

$$Q = \{V, E\} \quad (5)$$

$$V = \{1, \dots, n\} \quad (6)$$

$$E = \{(1, 2), \dots, (n-1, n), (n, 1)\} \quad (7)$$

As seen in Figure 4.38, The ROI in this study is in the form of a 2D planar (convex polygon) (5), where a collection of vertices and edges is represented by the formula (6)(7) (Preparata & Shamos, 2012; Vasquez-Gomez et al., 2020). L is the support line formed by the intersection with the polygon boundary line and forms a pair of antipodal points. The distance between the two support lines is created on the same polygon and is named width (Y. Li et al., 2011). When the Tractor is used for tillage operations, it is equipped with a puddler that points directly to the ground at a consistent harrow height. The tractor location is defined as $p = (x, y)$ in R^2 , which allows for the definition of the tractor path as $s(t): R \rightarrow R^2$.

3.7.2. Two-Dimensional Leveller/Puddler Footprint

When the Tractor plows an area, the Tractor departs from point ps , following the path by carrying a puddler and heading to the finish point pf (see figure 1.1). During the tillage process, the Puddler forms an area known as the tillage footprint of $I_x \times I_y$ so that the tillage footprint can be represented by equation (8). If s is the tillage path, then equation (9) is the coverage area (ρ) of the Puddler, and each waypoint of Q is represented in equation (10). The tractor tillage mission overview is represented in Figure 3.12.

$$Tl(p) = I_x \times I_y \quad (8)$$

$$C(s) = \cup_{p \in s} Tl(p) \quad (9)$$

$$A(Q) \subseteq C(\rho) \quad (10)$$

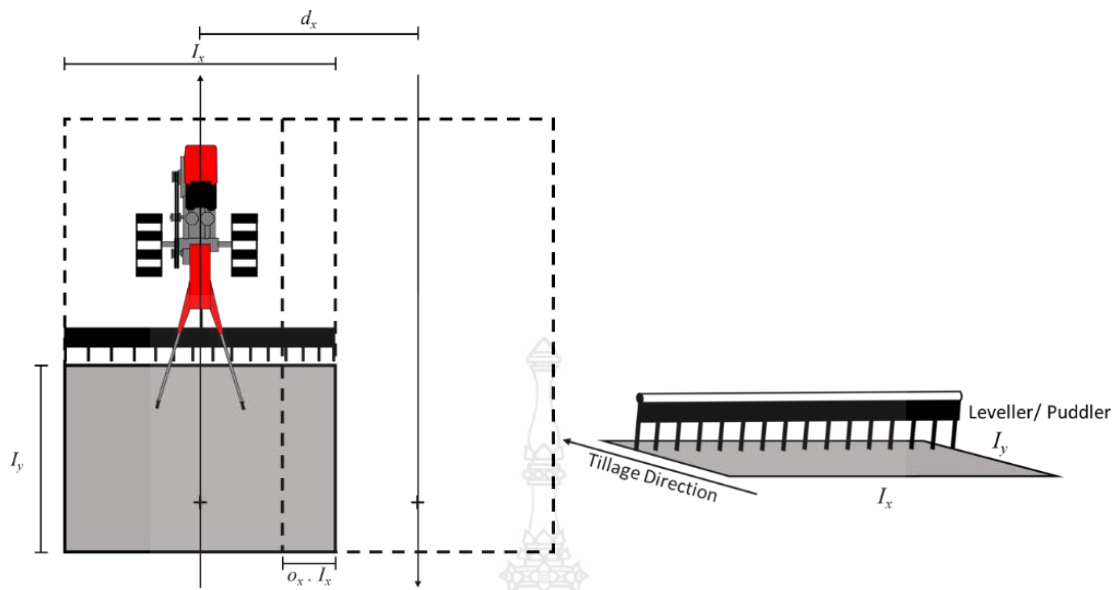


Figure 3.12. Top and Perspective View of Tractor.

3.7.3. Back and Forth Path (BFP)

In path planning, many possible path forms are created (Arkin et al., 2000). So, particular patterns such as spirals, zig-zags, stars, or back and forth are the solution to these problems. In this study, the BFP is implemented on a tractor that moves in a straight line at ROI. The advantages of BFP allow the Tractor to keep the Puddler stable and make it easy for autonomous vehicles to follow the tillage line.

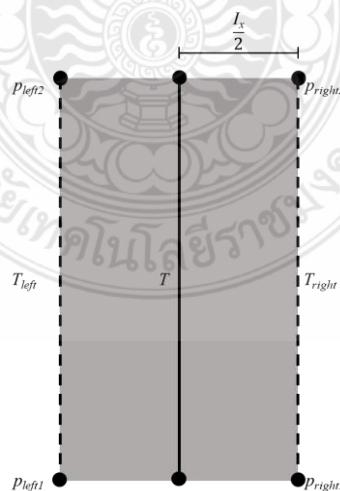


Figure 3.13. Area covered when the Tractor crosses the T tillage line

Tillage line T is a linear combination equation (11), which means that $\alpha \in R$ and $0 \leq \alpha \leq 1$ as seen in Figure 3.13. $C(T)$ is the region enclosed by T_{left} and T_{right} . These two lines are parallel to T by an $\frac{L_x}{2}$ distance. When tillage occurs at an ROI, one tillage line is insufficient to cover the entire area, so plowing with a BFP pattern is used in equation (12). Equation (13) is used to calculate the coverage area of Q .

$$T(\alpha) = (1 - \alpha)p_1 + \alpha p_2 \quad (11)$$

$$P = \{T_1, \dots, T_n\} \quad (12)$$

$$A(Q) \subseteq \cup_{F \in P} C(T) \quad (13)$$

These calculations deduce that BFP is a collection of sequential points used as a plowing route boundary or waypoint connected with a straight line between tillage lines. This research aims to develop a path planning platform for rectangular ROI convex polygons and validate the autonomous walk-behind hand tractors algorithm.

3.8 Embedded System Platform

Walk-behind tractors (affectionately referred to as walking tractors, pedestrian-controlled tractors, or power tillers) are prevalent in small and medium-sized rural communities. This Tractor offers several advantages, including its low price and ease of use. Additionally, these tractors can perform various agricultural tasks, including harvesting, crop protection, irrigation, threshing, and transporting (G. V. P. Kumar & Raheman, 2011). A G1000 Quick tractor equipped with a boxed embedded control system was used in this study. The Tractor's wheels turn due to a clutch handle being pulled. This platform is used to validate the path planning platform.

An autonomous navigation system enables tractors to navigate between waypoints autonomously (Gan & Lee, 2018). Researcher are attempting to construct an additional validation platform with limited funds. First, the list of waypoints generated by the platform informs of the final Global Positioning System (GPS) coordinates (longitude and latitude). The platform will then build a set of reference pathways (straight lines) that connect each waypoint. Additionally, these coordinates are used in a control system to direct the robot's movement between waypoints along the reference route. The results obtained from the GPS and IMU sensors contain a significant amount of noise.

This will obviously interfere with the tractor's movement system, hence the application of a filter in this system is essential. Kalman filter was selected because to its superior input stability for the controller. The controller, not the Tractor, determines the distance and heading. Figure 3.14 depicts the relationship between GPS sensors, IMU (Magnetometer), and Waypoint Techniques as inputs and outputs rather than controllers as servo motor drives.

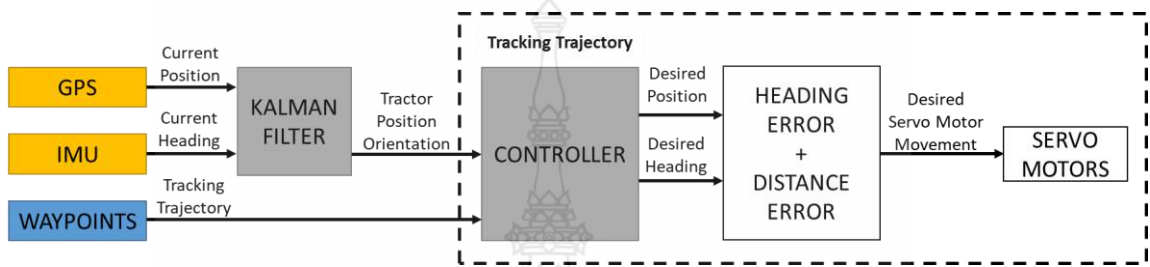


Figure 3.14. Waypoint System Overview

GPS technology is used to determine the current location of the Tractor using a GPS receiver (Berber et al., 2012). GPS technology is the backbone of the Autonomous Vehicle because it receives GPS coordinates from the location; the U-Blox Neo-M8N type sensor is used in this study. As indicated in Table 3.4, this type of GPS was chosen because it is economical for Indonesian farmers and offers a reasonable degree of precision.

Table 3.4 Neo M8N First Fix and Horizontal Accuracy

Parameter	Condition	GPS	GLONASS	BeiDou
First fix time	Cold start	29s	26s	27s
	Hot start	1s	1s	1s
Horizontal accuracy	Autonomous	2.5 m	2.5 m	2.5 m
	SBAS	2.0 m	2.0 m	2.0 m

After the tractor location is obtained, the distance between the current tractor position and the target position is calculated using the Haversine equation (14)(15)(16) (Karataş et al., 2021). In addition to distance, bearing angle is an essential metric for autonomous robotics as it indicates the robot's direction and helps the Tractor follow the

correct path (Karataş et al., 2021). The angle of the Tractor bearing concerning the initial latitude and longitude coordinates can be calculated using the equation (17)(18)(19).

$$x = \sin^2(\Delta\phi/2) + \cos \phi 1 \cdot \cos \phi 2 \cdot \sin^2 (\Delta\lambda/2) \quad (14)$$

$$y = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (15)$$

$$d = y \cdot \text{earth's_radius} \quad (16)$$

$$x = \sin (\Delta\lambda) \cos (\phi 1) \quad (17)$$

$$y = \cos (\phi 1) \cdot \sin(\Delta\phi 2) - \sin (\phi 1) \cdot \cos (\phi 2) \cos (\Delta\lambda) \quad (18)$$

$$\text{Bearing Angle} = \text{atan2} (x, y) \cdot 180/\pi \quad (19)$$

The heading angle is obtained from the magnetometer sensor found on the IMU. This sensor can provide an absolute correct direction, but its efficiency is reduced when a magnetic field other than the Earth's magnetic field is present (Fouché & Malekian, 2018). One source of significant magnetometer error is the annual variation in the tilt of the Earth's axis of rotation and the Earth's rotation around the sun, also known as the angle of declination. The declination angle can be determined using the equation (20), where a day is the number of days remaining before January. The declination angle should be added to the sensor-derived compass direction (Fouché & Malekian, 2018). This inaccuracy varies according to the sensor's location and can be found at <https://www.magnetic-declination.com/> (Magnetic-Declination.com, 2022). For instance, in Gianyar, Bali- Indonesia, the declination is 0° 43' EAST (POSITIVE). Autonomous UGVs require knowledge about their travel direction. The heading angle in the horizontal plane is defined as the angle measured clockwise from true north (Karataş et al., 2021). As a result, headings that rotate between 0° and 360° refer to true north (21).

$$D = \sin^{-1}(\sin (23,45^\circ) \cdot \sin (360/365 \cdot (\text{day}-81))) \quad (20)$$

$$\text{heading} = \text{atan} (Yh/ Xh) \quad (21)$$

3.9 RIFIS (Rice Field Sidewalk Detection) Related Works

In this study, we have reviewed the publication of publicly available rice field image datasets. These datasets include (Kiratiratanapruk et al., 2020; Nguyen et al., 2021; Shao et al., 2021; H. Wang et al., 2021; Yakkundimath et al., 2022; Yang et al., 2021) with details which can be seen in Table 1. This section presents the purpose, attributes, and differences between the dataset and the dataset we collected. In this study (H. Wang et al., 2021), high-resolution image-based deep learning approaches were used to panicle

datasets. The semi-supervised deep learning model training procedure was performed to annotate and modify the training data set. UAV Seedling dataset(Yang et al., 2021), this research is focused on the annotation of the UAV picture dataset. The dataset was obtained using a UAV with many rotors that flew over rice fields to collect data. In addition, semi-automatic annotations are introduced to provide training data for rice seedling detection. Rice ear dataset(Shao et al., 2021), this research provides a dataset of 3.300 rice ear samples that illustrate a variety of complex conditions, such as variable light and complex backgrounds, rice and leaves that overlap. The acquired photos were manually tagged, and a data improvement technique was employed to expand the sample size. Research (Kiratiratanapruk et al., 2020), examined six major rice cultivars. The rice disease database contains images of rice leaves collected from the planting area's farms. Pictures are taken under an unmanaged natural environment. Using an RGB camera(Nguyen et al., 2021), captures leaf disease picture data from rice plants. This study was conducted in the Mekong Delta (VMD) rice fields in Vietnam. This study(Yakkundimath et al., 2022) is also concerned with detecting rice illnesses. A DSLR camera was used to collect 1200 experimental photographs from a rice farm located on the University of Agricultural Sciences (UAS) campus in Dharwad, India. There are 750 photos impacted by fungal diseases, 250 images affected by bacterial diseases, and 200 images affected by viral diseases in the retrieved dataset. However, the field picture dataset initially collected with 1200 labeled photos has been expanded to 12,000 labeled images by using several image enhancement methods. To our knowledge, however, the publicly available picture data set for rice fields is restricted, and no RIFIS detection is available. To address this issue, we suggest creating a dataset of rice field sidewalk images named RIFIS.

Table 3.5. Summary of previous research datasets on rice fields.

Title	Targeted Domain	Annotation Type	Number of Data	Place
Paddy Rice Imagery Dataset for Panicle Segmentation (2021)[121]	Panicle detection and segmentation tasks	Polygon	400 images	Hokkaido University, Sapporo, Japan
A UAV Open Dataset of Rice Paddies for Deep Learning Practice (2021)[123]	Rice seedling detection	Bounding boxes	Rice seedling—28,047 images, Arable land—26,581 images	Wufeng District, Taichung, Taiwan
Rice Ear Counting Based on Image Segmentation and Establishment of a Dataset (2021)[122]	Rice ear detection	Polygon	3300 images (originally 1100 images before augmentation)	Sichuan Agricultural University, Ya'an City, Sichuan Province, China
Classification of Rice Diseases using Convolutional Neural Network Models (2022)[124]	Rice disease detection	Bounding boxes	12,000 images (originally 1200 images before augmentation)	University of Agricultural Sciences (UAS), Dharwad, India
Real-time Disease Detection in Rice Fields in the Vietnamese Mekong Delta (2020)[125]	Rice disease detection	Bounding boxes	116 images	Vietnamese Mekong Delta
Using Deep Learning Techniques to Detect Rice Diseases from Images of Rice Fields (2020)[126]	Rice disease detection	Polygon	6300 images	Thailand
Proposed Rice Field Sidewalk (RIFIS) (2022)	Rice field sidewalk	Bounding boxes and Polygon	3723 images and 18 videos	Denpasar, Bali, Indonesia

The salient contributions of this dataset were (1) it was the first novel dataset for the detection of the RIFIS in a two-wheeled hand tractor; (2) the diversity of features related to the foreground and background objects, state of the fields, level of illumination, luster, glare, standing water, cloud cover, and hand tractor movement. The proposed dataset presented RIFIS images collected using a tractor movement scenario with a spiral pattern in two locations in Bali, Indonesia. Based on researcher knowledge, no other RIFIS image dataset is currently available. In addition to datasets in the form of videos and images, the researcher also collected the location data (GPS) and orientation

(accelerometer, gyroscope, and compass) of tractors during the ploughing process using the internet of things (IoT) technology.



CHAPTER 4

RESEARCH RESULTS

4.1 Mechanical Design and Development

Several modifications were made to the Quick Walk-Behind Hand Tractor manufacturer with the G3000 and G1000 types. Changes were made to five parts of the tractor, i.e., the Head of Handle Bar, Steering Linkage, Tension Handle, Throttle Lever, and Main Pipe. Figure 4.1 illustrates the position of the four parts, while the fifth part, the Throttle Lever, can be seen in Figure 4.7.

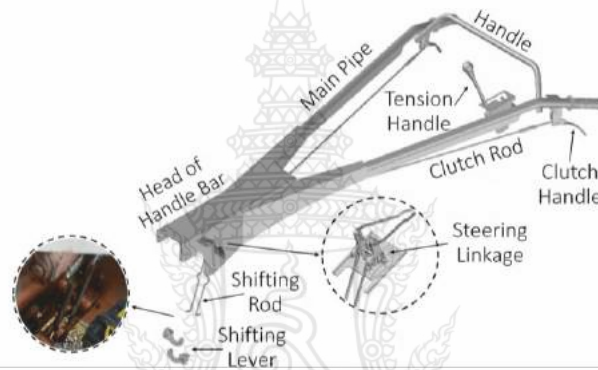


Figure 4.1. Parts of Handle Bar as the Control of the Quick G3000 Tractor.

4.1.1 Walk-Behind Hand Tractor

The two-wheel walk-behind hand tractor is a farm machine that can be used for tillage and other agricultural work with a draft device installed on the back of the machine. Compared to bull power, this machine has high efficiency (Dewangan & Tewari, 2009). It is a multipurpose machine since it can also function as a driver for water pumps, processing equipment, or trailers. The tractor must be equipped with tillage equipment, such as a chop, harrow, or rotary plough as a ploughing machine. In Indonesia, hand tractors with V-shaped frames, such as Quick G3000 & G1000, are the most widely used machines in the process of ploughing rice fields (Shiotsu et al., 2015) (Lakitan et al., 2019) (Paman et al., 2010).

4.1.2 Tractor Movement

The G3000 and G1000 tractors have a similar movement control mechanism. Both use two wheels to manoeuvre over paddy fields. Each wheel is driven by pulling two Clutch Handles independently or simultaneously. The steering linkage, shifting rod, and shift lever are moved when the Clutch Handle is retracted. The tractor turns left when the left Clutch Handle is pulled and vice versa when the right Clutch Handle is pulled. If the operator does not pull both, the tractor continues straight; nevertheless, the tractor comes to a complete stop if both are pulled. The tractor's direction of movement is depicted in Figure 4.2.

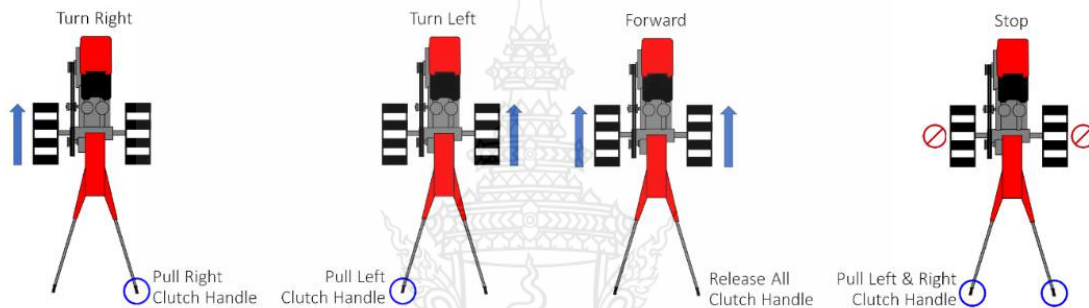


Figure 4.2. Clutch Handle Concept.

4.1.3 Handle Bar Head & Steering Linkage

In the handlebar head tractor G3000, four servo motors and two pulleys were added. This device is divided into two parts and arranged on the right and left parallel. As shown in Figure 4.3, each piece consists of two servo motors and one pulley. The servo motor and pulley are connected to the steering linkage using a stainless-steel wire rope, as shown in Figure 4.4. With this design, the tractor can be controlled to the right and left using the movement of the servo motors.

After analyzing the field testing of the prototype design, the costs incurred for purchasing four servos were found to be relatively high for the farmers. Hence, an improvement was made to the mechanical design of the G1000 tractor by using only one servo motor on each right or left side and removing the use of pulleys. As shown in Figure 4.5, the servo motor is directly connected to the clutch handle end.

Steering Linkage
Control System
4 Servo Motors
2 Pulleys

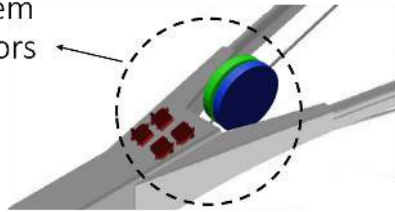


Figure 4.3 Modification of Tractor's Handle Bar.

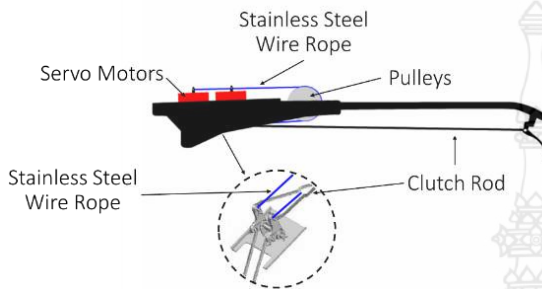


Figure 4.4. First Modification of Stainless-Steel Wire Connection.

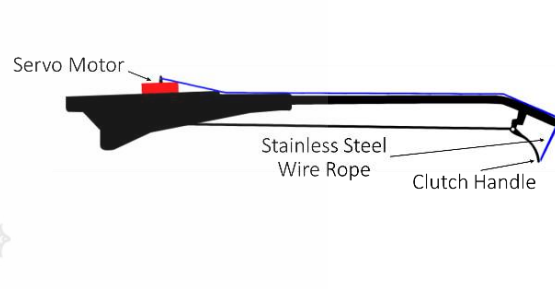


Figure 4.5. Second Modification of Stainless-Steel Wire Connection.

4.1.4 Tension Handle

The tension handle is used to manage the tightness of the V-Belt on the tractor. The tighter the V-Belt, the more speed and thrust increase. This part can be moved by pulling back or pushing forward, as shown in Figure 4.6; therefore, a modification was made to the G3000 and G1000 tractors by adding a DC motor with the concept of a screw system for linear movement.

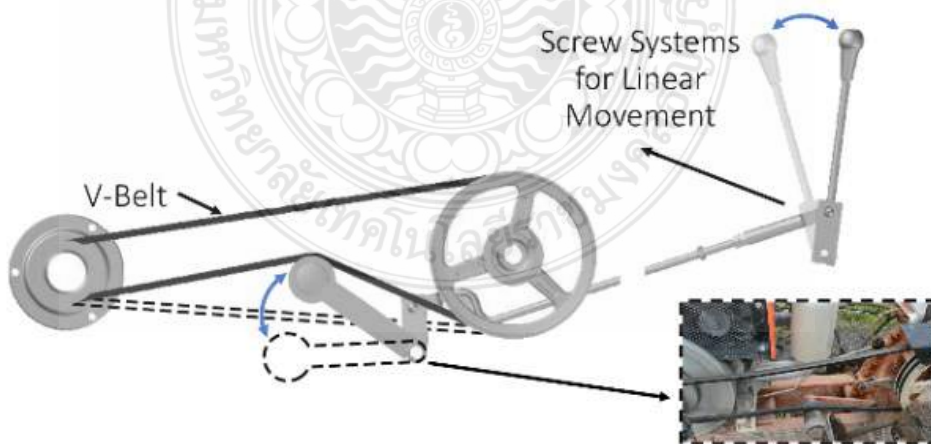


Figure 4.6. Tractor's V-Belt System.

4.1.5 Throttle Lever

Aside from managing the tension handle, farmers can adjust the tractor's speed and thrust by shifting the throttle lever to the right and left. The modification was made by adding a DC motor to be controlled remotely. The screw system for the linear movement was used to make the rotation of the motor will be converted into a right and left shifting motion. Figure 4.7 shows the Throttle Lever feeding modifications on the G3000 tractor.

Based on the field testing results, the prototype design required a long time to move the throttle lever. Based on this problem, a change was given and implemented to the G1000 tractor. The DC motor was installed without using a screw system for the linear movement concept in this design. Figure 4.8 presents the results of the revision design implementation.

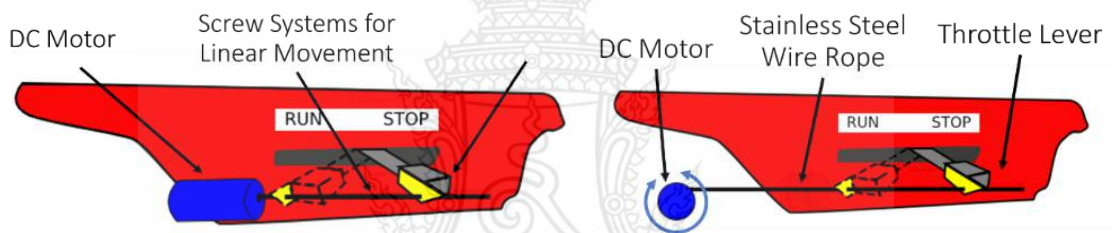


Figure 4.7. First Modification of Tractor Speed.

Figure 4.8. Second Modification of Tractor Speed.

4.1.6 Main Bar

By default, the G 3000 and G 1000 tractors have a retainer in the front to prevent them from falling forward; the front of the tractor is heavier without a counterweight in the rear. Modifications to the main bar are made by utilizing rubber wheels that are not used when the tractor is in ploughing mode and are equipped with iron wheels (Figure 4.9). With this modification, the tractor maintains a good balance and does not become stuck or sink into the mud when applied forward or reverse momentum. The tractor weight balancing hanger is installed on the main pipe near the clutch handle to ensure that the tractor balances independently of the operator. This hanger has a maximum capacity of two rubber wheels.

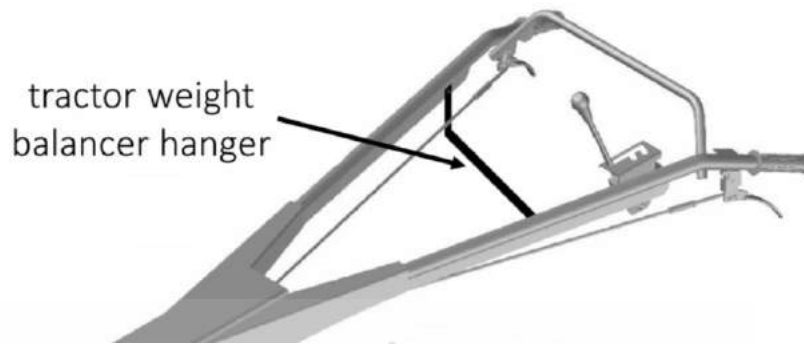


Figure 4.9. Modification for Tractor Balance.

4.2 Electrical Driver Unit Design

The electrical driver unit is a part that acts to manage the movement of the tractor based on the input received from the mobile application. This unit consists of an Arduino Uno microcontroller as the primary control device for several actuators. Several servo motors with a torque of 25 kg-cm were used to control the tractor movement to the left or right. For the G3000 tractor, four servos were used, while the G1000 tractor only used two. DC motor was used to control the tension handle and throttle lever with a voltage of 12 V and a torque of 10 kg-cm. A motor driver with type H-Bridge BTS7960 was used for these two DC motors. The HC-05 Bluetooth module communicated between the driver unit and the mobile application.

The power source is a 12V battery with a capacity of 7Ah. Some electronic modules require lower voltages, and in this design, two step-downs were used to lower the voltage from 12V to 5V and 6V. Figure 4.10 illustrates the wiring diagram of the driver unit. After completing the field test, the researcher simplified the electronic component circuit into a single microcontroller board. As preliminary research, a GPS sensor (Ublox 6M) and a Compass (HMC5883L) were installed to obtain tractor location and orientation data using TTGO-TCall ESP32 SIM800L Internet of Things (IoT) technology, details of the circuit can be seen in Figure 4.11. The researcher compared the filters to reduce sensor reading noise, namely the Kalman Filter and the Butterworth Low Pass Filter (LPF).

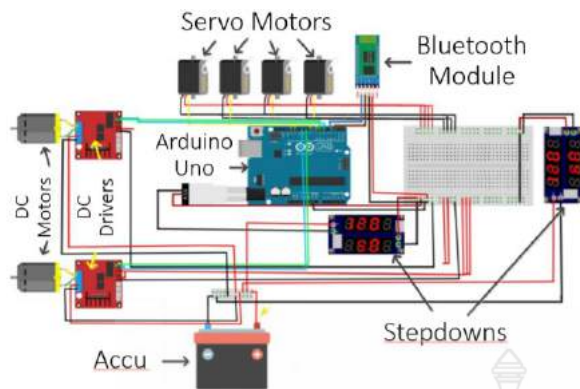


Figure 4.10. Wiring Diagram Driver Unit.

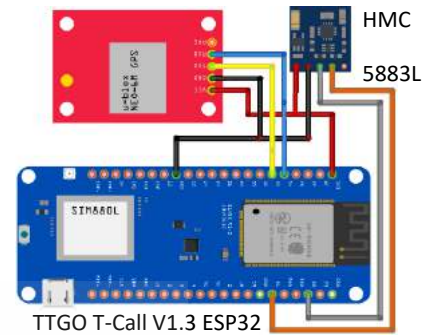


Figure 4.11. Wiring Diagram GPS and Compass Logger

4.3 Software Development

4.3.1 Arduino Software

Software embedded in the Arduino functioned as the slave to receive the message sent by the Android software through Bluetooth communication. The message received was then identified to obtain a set of sequence commands to do the updated value towards several actuators such as DC or servo motors. After the message was identified, the command was issued to write the message's value to each actuator and drive the tractor. Figure 4.12 depicts the flow diagram between Arduino and Android.

4.3.2 Android Software

Android Software acts as the master message sender to Arduino. This software shows the list of Bluetooth available surroundings and then connects them. Once connected, the users can select the available commands, as shown in Figure 4.13. The set commands are then sent in a message through Bluetooth communication.

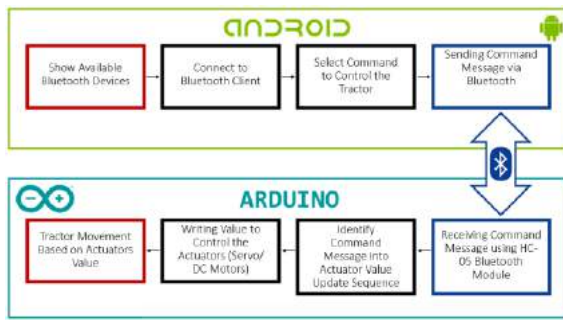


Figure 4.12. Arduini and Arduino Application.



Figure 4.13. GUI of Android Application

4.3.3 Filter Comparison for GPS and Compass Data

As preliminary research on an autonomous tractor, sensor readings and filter comparison are needed to record the tractor's behaviour. GPS and Compass are usually used for a vehicle to perform trajectory tracking. From these two sensors, several data are obtained regarding the position and orientation of the tractor. Still, the data obtained by the Ublox 6M and HMC5883L have noise and are very unstable. A filtering technique is applied to overcome this problem. Kalman Filter and Butterworth LPF are used to see the optimal filter results to reduce the noise of the two sensors.

Nomenclature for Kalman Filter			
x_k	the state vector	x_{k+1}	the process (system) model
φ_k	the state transition matrix, which connects the next time step's state vector to the current state	$\hat{U} = \hat{x}_k$	the post estimation of x_k using the linear process model
w_k	process noise	$U = y_k$	The Measurement (Sensor) Model
v_k	the measurement noise	Q_{kalman}	the covariance matrix of w_k
P_k	the error covariance between x_k , \hat{x}_k	R_{kalman}	the covariance matrix of v_k
H_k	the matrix that connects the measurement vector and the state vector	K_k	Kalman Gain

In using the Kalman Filter, the process (system) can be modelled in equation (1) with the measurement (sensor) model in equation (2). The noises in the predictions and measurement procedures were determined by measuring the shift and comparing it to predicted and measured data using the minimal error approach. See equation (3) for the covariance matrix of w_k and v_k (Q and R , respectively) derived from the inaccuracy in

predicted and measured changes compared to manually measured changes (Q. Li et al., 2016). Using the linear Kalman model, P_k equation (6) (Q. Li et al., 2016) represents the error covariance between x_k, \hat{x}_k equation (4), and (5) (Q. Li et al., 2016). Equation (7) gives the value of the Kalman gain (K_k) that minimizes the sum squared error (Q. Li et al., 2016), H_k and k is constant. Algorithm 1 is the application of the Kalman filter.

$$\begin{aligned}
 x_{k+1} &= \varphi_k x_k + w_k && \dots\dots\dots (1) \\
 U &= y_k = H_k x_k + v_k && \dots\dots\dots (2) \\
 Cov(w_k) &= Q_{kalman}, Cov(v_k) = R_{kalman} && \dots\dots\dots (3) \\
 x_k &= \hat{x}_k = \hat{U} && \dots\dots\dots (4) \\
 \hat{U} &= \hat{U} + K_k [U - H \cdot \hat{U}] && \dots\dots\dots (5) \\
 P_k &= (1 - K_k \cdot H_k) P_k + Q_{kalman} && \dots\dots\dots (6) \\
 K_k &= P_k \cdot H_k (H_k \cdot P_k \cdot H_k + R_{kalman})^{-1} && \dots\dots\dots (7)
 \end{aligned}$$

Algorithm 1: Kalman Filter.

Data: $\varphi=1.00, H=1.00, R_{kalman}=Cov(v_k)=0.33, Q_{kalman}=Cov(w_k)=15, P_0=0, K_0=0$

Result: \hat{U}

1. **while** true **do**
 2. **→** Obtain noisy U_i (at each time step)
 3. **→** GOTO KALMAN, obtain filtered U_i (at each time step)
 4. **end while**
 5. **function** KALMAN(U)
 6. **→** $K_k = \frac{P_k \cdot H_k}{H_k \cdot P_k \cdot H_k + R_{kalman}}$ (update Kalman Gain)
 7. **→** $\hat{U} = \hat{U} + K_k [U - H \cdot \hat{U}]$
 8. **→** $P_k = (1 - K_k \cdot H_k) P_k + Q_{kalman}$
 9. **→**
 10. **return** \hat{U} ;
-

Nomenclature for Butterworth Filter

$H(s)$	Butterworth transfer function	ζ	the damping ratio of the system
Ω_c	natural frequency	s	the plane
K	gain in passband	f	input frequency
f_h	cutoff frequency		

The Butterworth low pass filter, an anti-aliasing filter, provides maximum passband evenness. The higher the filter order, the longer the bandpass evenness. The general form of the transfer function model of a second-order Butterworth low pass filter can be described by equation (8) (Ingle & Proakis, 2012). The polar form of the low pass

filter is given by equation (9) [38]. The application of this filter can be seen in Algorithm 2.

$$H(s) = \frac{K}{s^2 + 2\zeta\Omega_c s + \Omega_c^2} \dots\dots\dots (8)$$

$$\left| \frac{\text{Output}}{\text{Input}} \right| = \frac{K}{\sqrt{1 + \left(\frac{f}{f_h}\right)^4}} \dots\dots\dots (9)$$

Algorithm 2: Butterworth Low Pass Filter.

Data: *nzeros* = 2, *npoles* = 2, *gain* = 8.524410156e+00, *xv*[*nzeros*+1], *yv*[*npoles*+1], *next_input_value*, *next_output_value*

Result: *next_input_value*

1. **while** true **do**
2. **→** Obtain noisy *next_input_value* (at each time step)
3. **→** GOTO *ButterworthLPF*, obtain filtered *next_input_value* (at each time step)
4. **end while**
5. **function** *ButterworthLPF*(*next_input_value*)
6. **→** *xv*[0] = *xv*[1]; *xv*[1] = *xv*[2];
7. **→** *xv*[2] = *next_input_value* / *gain*;
8. **→** *yv*[0] = *yv*[1]; *yv*[1] = *yv*[2];
9. **→** *yv*[2] = (*xv*[0] + *xv*[2]) + 2 * *xv*[1] + (-0.2947082939 * *yv*[0]) + (0.8254676161 * *yv*[1]);
10. **return**(*next_output_value* = *yv*[2]);

4.3.4 Simulation Design of Walk-Behind Tractor

Nomenclature for Tractor's Kinematics Model and Simulation			
ξ	the robot's pose in the local frame	ω_R	right wheel angular velocity (rad/s)
\dot{x}	x position	ω_L	left wheel angular velocity (rad/s)
\dot{y}	y position	v	linear velocity (m/s)
$\dot{\theta}$	the angle of tractor orientation	ω	angular velocity (rad/s)
R	the radius of the wheel	L	distance between the left and right wheels

This research carried out the kinematic mathematical modelling of the walk-behind hand tractor with the two-wheel differential drive. This kinematic model is used

for simulation using the pure pursuit and supervisory logic control algorithm with the waypoint concept in the Matlab/Simulink application.

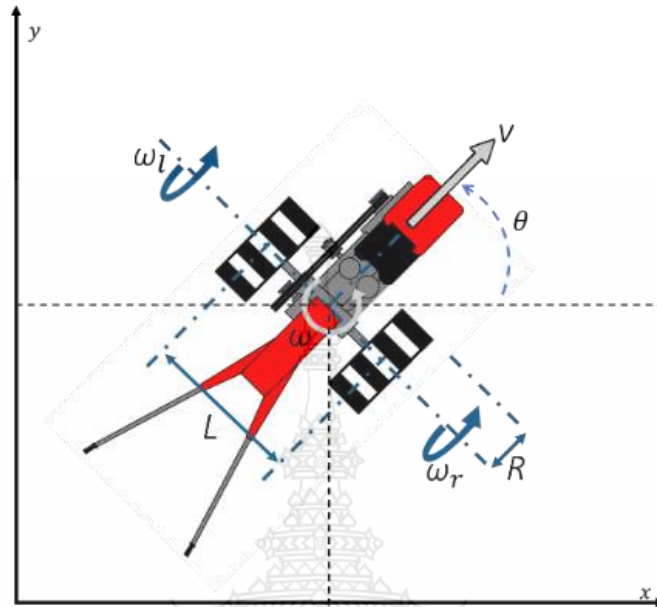


Figure 4.14. Differential Drive Kinematic Model

The kinematic model is based on the tractor types G 3000 and G 1000. Both have two independently driven wheels that can control velocity and angularity and are also known as differential-drive vehicles. A mathematical model of the tractor itself is needed to study and simulate the behaviour (operation of the walk-behind hand tractor); it is essential to do this before field testing is carried out. When modelling a tractor, the following two assumptions are made. The first assumption is that the tractor is moving at a constant speed. Second, the tractor wheels do not slip, and the surface for robot movement is flat. Figure 4.14 is a general description of the tractor model, which is used as the basis for the kinematics model of equations (10)-(13) with a differential drive type(Siegwart et al., 2011). A walk-behind hand tractor kinematics model can be written from this equation, a function of the left and right wheel angular velocity in a matrix (14). Equations (15) and (16) are forward kinematics calculations to relate the forward speed and angle of the tractor to the differential drive(Siegwart et al., 2011). At the same time, equations (17) and (18) are inverse kinematics to get the value of the angular velocity of the right and left wheels(Siegwart et al., 2011). Both Forward Kinematics and Inverse

Kinematics equations convert between body speed and wheel speed in visualizing tractor movement in Simulink.

$$\xi = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \dots\dots\dots (10)$$

$$\dot{x} = v \cos \theta \dots\dots\dots (11)$$

$$\dot{y} = v \sin \theta \dots\dots\dots (12)$$

$$\dot{\theta} = \omega \dots\dots\dots (13)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} R \cdot \cos \theta & R \cdot \cos \theta \\ R \cdot \sin \theta & R \cdot \sin \theta \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \dots\dots\dots (14)$$

$$v = \frac{R}{2} (\omega_R + \omega_L) \dots\dots\dots (15)$$

$$\omega = \frac{R}{L} (\omega_R - \omega_L) \dots\dots\dots (16)$$

$$\omega_L = \frac{1}{R} \left(v - \frac{\omega L}{2} \right) \dots\dots\dots (17)$$

$$\omega_R = \frac{1}{R} \left(v + \frac{\omega L}{2} \right) \dots\dots\dots (18)$$

4.4 TROLLS: Tractor Controlling System

This study proposes a novel controlling platform system (mechanical, electrical, and software) for walk-behind hand tractors with a low budget. The operator controls it via an android application using the Bluetooth connection. The prototype design was built and implemented on the G3000 tractor and tested on the farmland (Figure 4.23). Several revisions were made to obtain the final product's design from the field trial, later implemented and tested on the G1000 tractor. This final product was tested on the farmland, as shown in Figure 4.24. Modifications were made to the standard tractor by adding a novel mechanical and electrical driver unit design to be controlled remotely. In the first experiment (prototype design on the tractor G1000), the researcher used a stainless-steel wire rope and used four servo motors and two pulleys connected to the steering linkage (Figure 4.15). However, an evaluation of this design resulted in a reasonably high cost and inefficient use of power, so in the second experiment (the final product design on the G3000 tractor), only two servo motors were used, which were directly connected to the end of the clutch handle (Figure 4.16). Novel mechanical design modifications were applied to the tension handle (Figure 4.17) and throttle lever using a

screw system for linear movement. This movement system functions optimally on the tension handle in the first and second trials. However, the throttle lever mechanical design was not optimal in the first experiment (Figure 4.18), so it was replaced with a DC motor with an encoder (Figure 4.19). Another novel mechanical design is applied to the main bar by providing a tractor-weight balance hanger, shown in Figure 4.20. This weight balance hanger design works optimally in the first and second experiments.

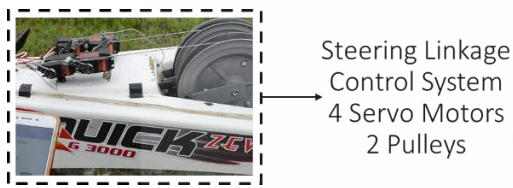


Figure 4.15. Modification on G3000

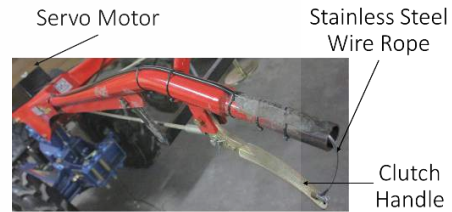


Figure 4.16. G1000 Modification

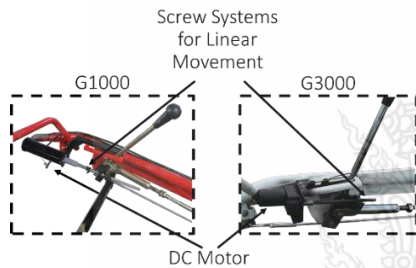


Figure 4.17. V-Belt Handle Bar Modification



Figure 4.18. G1000 Speed Modification



Figure 4.19. G3000 Speed Modification



Figure 4.20. Balancer for Tractor

Along with the mechanical design, the electrical driver unit functions as a slave, accepting commands and driving the actuator. The initial experiment included standalone or modular electronic components (Figure 20a). This driver unit design requires less cost-efficient; therefore, a single-board microcontroller board combines the functions of several modules into one (Figure 4.21). Android software is made as a master to send

commands to the driver unit. The software design was made portrait on the first try, but the operator deemed this design less than optimal (Figure 4.22). So based on these shortcomings, the software design was changed to landscape (Figure 4.13). The Bluetooth communication protocol is cost-effective and can span numerous agricultural fields in Indonesia, particularly the Bali region. This application has been accessible on the Google Play Store under TROLLS - Tractor Controlling System (STIKOM Bali developer). A cost and battery runtime comparison between the prototype and the final product was performed to validate this platform.



Figure 4.21. Prototype Version.



Figure 4.22. Final Product Version.

As shown in Table 4.1, the test results measured the cost-efficiency of the walk-behind hand tractor controlling system. A 21.74% cost efficiency is obtained by comparing the total cost of making the final product and the prototype. This cost efficiency showed that shifting the connection of stainless-steel wire rope from the steering linkage to the clutch handle could reduce the number of stepper motors. Also, combining several components into a single board microcontroller could save the cost spent compared to buying several components separately. This product is more affordable for Indonesian farmers in terms of cost when compared to previous studies (Guo et al., 2018; H. Wang & Noguchi, 2019b), with a lower level of precision. During the test, the accumulator battery with a voltage of 12V and a capacity of 7 Ah was used. In the prototype design, two batteries were arranged in parallel to fulfil the Current's need, and the final product also used two batteries. Based on the farmland field testing, the battery runtime's endurance could achieve more or less 12 hours. Battery utilization efficiency is 84.62 percent, as determined by the difference between the consumption of the final product and prototype batteries. Table 4.2 shows the final result of battery usage.



Figure 4.23. First Field Trial



Figure 4.24. Second Field Trial

Table 4. 1. The Tractor Controlling System Production Cost (Prototype & Final Product).

Component	Prototype (Qty)	Final Product (Qty)	Price	Prototype (\$)	Final Product (\$)
Servo Motor	4	2	14.5	58	29
DC Motor	2	2	20.6	41.2	41.2
Accumulator Battery	2	2	15	30	30
Arduino Uno	1	n/a	6	6	n/a
BTS7960 DC Driver	2	n/a	4.13	8.26	n/a
Bluetooth HC-05	1	1	3.44	3.44	3.44
Step-down	2	n/a	2.48	4.96	n/a
Single Board Microcontroller	n/a	1	7.6	n/a	7.6
Approximated Turning Machine Cost	1	1	35	35	35
Total Cost				186.86	146.24

Table 4. 2. Accumulator Battery Runtime Comparison (Prototype & Final Design).

Component	Prototype (Qty)	Final Product (Qty)	Current Each Component (A)	Prototype (A)	Final Product (A)
Servo Motor	4	2	1.9	7.6	3.8
DC Motor	2	2	1.4	2.8	2.8
Total Current (A)				10.4	6.6
Approximated Battery Run Time (Real-time Field Testing) (Hours)				6.5	12

The researcher also tracks the tractor's location and orientation, measuring cost efficiency and battery runtime. Data from GPS and Compass sensors (HMC5883L) were collected during the field test. The Kalman and Butterworth Low Pass Filter were then applied with visualizations, as shown in Figures 4.25 and 4.26. The two images show that the data generated from GPS and Compass is volatile; therefore, the filter can produce

stable data(X. Han et al., 2017; T. Wu & Hung, 2017). However, from these two images, the comparison results between the Kalman and Butterworth Low Pass Filters cannot be seen, so a root-mean-square error (RMSE) test is carried out on position and orientation predictions with different durations using equation (19)(Liu & Guo, 2021), where $M_n(X_n)$ is the actual measurement based on X_n , $M_n^p(x_n)$ is the prediction time, and N is the number of prediction times. From this test (Table 4.3), it can be concluded that the second-order Butterworth Low Pass Filter gets better results when compared to the Kalman Filter.

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (M_n^p(x_n) - M_n(X_n))^2} \dots\dots\dots (19)$$

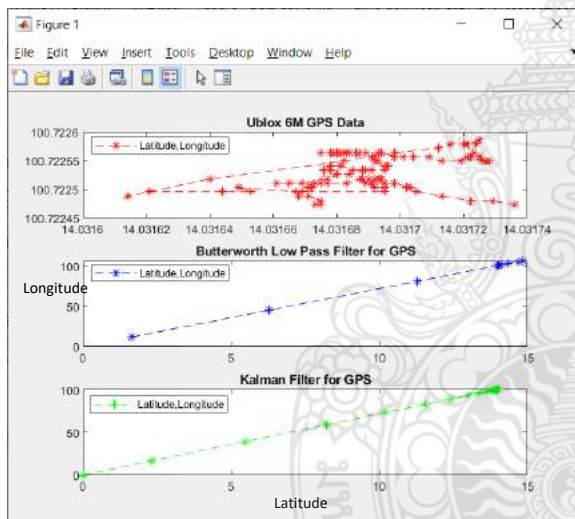


Figure 4.25. GPS Data Visualization

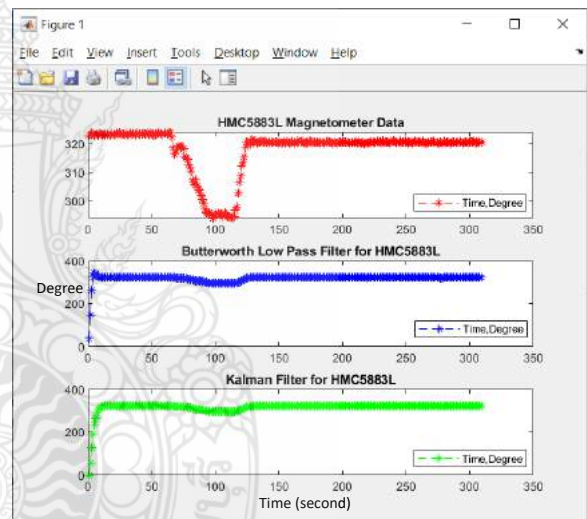


Figure 4.26. HMC5883L Data Visualization

Table 4.3. RMSE Comparison.

RMSE GPS (m)		RMSE HMC5883L (degree)	
Butterworth LPF	Kalman Filter	Butterworth LPF	Kalman Filter
0,0263888	0,1823068	0,8174105	1,0232198

Overall, the development process to create a two-wheeled hand tractor remote control system has been completed, but the automation process must be carried out. Therefore, a preliminary study was conducted using Simulink on kinematic model

mapping and tractor movement simulation. Equations (1)-(9) were then implemented into Simulink with two controller types. Figure 4.27 is a block diagram of the Pure Pursuit Control, while Figure 4.28 is a Supervisory Logic Control with details in Figure 4.29. The parameters and values used in this simulation can be seen in Table 4.4. The waypoint concept is the basis of this simulation, where the input location is (x, y) in the form of a 10×2 array and forms a back-and-forth path. Details of each Simulink block can be seen in Figures 4.30 -4.33.

Table 4.4. Simulation Parameters.

Parameters	Value
Desired Linear	
Velocity (v)	0.5 m/s
Maximum Angular	
Velocity (ω)	0.785 rad/s
Lookahead distance	0.35 m
L	1.6 m
R	0.4 m
Waypoints $[x_1, y_1; \dots; x_n, y_n]$	$[0,0; 0,5; 1,5; 1,1; 2,1; 2,5; 3,5; 3,1; 4,1; 4,5]$

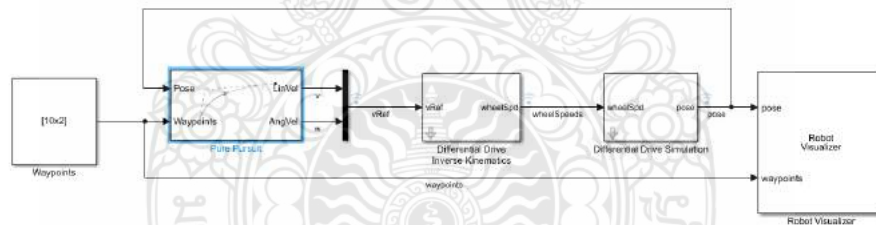


Figure 4.27. PPC Block Diagram

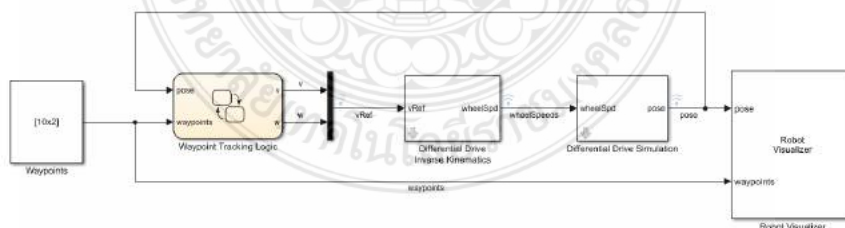


Figure 4.28. Supervisory Logic Control Block Diagram

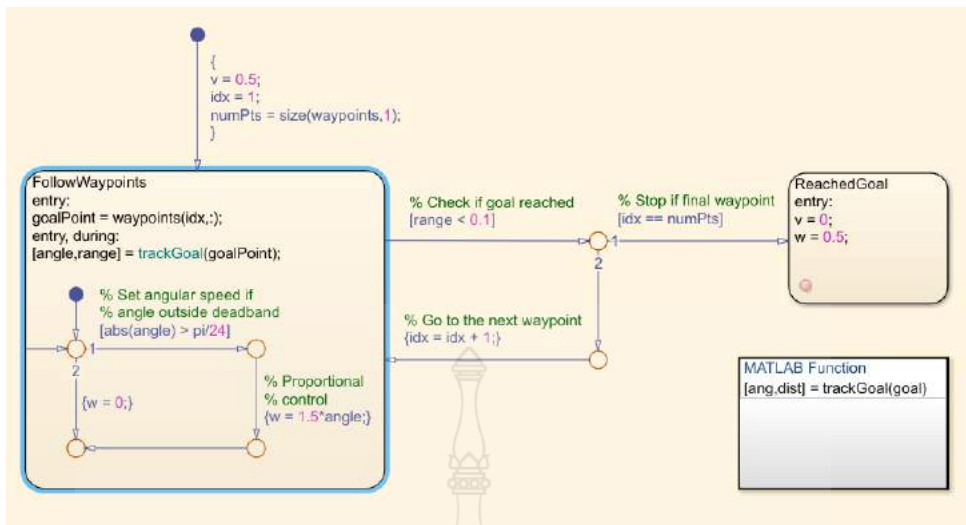


Figure 4.29. Detail of Waypoint System Simulation

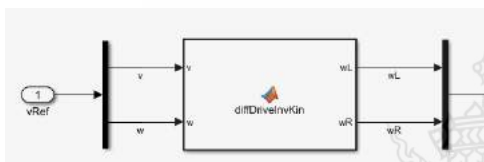


Figure 4.30. Differential Drive Block Detail

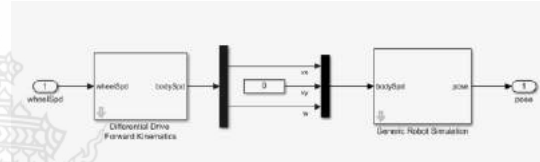


Figure 4.31. Differential Drive Block Detail

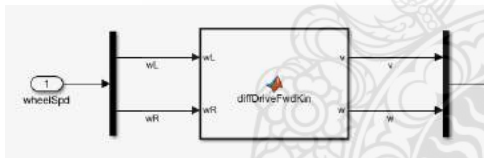


Figure 4.32. Differential Drive Block Detail

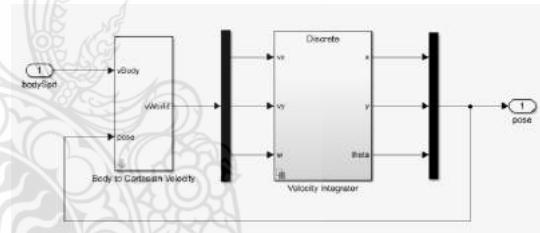


Figure 4.33. Differential Drive Block Detail

Figure 4.34 illustrates the simulation of tractor movement using Supervisory Logic Control, while Figure 4.35 simulates tractor movement using Pure Pursuit Control. Meanwhile, Figures 4.36 and 4.37 depict each controller's right and left wheel speeds. From the simulation results, it can be concluded that Supervisory Logic Control has a higher level of accuracy, reaching 10 out of 10 coordinates (100%), while Pure Pursuit Control has 6 out of 10 coordinates (60%).

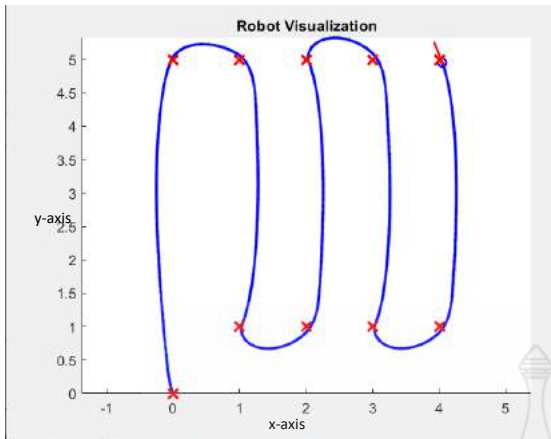


Figure 4.34. First Simulation Results

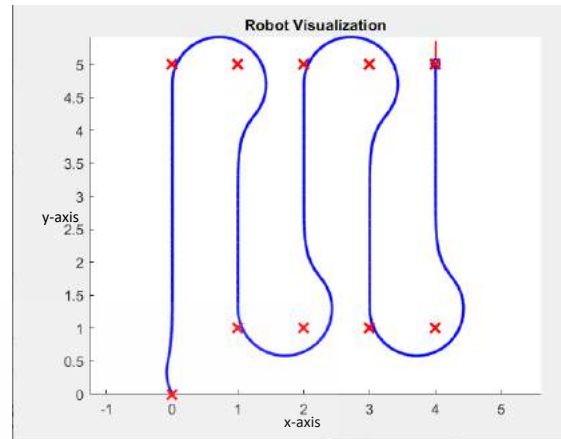


Figure 4.35. Second Simulation Results

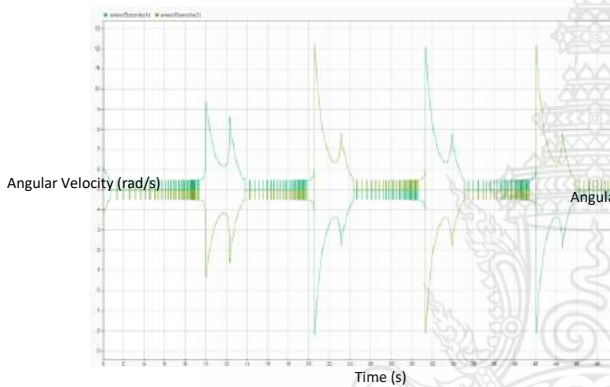


Figure 4.36. Wheels Speed Data for Supervisory Logic Control Simulation

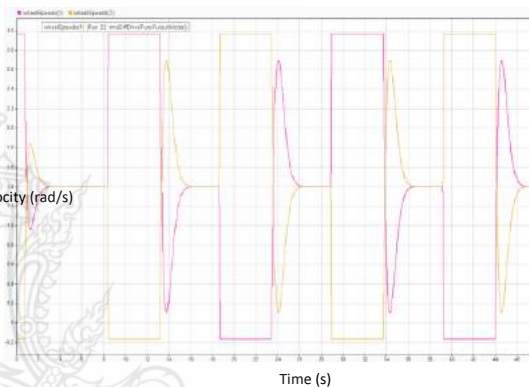


Figure 4.37. Wheels Speed Data for Pure Pursuit Control Simulation

4.5 Path Planning

This main research aims to compute the coverage path equation (22) to obtain an edge-vertex path with a Boustrophedon Cellular Decomposition pattern and test it on the QUICK G-1000 walk-behind tractor. The goal is to determine the edge-vertex routes within the polygon, the start-finish points, and the distance between tillage lines.

$$\tau = \{ps, p0 \dots pn, pf\} \quad (22)$$

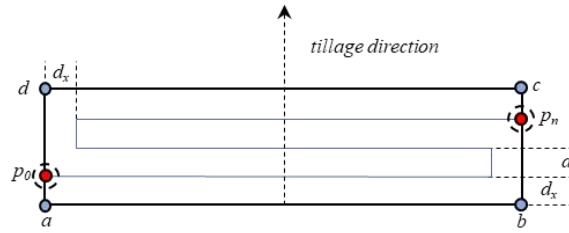


Figure 4.38. EVP Representation.

Figure 4.38 shows a path that forms an alternate direction when the tillage line generated is parallel to one side of the ROI. An Edge-Vertex Path (EVP) is formed inside polygon Q . EVP is formed by the intersection of waypoints and the $L_{tillage}$ line with polygon Q . In algorithm 1, the inputs are the polygon Q , the initial vertex b , the antipodal vertex c and d , and the distance between the tillage line dx . dx is the first user input determined by the size of the Puddler rather than the type. At the beginning process, the $L_{tillage}$ line is parallel to the sides (a, b) that have been displaced perpendicularly toward the c and d directions. After intersecting with polygon Q , $L_{tillage}$ subtracts or adds dx to create p_1 and p_2 . The next step is to combine the points into the path and connect them to the CalculateConnect function, which provides a perpendicular boundary between the two points. If the footprint intersects the polygon line, this method will be repeated and shifted to points c or d . This algorithm returns the path $\rho = \{p_0, \dots, p_n\}$, at the final step. EVP always begins from vertex a and sweeps towards vertex $b, c,$ and d .

Algorithm 1: Polygon ROI edge-vertex identification (GetPath(a, b, c, d)).
Calculating the BFP path begins with $(\rho = \{p_0, \dots, p_n\})$ for the polygon ($Q = V, E$). The Tractor begins its movement at vertex a and sweeps in the direction of c and d .

Data: Q, dx, a, b, c, d
Result: ρ
 $L_{tillage} \leftarrow \text{CreateLine}(a, b);$
 $L_{tillage} \leftarrow \text{Offset}(L_{tillage}, dx);$
 $\rho \leftarrow \emptyset;$
while $\text{Intersects}(C(L_{tillage}); Q)$ **do**
 if $(\text{IntersectEdgesLeft})$ **then**
 $ip_1; ip_2 \leftarrow \text{IntersectEdges}(L_{tillage}; E) + dx;$
 else if $(\text{IntersectEdgesRight})$ **then**
 $ip_1; ip_2 \leftarrow \text{IntersectEdges}(L_{tillage}; E) - dx;$
 end if
 $\rho \leftarrow \text{CalculateConnect}(\rho; ip_1; ip_2);$
 $L_{tillage} \leftarrow \text{Offset}(L_{tillage}, dx);$
end while
return $\rho;$

The result of implementing Algorithm 1 is a Laravel-based website platform that uses Google Maps for satellite map sources. The user can add a region of interest to the accessible map and specify the Interval Distance in meters and the starting and ending points. After determining the initial inputs, the final step is to click the Generate Paths button to generate the path automatically. A scenario with varying intervals from the same ROI, start point, and finish point is given to validate this platform. This platform is evaluated using 1, 1.5, 2, and 2.5 meters interval distance values. Figure 4.39 shows that the platform successfully auto-generating coordinates in four different scenarios. Furthermore, the coordinates generated by the path planning platform were validated by field tests to support the autonomous walk behind the Quick G1000 tractor.

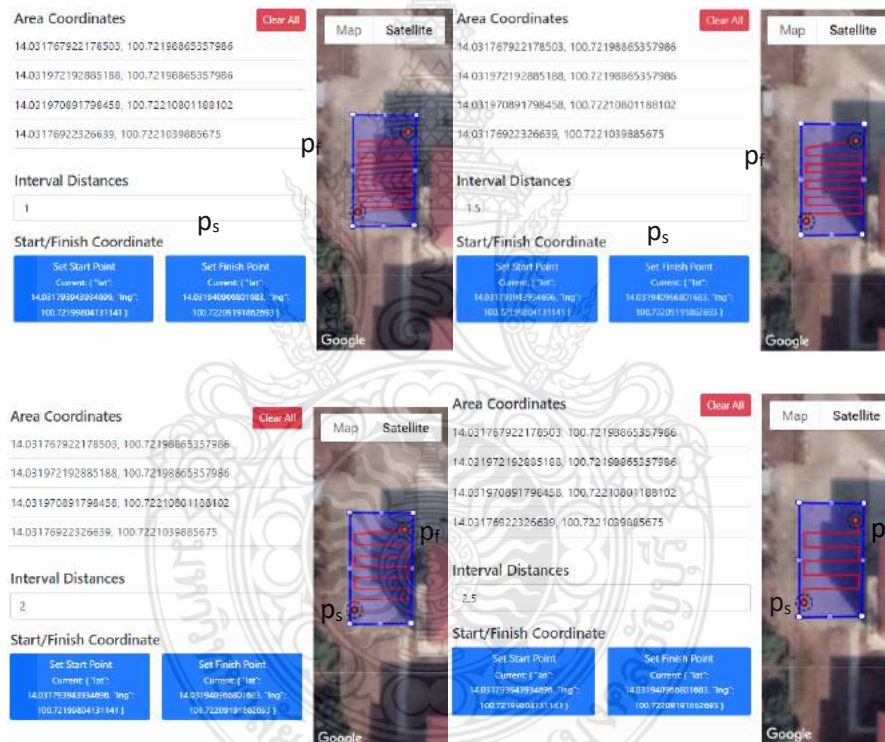


Figure 4.39. Path Planning Platform

As a validation scenario, an Embedded System Platform was installed on the Tractor. Figure 4.40 shows the Embedded System Platform box, which contains a controller and several motors that control the clutch handle and propel the tractor wheels autonomously via the waypoint method (coordinates from Path Planning Platform) at a constant and predetermined speed. In the controller, several algorithms are implanted to

read heading distance, error, and angle so that the Tractor can move autonomously, including Algorithms 2, 3, and 4.



Figure 4.40. Quick G1000 Tractor Box Controller Installation

Algorithm2: Compass Reading

Data: normalize, normalize_YAxis, normalize_XAxis, heading, declinationAngle, headingDegrees

Result: headingDegrees

1. normalize \leftarrow ReadCompassNormalize();
 2. heading = atan2(normalize_YAxis, normalize_XAxis);
 3. declinationAngle = (\angle + (min/ 60)) / (180/ π);
 4. heading -= declinationAngle;
 5. **if** (heading less than 0) **then**
 6. heading += 2 * π ;
 7. **else if** (heading more than 2 * π) **then**
 8. heading -= 2 * π ;
 9. **end if**
 10. headingDegrees \leftarrow ConvertToDegrees(heading);
 11. return headingDegrees;
-

Algorithm 2 is the compass sensor reading. The reading results are normalized to vector form and then used as input for the heading computation (atan2(normalize YAxis, normalize XAxis)). To compensate for the compass's divergence from the Earth's actual north pole, the declination angle (δ), which may be found at <https://www.magnetic-declination.com/> (Magnetic-Declination.com, 2022), must be calculated. The declination angle for Gianyar, Bali- Indonesia, is 0° 43' EAST (POSITIVE). There are two conditions

in which the heading calculation results should be corrected: larger than $2 * \pi$ and fewer than 0 degrees. After that, the result is translated into degrees.

Algorithm 3: Calculate Distance to Target: distance from the current location to the target waypoint

Data: deltaLongitude, currentLong, currentLat, targetLong, targetLat

Result: distanceToTarget

1. $\Delta\text{Longitude} \leftarrow \text{radians}(\text{currentLong} - \text{targetLong});$
 2. $\text{lat1} \leftarrow \text{radians}(\text{currentLat});$
 3. $\text{lat2} \leftarrow \text{radians}(\text{targetLat});$
 4. $\Delta\text{Longitude} = \text{sq}((\cos(\text{lat1}) * \sin(\text{lat2})) - (\sin(\text{lat1}) * \cos(\text{lat2}) * \cos(\Delta\text{Longitude})));$
 5. $\Delta\text{Longitude} += \text{sq}(\cos(\text{lat2}) * \cos(\Delta\text{Longitude}));$
 6. $\Delta\text{Longitude} = \text{sqrt}(\text{deltaLongitude});$
 7. $\text{denom} = (\sin(\text{lat1}) * \sin(\text{lat2})) + (\cos(\text{lat1}) * \cos(\text{lat2}) * \cos(\Delta\text{Longitude}));$
 8. $\Delta\text{Longitude} = \text{atan2}(\Delta\text{Longitude}, \text{denom});$
 9. $\text{distanceToTarget} = \Delta\text{Longitude} * 6372795;$
 10. return distanceToTarget;
-

Algorithm 3 is a distance computation between the robot's current location and the target waypoint. The GPS sensor value represents the current location's degree of longitude and latitude, and the target waypoint must be translated to radians. Following that, the Haversine formula is used to calculate the great-circle distance.

Algorithm 4: Calculate robot turning to get to the waypoint target

Data: headingError, targetHeading, currentHeading, errorOutput, headingTolerance, motorMovement

Result: motorMovement

1. $\text{headingError} = \text{targetHeading} - \text{currentHeading};$
 2. **if** (headingError less than -180) **then**
 3. $\text{headingError} += 360;$
 4. **else if** (headingError more than 180) **then**
 5. $\text{headingError} -= 360;$
 6. **end if**
 7. $\text{errorOutput} \leftarrow \text{calcErrorOutput}(\text{headingError})$
 8. $\text{motorMovement} \leftarrow \text{calcMotorMovement}(\text{errorOutput})$
 9. return motorMovement;
-

Algorithm 4 is a heading error computation when the target heading value differs from the current heading value. The heading error calculation findings will be utilized as input for the robot's movement, allowing it to operate autonomously. The direction of the Tractor's heading is depicted in Figure 4.41. Two restrictions limit the results of the Tractor heading angle calculation.

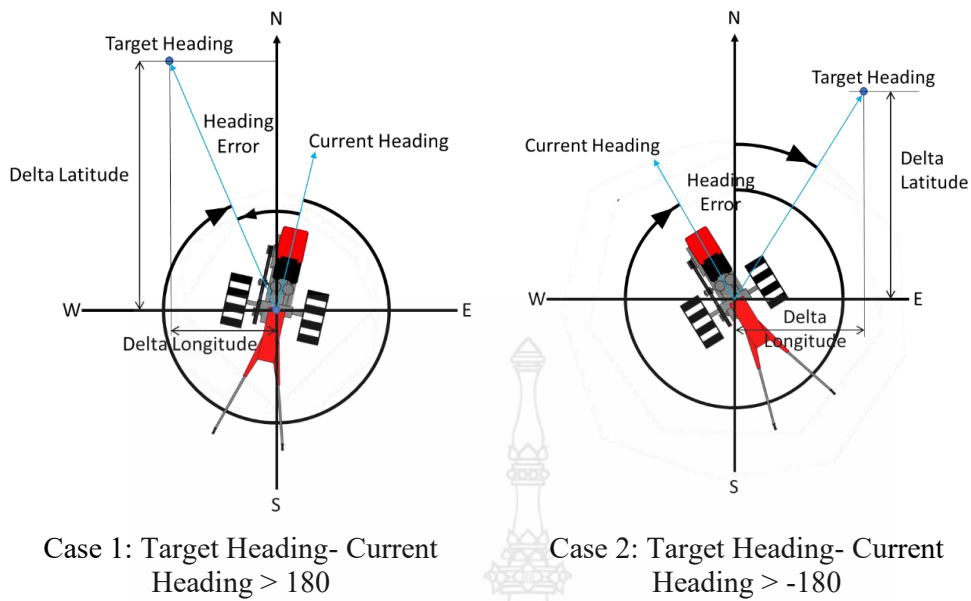


Figure 4.41. Error handling for Tractor Heading Angle

Field tests were conducted and compared to simulation results to validate the Path Planning and Embedded System Platform. In this test, one scenario with the following characteristics was created:

1. Figure 4.42 depicts a waypoint's coordinates with ROI, start-finish points, and four meters of interval distance. The value of four meters is utilized since the accuracy of the employed GPS is insufficient to overcome interval values of less than four meters.
2. Field test results are compared with the tractor model simulation using the MatLab robotic toolbox with a differential drive kinematic model to determine the estimated plowing time.
3. The simulation parameters used to represent the G1000 Quick tractor are shown in Table 4.5.
4. The ROI and the waypoint coordinates used in the simulation are the same as the size generated from the Path Planning Platform.

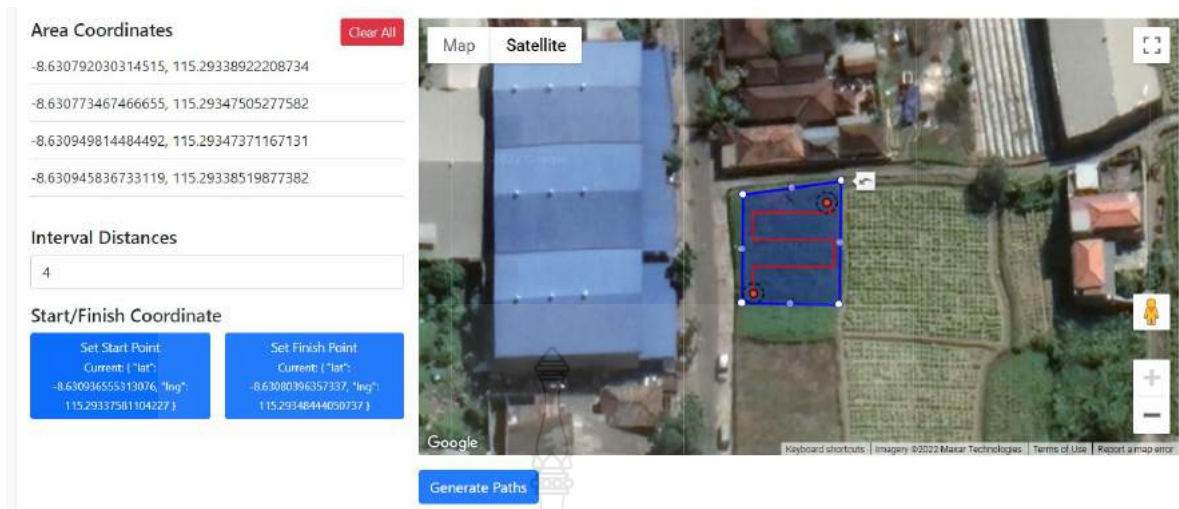


Figure 4.42. Scenario for Path Planning and Waypoint Testing

Table 4.5. Parameters for Matlab Simulation

Parameter	Unit	Value
Linear Velocity	m/s	0.56
Wheel Radius	rad/s	$\pi/4$
Distance between wheels	m	1.8

Figure 4.43 shows the Simulink block diagram for the simulation. The waypoint block is an input containing an array of coordinates. The experimental scenario of the Waypoint block has an array measuring 8 x 2. Based on the simulation results depicted in Figure 4.44, it is evident that the Tractor follows the given coordinates of the path. This simulation shows that the estimated tractor movement time is 84 seconds. The simulation results are then compared with the tractor movement time in-field trials, as shown in Table 3.

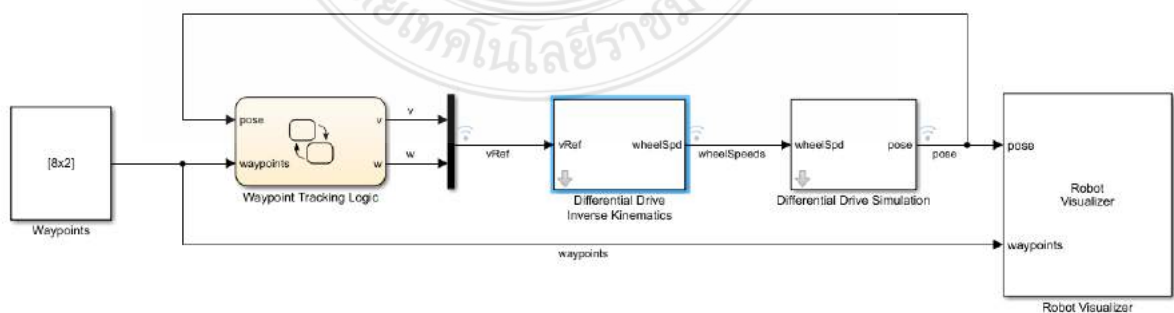


Figure 4.43. Simulation Block Diagram

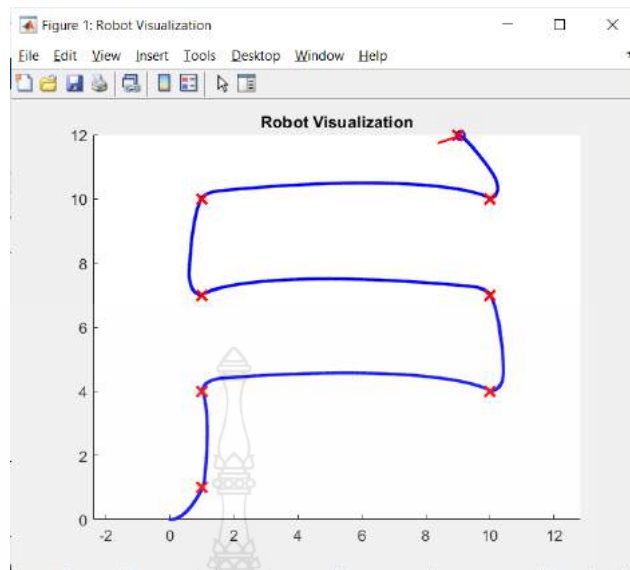


Figure 4.44. Simulation Results.



Figure 4.45. Tractor Movement for Field Trial.

Field trials were carried out in one of the rice fields in the Ketewel area, Gianyar Regency, Bali-Indonesia. The researcher used the same region of interest, starting point, finish point, and four meters distance interval. Each point is recorded using an IoT (Internet of Things) based logger module installed on the Tractor. The data is stored in the database and processed using a text editor into .gpx format, then visualized using the Google Earth application in Figure 4.45. The white line and red dots represent the robot's target path and waypoints, while the yellow dots represent the tractor path and the recording results of the logger module. This experiment demonstrates how a robot can

calculate tractor orientation and pass waypoints. As seen in Figure 4.45, the highest error distance between the target waypoint and the tractor position based on the GPS and Kalman Filter is 2.61 meters. The researcher also recorded the time data of plowing carried out in simulations and field tests with an area of interest of 302.65 m², and the results are shown in Table 4.6. The time required to generate the path planning coordinates is 685 milliseconds, while the time between simulation and trial is different. There is a difference of a few seconds where the simulation conditions are faster than the actual field trial.

Table 4.6. Comparison of plowing process time between simulation results and field trials

Polygonal ROI	Processing time	Simulation tillage time	Tillage time
Experiment 1	685 milliseconds	84 second	132 second

4.6 RIFIS: Rice Field Sidewalk Detection Using Machine Learning

4.6.1 Rice Field Sidewalk Dataset

A new dataset, RIFIS, is suggested as an alternative to earlier datasets (Kiratiratanapruk et al., 2020; Nguyen et al., 2021; Shao et al., 2021; H. Wang et al., 2021; Yakkundimath et al., 2022; Yang et al., 2021) that concentrated more on rice plant diseases. The RIFIS dataset presented in this work consisted of 16 videos with a size of 48.7 GB and 970 high-definition RGB images (1920 × 1080 pixels) and their annotations. Since the acquired raw material was a 1920 × 1080 pixel high-definition video, it was possible to extract several image sequences from a single video. By using this method, 24 images were recovered from each second of the raw video. The extracted images were named by concatenating the raw video source name and a postfix value that specified the order in which the images were extracted in the order in which they were made. For example, a raw video named “GH010327.MP4” (Figure 4.46a) was extracted into several image sequences starting at “GH010327_010000.PNG”. After that, several images were selected to be annotated and were given a name starting from “Sequence 010000.JPG” (Figure 4.46b).

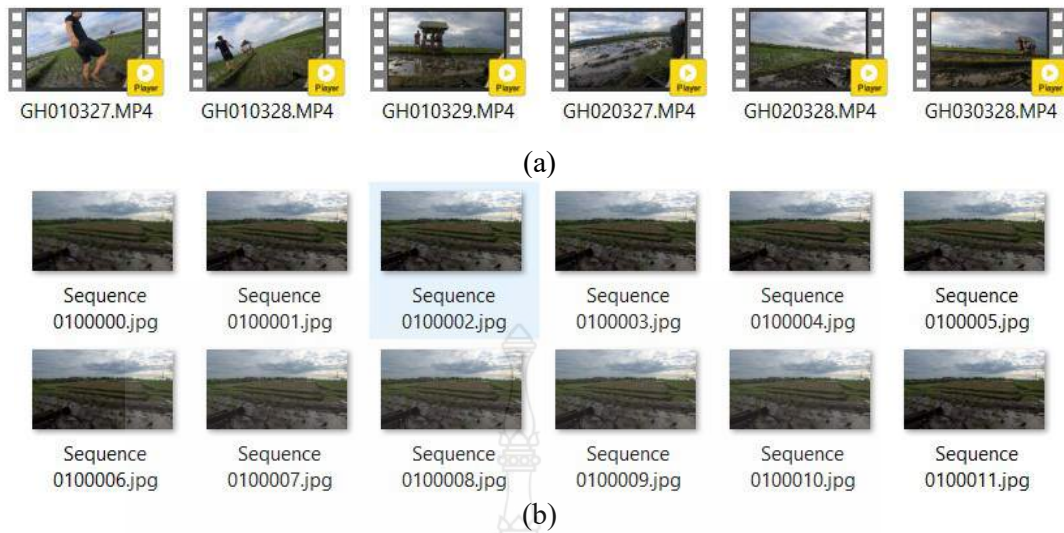


Figure 4.46. Dataset Collection

Recognizing the surrounding environment was one of the requirements so that the tractor could recognize the inside and outside areas of the rice field. The easiest way to divide these two conditions was to detect the RIFIS. Based on the collected video dataset, observations were made on the environmental conditions of the rice fields to obtain several features that could be used. The dataset had 19 unique features. Combining day circumstances, weather and ambient factors, paddy fields, partial occlusion, foreground objects, and backdrops provided difficulties for the RIFIS detection algorithm. These 19 characteristics are categorized in Table 4.7. The data collection was only carried out in the afternoon due to limited available funds, so land leases, cameras, tractors, operators, etc., had limitations.

Table 4.7. RIFIS dataset features.

Day Condition	Weather Condition	Rice Field State	Environmental Condition	Occlusion	Presence of O
1. Afternoon;	2. Partially Cloudy;	3. Partially Covered by Grass;	6. Mild to Strong Glare;	9. Partial Occlusion by Grass;	13. Grass;
		4. Watery;	7. Variation in Rice Field Surface Color;	10. Partial Occlusion by Humans;	14. Irrigation Channel;
		5. Partially Ploughed;	8. Not Smooth Color Transition Between Sidewalk and Rice Field Area;	11. Partial Occlusion by Tractor Wheel;	15. Humans;
				12. Partial Occlusion by Small Irrigation Channel;	16. Small Huts
					17. Houses;
					18. Sky (Clouds);
					19. Trees.

As discussed previously, the RIFIS dataset contained images comprising 19 features. In Figure 4.47, the researcher presented several examples of a RIFIS showing a combination of features, such as different levels of illumination (Figure 4.47a); strong glare and paddy field conditions (Figure 4.47b); and small irrigation channels (Figure 4.47c), partial human occlusion (Figure 4.47d,e), cloudy afternoons and partially ploughed rice fields that make the sky reflected in puddles and detected as clouds (Figure 4.47f); fog; foreground objects and city skylines (Figure 4.47g); huts; pools of water and glare (Figure 4.47h); and partial occlusion by grass (Figure 4.47i).

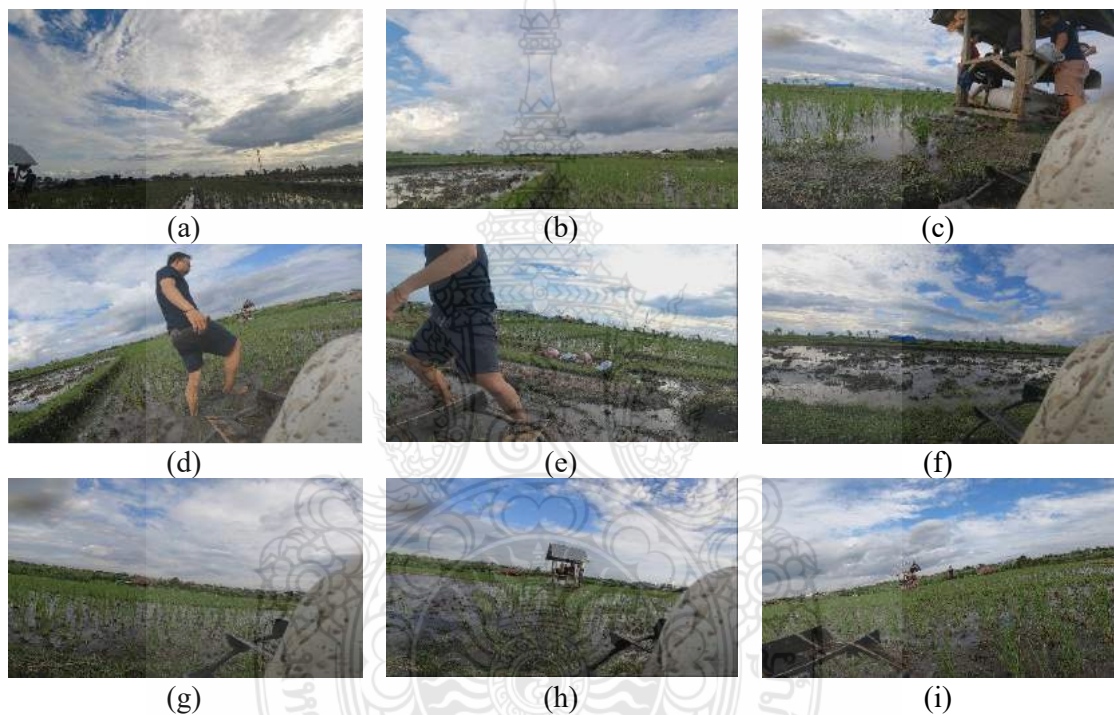


Figure 4.47. Features Variation of RIFIS Dataset.

The final collection of images was manually annotated using the website-based tool makesense.ai. Annotations had two purposes: first, to identify the RIFIS, and second, as a benchmark for evaluating the RIFIS detection algorithm's performance. The researcher manually drew and labeled sidewalk area polygons for each image. The annotation software outputted a JSON file from which the RIFIS polygon points and recommended ground truth (GT) values were extracted and calculated. Figure 4.48 depicts the manual annotation procedure using the software makesense.ai

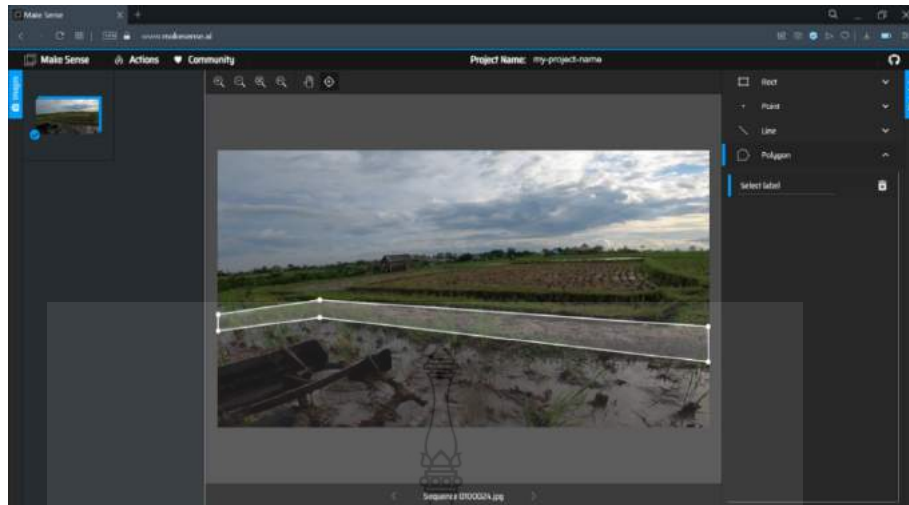


Figure 4.48. Images Annotation.

The ground truth (GT) value identifies the real position of the object of interest within an image. A GT schema depicted in Figure 4 was developed to obtain the rice field sidewalk GT values. There were three GT schemas, namely the RIFIS area, which formed triangular, square, and concave polygons. The GT schema presented in Figure 4.49a consisted of eight points forming the RIFIS polygon (sidewalk) area, namely P1 (x1,y1), P2 (x2,y2), P3 (x3,y3), P4 (x4,y4), P5 (x5,y5), P6 (x6,y6), P7 (x7,y7), and P8 (x8,y8). Meanwhile, Figure 4.49b only had six points, Figure 4.49c had four points, and Figure 4.49d had three points. The sidewalk area separated the rice field and the outside area.

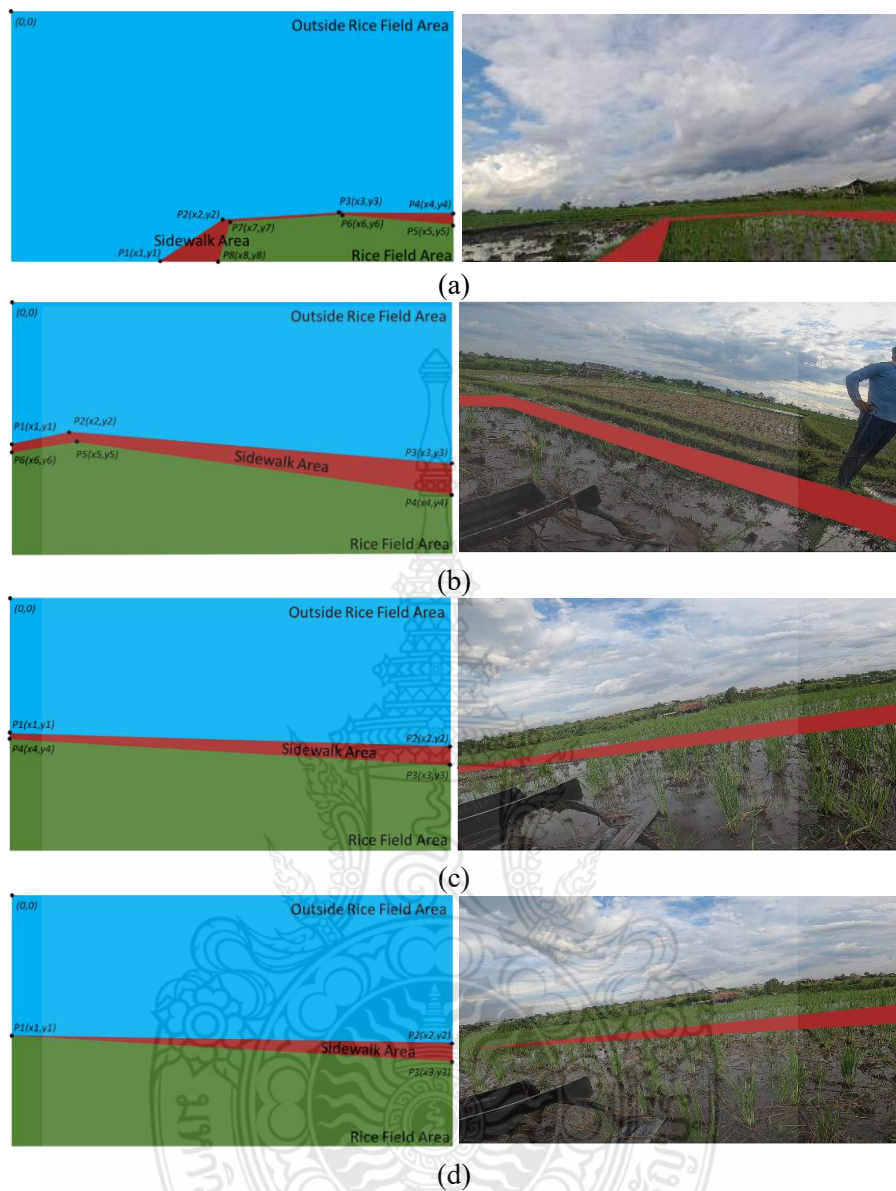


Figure 4.49. Ground truth labeling.

Table 4.8 shows the structure of the RIFIS JSON file as the annotation results, containing two main parts (image and annotation arrays). In the JSON file’s annotations field, “id” represented a single image object, “iscrowd” indicated whether the segmentation pertained to a single object or a group/cluster of objects, and “category_id” corresponded to a unique category listed in the categories section. There were two distinct types of labeling: (1) annotation of polygonal segmentation and (2) annotation of the rectangular bounding box. Figure 4.50 represents examples of image labeling from the RIFIS dataset. As shown in Figure 4.50a, the polygonal segmentation annotation included

a float array segmentation list of vertices (x, y pixel positions). Figure 4.50b shows the x and y coordinates of the upper left and lower right corner arrays for the rectangular bounding box. “Area” represented the area of the bounding box in each image. Object detection was typically described as detecting a rectangular bounding box and a class label for each object of interest in an image. In instances of segmentation, a pixel-by-pixel segmentation was created for each occurrence. The target object was the rice field sidewalk, which was unsuitable for object detection, segmentation, or depth perception tasks, all of which are required by other systems, such as autonomous or assistance systems. The proposed dataset included a variety of annotations for the sidewalk environment. To the best of the researcher's knowledge, this was the first large-scale sidewalk dataset that included annotations for instance-level objects (bounding box and polygon segmentation) and ground-truth depth.

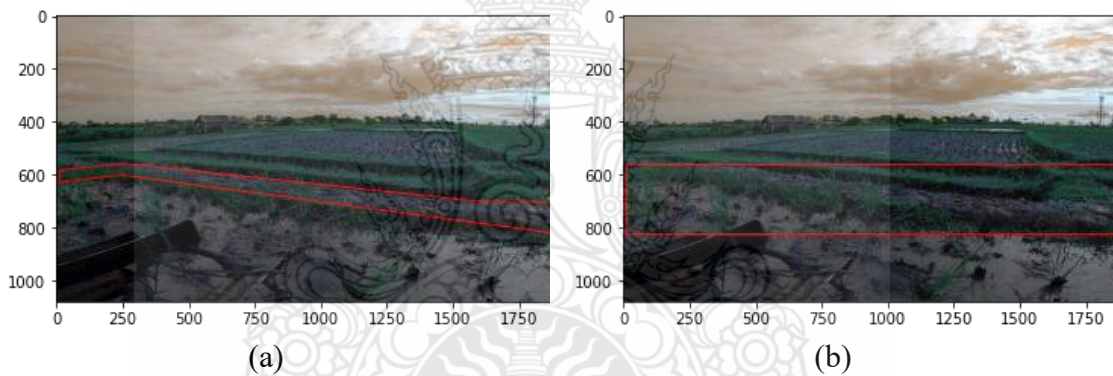


Figure 4.50. Dataset Labeling.

Table 4.8. The structure of the RIFIS JSON file.

Images []		Annotations []	
id	integer	id	integer
width	integer	iscrowd	Boolean
height	integer	image_id	integer
file_name	string	category_id	integer
		segmentation	float []
		bbox	float []
		area	float

4.6.2. Tractor Location and Orientation Dataset

The data obtained through sensors mounted on the tractor were then stored in a database using internet of things technology with the MQTT protocol. The stored data had an index ('id') as the primary key, followed by data on the date that the data were recorded, in the format "YYY-MM-DD HH:MM:SS". Tractor orientation data were obtained from 'yaw', 'pitch', and 'roll' values from the gyroscope sensor; 'x', 'y', 'z' values from the accelerometer sensor; and 'a' (azimuth) values from the compass sensor. The location data of the tractor were recorded using a GPS sensor where the coordinates (longitude and latitude) were the primary reference. The data recorded on the MQTT server were then exported into .sql form to be processed on the local server. The data were cleaned of noise from GPS reading errors, which were then exported into .xlsx to be more easily analyzed and used. After cleaning, there were a total of 3728 data. Figure 4.51a shows the electrical component implementation of the data logger; meanwhile, Figure 4.51b shows the final packaging of the data logger; Researcher used an external antenna to enhance the ESP32 TTGO T-Call signal. The description for this hardware logger can be seen in Table 4.9.



(a) (b)
Figure 4.51. Data Logger for Location and Orientation.

Table 4.9. Data description captured by the logger device.

Device	Data Variable	Example Value	Unit
ESP32 TTGO T-Call	Date-Time	2021-12-21 10:18:06	yyyy-mm-dd hh:mm:ss
Gyroscope	Yaw	-36.219238	deg/s
	Pitch	2.912616	deg/s
	Roll	-13.965352	deg/s
Accelerometer	X	-39	m/s ²
	Y	-87	m/s ²
	Z	266	m/s ²
Magnetometer	Azimuth	284	deg
GPS	Longitude	-8.632576	deg
	Latitude	115.144852	deg

4.6.3. Foldering Structure

The hierarchical folder structure of the RIFIS dataset is shown below:

- RIFIS
 - Images
 - dataset
 - annotations.json
 - LocationOrientation
 - Location-orientation.xlsx
 - Videos
 - FrontCamera
 - LeftCamera
 - RightCamera

4.6.4. Dataset Acquisition Methods

4.6.4.1. Location and Source of Collection

Rice field sidewalks are the boundaries of rice fields from one plot to another, usually measuring 30 cm or more. In addition to functioning as a barrier to rice fields, docks, or rice field sidewalks, there are also many functions and uses for farmers. It can reach a width of 1 m or more in certain areas. In some regions, farmers can use rice field

sidewalks as access roads for farming by farmers to transport crops and fertilizers during the fertilization period for rice plants. Routine maintenance of rice field sidewalks is carried out by cleaning them from weeds and sweeping or spraying herbicides. In addition to treating weeds, the barriers must be added with mud and trimmed to keep the rice fields from collapsing.

The rice field is one of the sub-agricultures that provide staple food. Generally, rice fields are used for rice cultivation. However, several stages must be carried out before carrying out the rice planting process, including the process of ploughing the fields. Ploughing is the activity of cultivating the land by turning the soil so that the soil becomes smooth and easy to plant in. The process of ploughing rice fields consists of two processes, namely the process of loosening the soil and the process of refining the soil. The process of loosening the soil currently still uses a tractor. Many tractors are available today, both two- and four-wheeled. In general, the movement of the tractor when carrying out the process of ploughing the fields forms a spiral pattern, as in Figure 4.52, which was the scenario for collecting RIFIS dataset images in this study. The tractor moved from the start to the finish points with the RIFIS as a barrier. The path that was traversed is called the footprint. It can be seen in the top-view image (using a drone) of the RIFIS image data collection scenario in Figure 4.53.

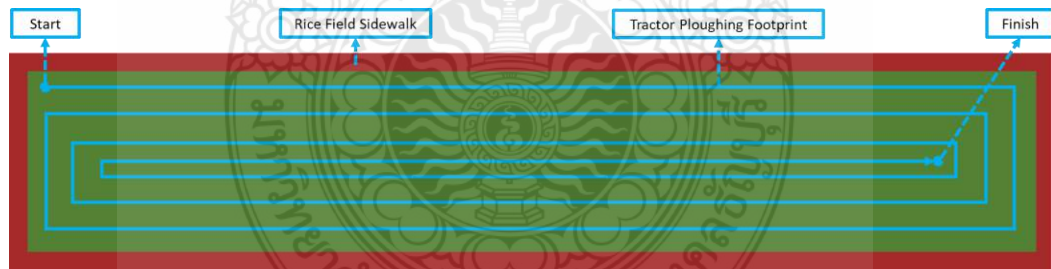


Figure 4.52. Dataset Collection Scenario.



Figure 4.53. Dataset Collection Process.

The selection of the observation location was the main factor that affected the dynamics of the features in the RIFIS image. For example, the observation location was in a rice field area where the neighboring rice fields were in a condition where some had been ploughed, and some had not. The condition of the cultivated rice fields had similar characteristics to RIFISs, producing dynamic conditions according to reality. Considering this fact, two locations with different longitude and latitude coordinates in Bali, Indonesia were selected for the data collection experiment (Figure 4.54a). More details about these locations are provided in Figure 4.54b and Table 4.10.

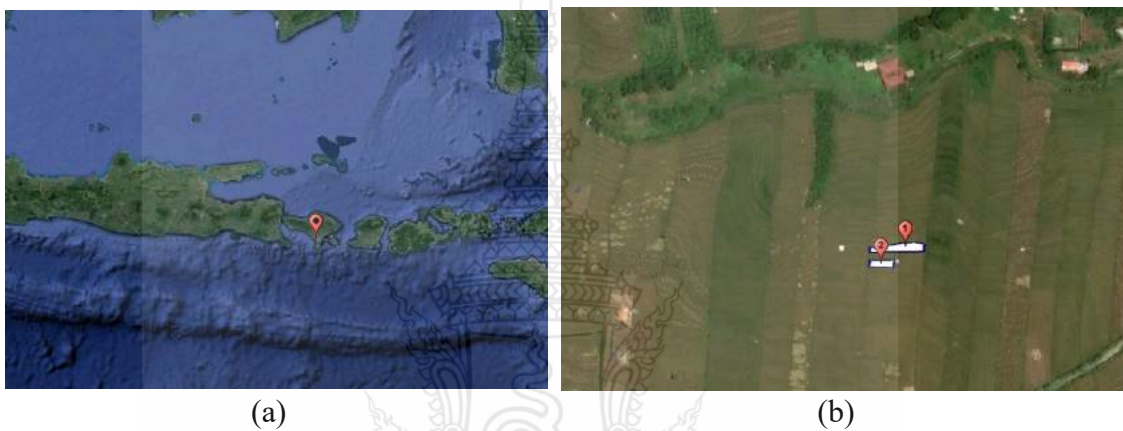


Figure 4.54. Data collection site.

Table 4.10. Details of geographical locations for data collection.

Nature of Location	Location Name	Geographical Coordinates
Rice Field 1	Uma Desa Canggu	-8.632394°; 115.144956°
Rice Field 2	Uma Desa Canggu	-8.632368°; 115.144836°

4.6.4.2. Camera and Recording Support

To capture RIFIS images in the process of ploughing fields, the researcher used a GoPro Hero 9 camera. The camera settings used were auto (zoom 1.0×) with an image resolution of 1920 × 1080 and a 60 frames per second (fps) frame rate. The lens setting used in this research was wide, with an ISO in the minimum value range of 100 to a maximum of 6400. The three cameras were mounted on the top of the front of the tractor. The first camera faced the right diagonal, the second camera faced forward, and the third

camera faced the left diagonal. The camera placement on the tractor can be seen in Figure 4.55. Figure 4.56, it can be seen the results of the captures of the three cameras. The researcher recorded all video sequences of the dataset by placing the camera on top of a tractor, ploughing a field with three different shots (diagonal left, front, and right). Three sets of footage were taken with an above-shot camera angle relative to the RIFIS.

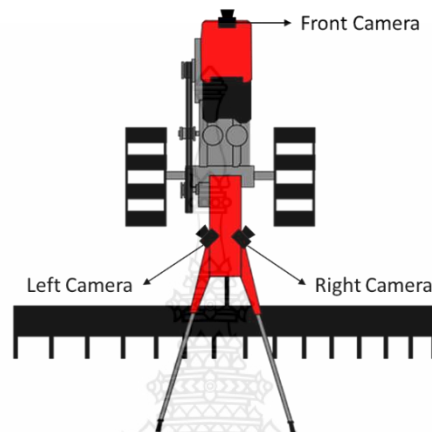


Figure 4.55. Camera Position on the Tractor.

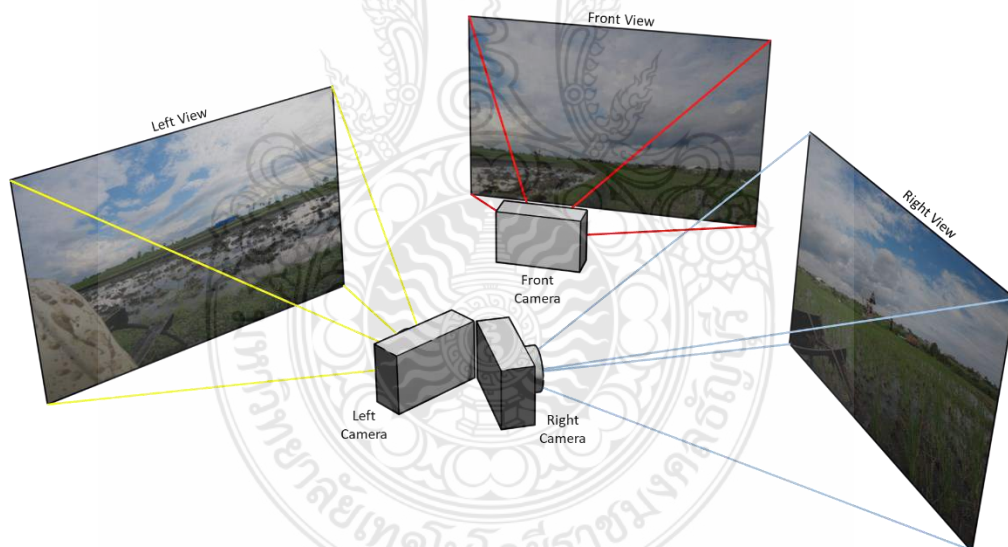


Figure 4.56. Illustration of Camera Results.

4.6.4.3. GPS, MPU, and Compass

The location and orientation data of the tractor were recorded to view and analyze the movement patterns as supplementary data. A set of hardware was embedded in the tractor to achieve this goal. IoT technology with the MQTT protocol was used as a liaison between the hardware and the server. ESP32 Lilygo T-Call 1.4 is a microcontroller

equipped with a SIM800L module. This allowed it to communicate over the internet without needing a separate access point module. Three sensors were used to obtain tractor movement data, namely the U-Blox Neo-6M as a GPS module to obtain location data for longitude and latitude coordinates. To obtain tractor orientation data, an MPU6050 GY-521 was used as the gyroscope–accelerometer sensor and a GY-271 as the compass sensor. For this study, a 1-s interval was used to record all the location and orientation data of the tractor. Figure 4.57 illustrates the wiring in the three sensors and microcontroller diagrams during the data collection experiment.

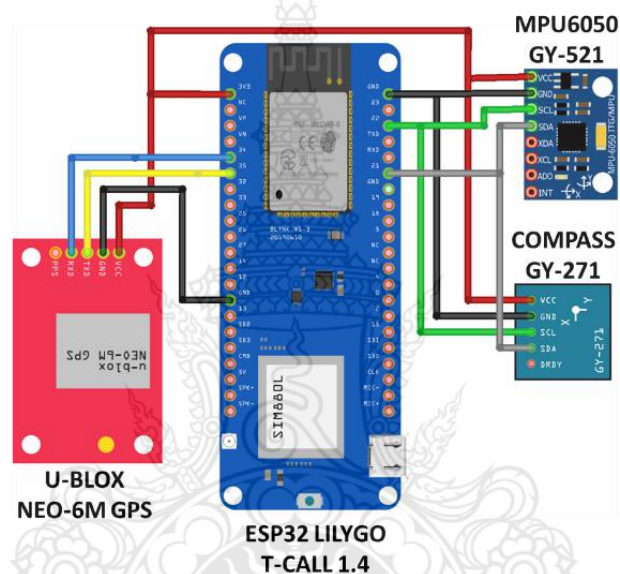


Figure 4.57. Wiring Diagram of GPS – Compass- MPU6050 Sensor.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

In conclusion, this study has presented a two-wheeled walk-behind tractor controlling system. The design and development process included mechanical modifications to the tractor, testing the functionality of the electronic component circuit, and integrating the electronic component circuit with the created software. The handlebar head was modified mechanically and electronically by adding several actuators to pull the clutch handle and move the steering linkage. A DC motor was installed as a controlling actuator on the tension handle and throttle lever. The final modification to the main bar was the addition of a weight balancer hanger to ensure that the tractor's front and rear loads were balanced. This system used Bluetooth to communicate between the actuator and the mobile application. The integration of software and electronic circuits has been completed successfully, allowing tractor operators to operate remotely. Numerous trials have been conducted to evaluate the tractor's movement in an open field. The field test was done in two phases: prototype and final product, to generate the low-cost controlling system of the two-wheeled tractor using low energy use. The test results show a cost efficiency of 21.74% and 84.62% battery usage efficiency for the final product. The prototype design has been put through its paces on a G 3000 tractor. Numerous analysis results and enhancements are obtained from this test, which is required to create more efficient cost and battery life products. Based on the first test results, mechanical and electronic design modifications have been made and tested on a tractor of the G1000 type, with improved outcomes.

In addition, preliminary studies were carried out to create mathematical models and simulations of tractor movements. This model is based on kinematics for differentially driven two-wheeled vehicles. The model moves in the direction specified by the coordinates (back and forth path). The simulation results show that Supervisory Logic Control reaches 100%, better than Pure Pursuit Control, which only gets 60% from the specified point. These two-controller algorithms are the results of simulations in the Matlab/Simulink application and became a limitation of this research. The researcher also collect data from GPS and Compass sensors during the field trials using IoT technology.

The raw data results are noisy and unstable; therefore, Kalman and second-order Butterworth Low Pass Filters are used. The RMSE test results show that the Butterworth LPF performance is better than the Kalman Filter. Further research is needed to determine the effects of field tests on tractors. Therefore, a path planning and waypoint platform to automate tractor movements is carried out.

A Two-Dimensional path planning platform using an edge-vertex path algorithm for the Autonomous Walk Behind Hand Tractor has been presented. The algorithm automatically generates paths by considering the path interval distance, start point, and finish point. Field trials were conducted with a walk-behind tractor to validate the resulting waypoint coordinates for rice field plowing missions.

In this study, the researcher also introduced a novel, comprehensive, and diverse dataset called the RIFIS dataset to allow the researchers to develop the process of automation of ploughing fields using hand tractors. The RIFIS dataset contained 3723 images, 18 videos, and a JSON file with polygonal and bounding box labeling values for 970 images. The RIFIS dataset could automate the ploughing of rice fields not just at the time of rice planting but also at the time of rice harvest and for various other purposes throughout the year. This was the first-ever compilation of rice field sidewalk annotations. The RIFIS enabled training deep learning models for sidewalk detection in paddy fields. To assess the quality of the RIFIS dataset, a Mask-RCNN model was employed to develop a preliminary sidewalk detection algorithm. It was projected to improve the fine-grained segmentation of sidewalk site discoveries and reduce false positives and negatives for deep learning models. As supplementary data, the tractor location and orientation excel files were included with 'yaw', 'pitch', and 'roll' values obtained from the gyroscope sensor; 'x', 'y', and 'z' values from the accelerometer sensor; and 'a' (azimuth) values from the compass sensor and the location of the tractor from the GPS sensor. This allowed the researchers to examine the movement patterns of the tractor. The main goal of the researcher RIFIS dataset was that the research and models based on the RIFIS dataset could be used for sidewalk detection, distance prediction, tractor location, and orientation tracking to build an innovative tractor autonomous control system.

In conclusion, this research has fulfilled all of the proposed objectives by developing a Tractor Controlling System (TROLLS) via remote control so that farmers

do not need to be tired of plowing fields. In addition, a path-planning platform has also been created to assist farmers in implementing waypoint technology on tractors by considering various sizes of puddlers. The collection of datasets is also carried out as material for analysis of sidewalk detection using a camera. The Deep Learning Mask-RCNN technique was used to validate the collected dataset.

This study had four significant limitations that could be addressed in future research. First, future research aims to design a path with a spiral pattern and to account for external disturbances such as land contours when constructing the path. In addition, the use of IoT technology is possible in the future because this path-planning platform can be implemented in the cloud. Second, the use of low-cost technology makes this platform less precise. Future development using Real-Time Kinematic (RTK) GNSS could achieve better precision. Third, the RIFIS dataset was exclusively collected from paddy fields in Bali, Indonesia. Fourth, this research was limited to collecting images, videos, and annotations of paddy field sidewalks. Further research on integrating camera detection results and sensor readings is still needed. As a future development, sidewalk detection results using Mask R-CNN can be combined with basic image processing and detecting the distance between the lower center point of the image and the generated mask. The basic concepts of further research that can be developed as seen in Figure 5.1. This method can be implemented on all three cameras and then combined with the reading of several sensors to decide the tractor's movement. In the future, researchers aim to make more in-depth comparisons to more precisely detect the sidewalk's location and automate cultivating rice fields using hand tractors.

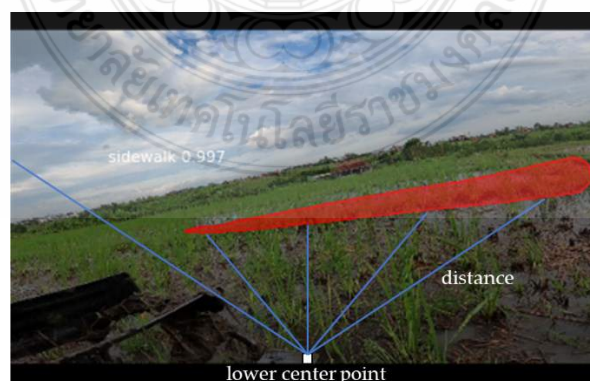


Figure 5. 1. Future research.

List of Bibliography

- Abouzahir, S., Sadik, M., & Sabir, E. (2017). **IoT-empowered smart agriculture: A real-time light-weight embedded segmentation system**. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10542 LNCS, 319–332. https://doi.org/10.1007/978-3-319-68179-5_28
- Adiyaksa, F., & Nugroho Djojomartono, P. (2020). Evaluation of Land Use Change of Agricultural Land into Industrial Area in Kendal Regency in Period 2014-2018. *JGISE*, 3(1). <https://doi.org/10.22146/jgise>
- Ahmadi Jeyed, H., & Ghaffari, A. (2019). **Nonlinear estimator design based on extended Kalman filter approach for state estimation of articulated heavy vehicle**. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-Body Dynamics*, 233(2), 254–265. <https://doi.org/10.1177/1464419318772173>
- Alipour, K., Robat, A. B., & Tarvirdizadeh, B. (2019). Dynamics modeling and sliding mode control of tractor-trailer wheeled mobile robots subject to wheels slip. *Mechanism and Machine Theory*, 138, 16–37. <https://doi.org/10.1016/j.mechmachtheory.2019.03.038>
- Almoaili, E., & Kurdi, H. (2020). **Path planning algorithm for unmanned ground vehicles (UGVs) in known static environments**. *Procedia Computer Science*, 177, 57–63. <https://doi.org/10.1016/j.procs.2020.10.011>
- An, V., Qu, Z., Crosby, F., Roberts, R., & An, V. (2020). A Triangulation-Based Coverage Path Planning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(6), 2157–2169. <https://doi.org/10.1109/TSMC.2018.2806840>
- Anitha, K. (2018). Role of Kalman Filters in Probabilistic Algorithm. *International Journal of Pure and Applied Mathematics*. 118 (11), 5–10. <https://doi.org/10.12732/ijpam.v118i11.2>
- Ardli Swardana. (2020). **Pemanfaatan Data SIG Untuk Analisis Perubahan Penggunaan Lahan Sawah di Kabupaten Garut (2009-2018)**. *Conference on*

- Innovation and Application of Science and Technology (CIASTECH 2020)*, 3(1), 299–304.
- Arkin, E. M., Fekete, S. P., & Mitchell, J. S. B. (2000). Approximation algorithms for lawn mowing and milling. In *Computational Geometry*, 17(1-2), 25-50.
- Asadi, K., Kalkunte Suresh, A., Ender, A., Gotad, S., Maniyar, S., Anand, S., Noghabaei, M., Han, K., Lobaton, E., & Wu, T. (2020). An integrated UGV-UAV system for construction site data collection. *Automation in Construction*, 112. <https://doi.org/10.1016/j.autcon.2019.103068>
- Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787–2805. <https://doi.org/10.1016/j.comnet.2010.05.010>
- Balafoutis, A. T., van Evert, F. K., & Fountas, S. (2020). Smart farming technology trends: Economic and environmental effects, labor impact, and adoption readiness. *Agronomy*, 10(5). <https://doi.org/10.3390/agronomy10050743>
- Bao, J., Yao, X., Tang, H., & Song, A. (2018). *Outdoor Navigation of a Mobile Robot by Following GPS Waypoints and Local Pedestrian Lane*. IEEE.
- Bar-Shalom, Y., Tian, X., & Willett, P. K. (2011). *Tracking and data fusion : a handbook of algorithms*. YBS Publishing.
- Berber, M., Ustun, A., & Yetkin, M. (2012). Comparison of accuracy of GPS techniques. *Measurement: Journal of the International Measurement Confederation*, 45(7), 1742–1746. <https://doi.org/10.1016/j.measurement.2012.04.010>
- Berrabah, S. A., & Baudoin, Y. (2011). **GPS data correction using encoders and inertial navigation system (INS) sensors**. In *Using Robots in Hazardous Environments* (pp. 269–282). Elsevier. <https://doi.org/10.1533/9780857090201.2.269>
- Binh, N. T., Tung, N. A., Nam, D. P., & Quang, N. H. (2019). An Adaptive Backstepping Trajectory Tracking Control of a Tractor Trailer Wheeled Mobile Robot. *International Journal of Control, Automation and Systems*, 17(2), 465–473. <https://doi.org/10.1007/s12555-017-0711-0>
- Bleiholder, J., & Naumann, F. (2009). Data Fusion. *ACM Computing Surveys*, 41(1), 1–41. <https://doi.org/10.1145/1456650.1456651>

- Blok, P. M., Kootstra, G., Elghor, H. E., Diallo, B., van Evert, F. K., & van Henten, E. J. (2022). Active learning with MaskAL reduces annotation effort for training Mask R-CNN on a broccoli dataset with visually similar classes. *Computers and Electronics in Agriculture*, **197**. <https://doi.org/10.1016/j.compag.2022.106917>
- Bohrer, B., Tan, Y. K., Mitsch, S., Sogokon, A., & Platzer, A. (2019). A Formal Safety Net for Waypoint-Following in Ground Robots. *IEEE Robotics and Automation Letters*, **4**(3), 2910–2917. <https://doi.org/10.1109/LRA.2019.2923099>
- Bonadies, S., & Gadsden, S. A. (2019). An overview of autonomous crop row navigation strategies for unmanned ground vehicles. *Engineering in Agriculture, Environment and Food*, **12**(1), 24–31. <https://doi.org/10.1016/j.eaef.2018.09.001>
- Boursianis, A. D., Papadopoulou, M. S., Diamantoulakis, P., Liopa-Tsakalidi, A., Barouchas, P., Salahas, G., Karagiannidis, G., Wan, S., & Goudos, S. K. (2020). Internet of Things (IoT) and Agricultural Unmanned Aerial Vehicles (UAVs) in smart farming: A comprehensive review. *Internet of Things*, **18**, 100187. <https://doi.org/10.1016/j.iot.2020.100187>
- Cabreira, T. M., Brisolará, L. B., & Ferreira Paulo, R. (2019). Survey on coverage path planning with unmanned aerial vehicles. *Drones*, **3**(1), 1–38. <https://doi.org/10.3390/drones3010004>
- Castanedo, F. (2013). A review of data fusion techniques. In *The Scientific World Journal*, **2013**. Hindawi Publishing Corporation. <https://doi.org/10.1155/2013/704504>
- Castillejo, P., Johansen, G., Cürüklü, B., Bilbao-Arechabala, S., Fresco, R., Martínez-Rodríguez, B., Pomante, L., Rusu, C., Martínez-Ortega, J. F., Centofanti, C., Hakojärvi, M., Santic, M., & Häggman, J. (2020). Aggregate Farming in the Cloud: The AFarCloud ECSEL project. *Microprocessors and Microsystems*, **78**. <https://doi.org/10.1016/j.micpro.2020.103218>
- Catania, P., Comparetti, A., Febo, P., Morello, G., Orlando, S., Roma, E., & Vallone, M. (2020). Positioning accuracy comparison of GNSS receivers used for mapping and guidance of agricultural machines. *Agronomy*, **10**(7). <https://doi.org/10.3390/agronomy10070924>

- Causa, F., & Fasano, G. (2021). Multiple UAVs trajectory generation and waypoint assignment in urban environment based on DOP maps. *Aerospace Science and Technology, 110*. <https://doi.org/10.1016/j.ast.2021.106507>
- Cecchinell, C., Jimenez, M., Mosser, S., & Riveill, M. (2014). *An Architecture to Support the Collection of Big Data in the Internet of Things*. In 2014 IEEE World congress on services, 442–449. <https://doi.org/10.1109/services.2014.83>
- Chang, L., Chen, Y. T., Wang, J. H., & Chang, Y. L. (2021). Rice-field mapping with sentinel-1a sar time-series data. *Remote Sensing, 13*(1), 1–25. <https://doi.org/10.3390/rs13010103>
- Choset, H., & Pignon, P. (1998). **Coverage Path Planning: The Boustrophedon Cellular Decomposition**. *Field and Service Robotics. Springer, London*, 203–209.
- Cohen, R., Fernie, G., & Fekr, A. R. (2020). A vision-based approach for sidewalk and walkway trip hazards assessment. *International Journal of Environmental Research and Public Health, 17*(22), 1–18. <https://doi.org/10.3390/ijerph17228438>
- Crisnapati, P. N., Maneetham, D., & Triandini, E. (2023). Trolls: a novel low-cost controlling system platform for walk-behind tractor. *International Journal of Electrical and Computer Engineering (IJECE), 13*(1), 842. <https://doi.org/10.11591/ijece.v13i1.pp842-858>
- Cutulle, M. A., & Maja, J. M. (2021). Determining the utility of an unmanned ground vehicle for weed control in specialty crop systems. *Italian Journal of Agronomy, 16*(4). <https://doi.org/10.4081/ija.2021.1865>
- Dadashzadeh, M., Abbaspour-Gilandeh, Y., Mesri-Gundoshmian, T., Sabzi, S., Hernández-Hernández, J. L., Hernández-Hernández, M., & Ignacio Arribas, J. (2020). Weed classification for site-specific weed management using an automated stereo computer-vision machine-learning system in rice fields. *Plants, 9*(5). <https://doi.org/10.3390/plants9050559>
- Das, A., Ghosal, M., & Das, D. (2020). Studies on applications of electronics in wheel slip control of agricultural tractor. *International Journal of Chemical Studies, 8*(2), 1483–1487. <https://doi.org/10.22271/chemi.2020.v8.i2w.8970>

- de Simone, M. C., Rivera, Z. B., & Guida, D. (2018). Obstacle avoidance system for unmanned ground vehicles by using ultrasonic sensors. *Machines*, *6*(2). <https://doi.org/10.3390/machines6020018>
- Dewangan, K. N., & Tewari, V. K. (2009). Characteristics of hand-transmitted vibration of a hand tractor used in three operational modes. *International Journal of Industrial Ergonomics*, *39*(1), 239–245. <https://doi.org/10.1016/j.ergon.2008.08.007>
- Dihingia, P. C., Kumar, G. V. P., Sarma, P. K., & Neog, P. (2018). Hand-Fed Vegetable Transplanter for Use with a Walk-Behind-Type Hand Tractor. *International Journal of Vegetable Science*, *24*(3), 254–273. <https://doi.org/10.1080/19315260.2017.1413477>
- Ding, Y., Yang, L., Zhang, D., Cui, T., Li, Y., Zhong, X., Xie, C., & Ding, Z. (2021). Novel low-cost control system for large high-speed corn precision planters. *International Journal of Agricultural and Biological Engineering*, *14*(2), 151–158. <https://doi.org/10.25165/J.IJABE.20211402.6053>
- El-Sheimy, N., & Youssef, A. (2020). Inertial sensors technologies for navigation applications: state of the art and future trends. *Satellite Navigation*, *1*(1). <https://doi.org/10.1186/s43020-019-0001-5>
- Fabbri, A., Cevoli, C., & Cantalupo, G. (2017). A method for handlebars ballast calculation in order to reduce vibrations transmissibility in walk behind tractors. *Journal of Agricultural Engineering*, *48*(2), 81–87. <https://doi.org/10.4081/jae.2017.599>
- Fang, S., Lu, Z., Wang, Z., Diao, X., Lu, Y., Gong, J., & Zhu, C. (2017). Design and prototype performance experiments of steering-by-wire hydraulic pressure system of tractor. *Nongye Gongcheng Xuebao/Transactions of the Chinese Society of Agricultural Engineering*, *33*(10), 86–93. <https://doi.org/10.11975/j.issn.1002-6819.2017.10.011>
- Fotio Tiotsop, L., Servetti, A., & Masala, E. (2020). An integer linear programming model for efficient scheduling of UGV tasks in precision agriculture under human supervision. *Computers and Operations Research*, *114*. <https://doi.org/10.1016/j.cor.2019.104826>

- Fouché, G. J., & Malekian, R. (2018). Drone as an autonomous aerial sensor system for motion planning. *Measurement: Journal of the International Measurement Confederation*, *119*, 142–155. <https://doi.org/10.1016/j.measurement.2018.01.027>
- Galceran, E., & Carreras, M. (2013). *A Survey on Coverage Path Planning for Robotics. Robotics and Autonomous systems*, *61*(12), 1258-1276.
- Gan, H., & Lee, W. S. (2018). *Development of a Navigation System for a Smart Farm. IFAC-PapersOnLine*, *51*(17), 1–4. <https://doi.org/10.1016/j.ifacol.2018.08.051>
- Guo, J., Li, X., Li, Z., Hu, L., Yang, G., Zhao, C., Fairbairn, D., Watson, D., & Ge, M. (2018). Multi-GNSS precise point positioning for precision agriculture. *Precision Agriculture*, *19*(5), 895–911. <https://doi.org/10.1007/s11119-018-9563-8>
- Gupta, C., Tewari, V. K., Ashok Kumar, A., & Shrivastava, P. (2019). Automatic tractor slip-draft embedded control system. *Computers and Electronics in Agriculture*, *165*. <https://doi.org/10.1016/j.compag.2019.104947>
- Hadas, E., Jozkow, G., Walicka, A., & Borkowski, A. (2019). Apple orchard inventory with a LiDAR equipped unmanned aerial system. *International Journal of Applied Earth Observation and Geoinformation*, *82*. <https://doi.org/10.1016/j.jag.2019.101911>
- Han, J., Xia, C., Shang, G., & Gao, X. (2017). In-field experiment of electro-hydraulic tillage depth draft-position mixed control on tractor. *IOP Conference Series: Materials Science and Engineering*, *274*(1). <https://doi.org/10.1088/1757-899X/274/1/012028>
- Han, X., Kim, H. J., Jeon, C. W., Moon, H. C., & Kim, J. H. (2017). Development of a low-cost GPS/INS integrated system for tractor automatic navigation. *International Journal of Agricultural and Biological Engineering*, *10*(2), 123–131. <https://doi.org/10.3965/j.ijabe.20171002.3070>
- Hasheminasab, S. M., Zhou, T., & Habib, A. (2020). GNSS/INS-Assisted structure from motion strategies for UAV-Based imagery over mechanized agricultural fields. *Remote Sensing*, *12*(3). <https://doi.org/10.3390/rs12030351>
- Huang, W. H. (2001). *Optimal Line-sweep-based Decompositions for Coverage Algorithms*. In Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164), 1, 27-32). IEEE.

- Ingle, V. K., & Proakis, J. G. (2012). *Digital Signal Processing Using MATLAB® Third Edition* (C. Valentine, Ed.; Third). Cengage Learning.
- Jain, A., & Kanhangad, V. (2018). Human Activity Classification in Smartphones Using Accelerometer and Gyroscope Sensors. *IEEE Sensors Journal*, *18*(3), 1169–1177. <https://doi.org/10.1109/JSEN.2017.2782492>
- Janulevičius, A., Damanauskas, V., & Pupinis, G. (2018). Effect of variations in front wheels driving lead on performance of a farm tractor with mechanical front-wheel-drive. *Journal of Terramechanics*, *77*, 23–30. <https://doi.org/10.1016/j.jterra.2018.02.002>
- Javad, M., & Saeid, S. (2021). Design, Fabrication and Evaluation A New Mechanism to Automatic Weight Transfer Control System on A Tractor. In *Emirates Journal for Engineering Research*, *26*(2). <https://scholarworks.uaeu.ac.ae/ejer> Available at:<https://scholarworks.uaeu.ac.ae/ejer/vol26/iss2/2>
- Jeon, C. W., Kim, H. J., Yun, C., Gang, M. S., & Han, X. (2021). An entry-exit path planner for an autonomous tractor in a paddy field. *Computers and Electronics in Agriculture*, *191*. <https://doi.org/10.1016/j.compag.2021.106548>
- Jinghong, L., Yanan, T., Yantao, Z., & Shuoyang, L. (2018). **Design and implementation of blind sidewalk recognition system based on FPGA**. In *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*, 2325–2329. <https://doi.org/10.1109/CCDC.2018.8407514>
- Journal, I. (2021). *Design and Implementation of Women Safety System Based On IOT Technology*, *International Research Journal of Engineering and Technology (IRJET)*, *8*(6), 1258-1261.
- Kapanoglu, M., Alikalfa, M., Ozkan, M., Yazıcı, A., & Parlaktuna, O. (2012). A pattern-based genetic algorithm for multi-robot coverage path planning minimizing completion time. *Journal of Intelligent Manufacturing*, *23*(4), 1035–1045. <https://doi.org/10.1007/s10845-010-0404-5>
- Kaplan, E., & Hegarty, C. (2017). *Understanding GPS/GNSS: Principles and Applications Second Edition*, Artech house.

- Kapsalis, D., Sename, O., Milanés, V., & Martínez Molina, J. J. (2021). Design and Experimental Validation of an LPV Pure Pursuit Automatic Steering Controller. *IFAC-PapersOnLine*, *54*(2), 63–68. <https://doi.org/10.1016/j.ifacol.2021.06.010>
- Karataş, G. B., Karagoz, P., & Ayran, O. (2021). Trajectory pattern extraction and anomaly detection for maritime vessels. *Internet of Things (Netherlands)*, *16*. <https://doi.org/10.1016/j.iot.2021.100436>
- Kassaeiyan, P., Alipour, K., & Tarvirdizadeh, B. (2020). A full-state trajectory tracking controller for tractor-trailer wheeled mobile robots. *Mechanism and Machine Theory*, *150*. <https://doi.org/10.1016/j.mechmachtheory.2020.103872>
- Kiratiratanapruk, K., Temniranrat, P., Kitvimonrat, A., Sinthupinyo, W., & Patarapuwadol, S. (2020). Using Deep Learning Techniques to Detect Rice Diseases from Images of Rice Fields. In H. Fujita, P. Fournier-Viger, M. Ali, & J. Sasaki (Eds.), *Trends in Artificial Intelligence Theory and Applications Artificial Intelligence Practices*, *12144*, 225–237. Springer International Publishing. <https://doi.org/10.1007/978-3-030-55789-8>
- Kitiashvili, I. N. (2019). *Application of Synoptic Magnetograms for Prediction of Solar Activity Using Ensemble Kalman Filter*. In In Solar Heliospheric Interplanetary Environment (SHINE).
- Kragh, M. F., Christiansen, P., Laursen, M. S., Larsen, M., Steen, K. A., Green, O., Karstoft, H., & Jørgensen, R. N. (2017). FieldSAFE: Dataset for obstacle detection in agriculture. *Sensors (Switzerland)*, *17*(11). <https://doi.org/10.3390/s17112579>
- Kumar, G. V. P., & Raheman, H. (2011). Development of a walk-behind type hand tractor powered vegetable transplanter for paper pot seedlings. *Biosystems Engineering*, *110*(2), 189–197. <https://doi.org/10.1016/j.biosystemseng.2011.08.001>
- Kumar, H., & Pimparkar, P. (2018). Data Fusion for the Internet of Things. *International Journal of Scientific and Research Publications (IJSRP)*, *8*(3). <https://doi.org/10.29322/ijsrp.8.3.2018.p7541>
- Lakhwani, K., Gianey, H., Agarwal, N., & Gupta, S. (2019). Development of IoT for Smart Agriculture a Review. *Advances in Intelligent Systems and Computing*, *841*, 425–432. https://doi.org/10.1007/978-981-13-2285-3_50

- Lakitan, B., Lindiana, L., Widuri, L. I., Kartika, K., Siaga, E., Meihana, M., & Wijaya, A. (2019). Inclusive and ecologically-sound food crop cultivation at tropical non-tidal wetlands in Indonesia. *Agrivita*, *41*(1), 23–31. <https://doi.org/10.17503/agrivita.v40i0.1717>
- Le, A. V., Prabakaran, V., Sivanantham, V., & Mohan, R. E. (2018). Modified a-star algorithm for efficient coverage path planning in tetris inspired self-reconfigurable robot with integrated laser sensor. *Sensors (Switzerland)*, *18*(8). <https://doi.org/10.3390/s18082585>
- Lee, S. K., Yoon, S. Y., & Won, J. S. (2018). Vegetation height estimate in rice fields using single polarization TanDEM-X Science Phase data. *Remote Sensing*, *10*(11). <https://doi.org/10.3390/rs10111702>
- Lezoche, M., Panetto, H., Kacprzyk, J., Hernandez, J. E., & Alemany Díaz, M. M. E. (2020). Agri-food 4.0: A survey of the Supply Chains and Technologies for the Future Agriculture. In *Computers in Industry*, *117*. Elsevier B.V. <https://doi.org/10.1016/j.compind.2020.103187>
- Li, Q., Li, R., Ji, K., & Dai, W. (2016). **Kalman filter and its application**. *Proceedings - 8th International Conference on Intelligent Networks and Intelligent Systems, ICINIS 2015*, 74–77. <https://doi.org/10.1109/ICINIS.2015.35>
- Li, S., Lu, J., Liang, G., Wu, X., Zhang, M., Plougonven, E., Wang, Y., Gao, L., Abdelrhman, A. A., Song, X., Liu, X., & Degré, A. (2021). Factors governing soil water repellency under tillage management: The role of pore structure and hydrophobic substances. *Land Degradation and Development*, *32*(2), 1046–1059. <https://doi.org/10.1002/ldr.3779>
- Li, Y., Chen, H., Joo Er, M., & Wang, X. (2011). Coverage path planning for UAVs based on enhanced exact cellular decomposition method. *Mechatronics*, *21*(5), 876–885. <https://doi.org/10.1016/j.mechatronics.2010.10.009>
- Liu, J., & Guo, G. (2021). Vehicle Localization during GPS Outages with Extended Kalman Filter and Deep Learning. *IEEE Transactions on Instrumentation and Measurement*, *70*. <https://doi.org/10.1109/TIM.2021.3097401>
- Livada, B., Vujić, S., Radić, D., Unkašević, T., & Banjac, Z. (2019). Digital magnetic compass integration with stationary, land-based electro-optical multi-sensor

- surveillance system. *Sensors (Switzerland)*, **19**(19).
<https://doi.org/10.3390/s19194331>
- Ludwig, S. A., & Jiménez, A. R. (2018). **Optimization of gyroscope and accelerometer/magnetometer portion of basic attitude and heading reference system**. *5th IEEE International Symposium on Inertial Sensors and Systems, INERTIAL 2018 - Proceedings*, 1–4. <https://doi.org/10.1109/ISISS.2018.8358127>
- Magnetic-Declination.com. (2022, February 1). *Magnetic Declination*.
<https://www.magnetic-declination.com>
- Mahbub, M. (2020). A smart farming concept based on smart embedded electronics, internet of things and wireless sensor network. *Internet of Things*, **9**, 100161.
<https://doi.org/10.1016/j.iot.2020.100161>
- Mammarella, M., Comba, L., Biglia, A., Dabbene, F., & Gay, P. (2020). **Cooperative Agricultural Operations of Aerial and Ground Unmanned Vehicles**. *2020 IEEE International Workshop on Metrology for Agriculture and Forestry, MetroAgriFor 2020 - Proceedings*, 224–229.
<https://doi.org/10.1109/MetroAgriFor50201.2020.9277573>
- Mammarella, M., Comba, L., Biglia, A., Dabbene, F., & Gay, P. (2021). Cooperation of unmanned systems for agricultural applications: A theoretical framework. *Biosystems Engineering*, **223**, 61-80.
<https://doi.org/10.1016/j.biosystemseng.2021.11.008>
- Mansouri, S. S., Kanellakis, C., Fresk, E., Kominiak, D., & Nikolakopoulos, G. (2018). Cooperative coverage path planning for visual inspection. *Control Engineering Practice*, **74**, 118–131. <https://doi.org/10.1016/j.conengprac.2018.03.002>
- Mebarki, N., Rekioua, T., Mokrani, Z., & Rekioua, D. (2015). Supervisor control for stand-alone photovoltaic/hydrogen/ battery bank system to supply energy to an electric vehicle. *International Journal of Hydrogen Energy*, **40**(39), 13777–13788.
<https://doi.org/10.1016/j.ijhydene.2015.03.024>
- Moinfar, A. M., Shahgholi, G., Gilandeh, Y. A., & Gundoshmian, T. M. (2020). The effect of the tractor driving system on its performance and fuel consumption. *Energy*, **202**. <https://doi.org/10.1016/j.energy.2020.117803>

- Moravec, H. P., & Elfes, A. (1985). **High Resolution Maps from Wide Angle Sonar.** *1985 IEEE International Conference on Robotics and Automation*, 116–121.
- Namdari, M., Rafiee, S., & Jafari, A. (2011). Using the FMEA method to Optimize fuel consumption in Tillage by Moldboard Plow. In *International Journal of Applied Engineering Research, Dindigul*, 1(4), 734-742.
- Negrete, J. C. (2020). Analysis of the current situation of two wheels tractors in Mexico. *Horticulture International Journal*, 4(1), 28-33.
<https://doi.org/10.15406/hij.2020.04.00152>
- Nguyen, T. T., Ospina, R., Noguchi, N., Okamoto, H., & Ngo, Q. H. (2021). Real-time disease detection in rice fields in the vietnamese mekong delta. *Environmental Control in Biology*, 59(2), 77–85. <https://doi.org/10.2525/ecb.59.77>
- Nukala, R., Panduru, K., Shields, A., Riordan, D., Doody, P., & Walsh, J. (2016, August 1). **Internet of Things: A review from “Farm to Fork”.** In *2016 27th Irish Signals and Systems Conference, ISSC 2016*, 1-6.
<https://doi.org/10.1109/ISSC.2016.7528456>
- Nwakaire, J. N., Ezeagba, A. C., & Ogoegbulem, O. C. (2018). Refurbishment and evaluation of a two-wheeled tractor. *Nigerian Journal of Technology*, 37(4), 1168.
<https://doi.org/10.4314/njt.v37i4.42>
- Ohi, N., Lassak, K., Watson, R., Strader, J., Du, Y., Yang, C., Hedrick, G., Nguyen, J., Harper, S., Reynolds, D., Kilic, C., Hikes, J., Mills, S., Castle, C., Buzzo, B., Waterland, N., Gross, J., Park, Y.-L., Li, X., & Gu, Y. (2018). **Design of an Autonomous Precision Pollination Robot.** *2018 IEEE RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7711–7718.
- Ospina, R., & Noguchi, N. (2020). Improved inclination correction method applied to the guidance system of agricultural vehicles. *International Journal of Agricultural and Biological Engineering*, 13(6), 183–194.
<https://doi.org/10.25165/j.ijabe.20201306.6012>
- Paman, U., Uchida, S., & Inaba, S. (2010). Economic potential of tractor hire business in Riau Province, Indonesia: A case study of small tractors for small rice farms. In *Agric Eng Int: CIGR Journal* 12(1). <http://www.cigrjournal.org>

- Preparata, F. P., & Shamos, M. I. (2012). *Computational Geometry: An Introduction* (D. Gries & F. Schneider, Eds.). Springer Science & Business Media.
- Qadri, S., Aslam, T., Nawaz, S. A., Saher, N., Razzaq, A., Ur Rehman, M., Ahmad, N., Shahzad, F., & Furqan Qadri, S. (2021). A machine vision approach for classification the rice varieties using statistical features. *International Journal of Food Properties*, *24*(1), 1615–1630. <https://doi.org/10.1080/10942912.2021.1986523>
- Qi, L., Zhang, T., Xu, K., Pan, H., Zhang, Z., & Yuan, Y. (2021). A novel terrain adaptive omni-directional unmanned ground vehicle for underground space emergency: Design, modeling and tests. *Sustainable Cities and Society*, *65*. <https://doi.org/10.1016/j.scs.2020.102621>
- Quaglia, G., Visconte, C., Scimmi, L. S., Melchiorre, M., Cavallone, P., & Pastorelli, S. (2019). Design of the positioning mechanism of an unmanned ground vehicle for precision agriculture. *Mechanisms and Machine Science*, *73*, 3531–3540. https://doi.org/10.1007/978-3-030-20131-9_348
- RADMANESH, M., SHARMA, B., KUMAR, M., & FRENCH, D. (2021). PDE solution to UAV/UGV trajectory planning problem by spatio-temporal estimation during wildfires. *Chinese Journal of Aeronautics*, *34*(5), 601–616. <https://doi.org/10.1016/j.cja.2020.11.002>
- Rahman, M. M., Ishii, K., & Noguchi, N. (2019). Optimum harvesting area of convex and concave polygon field for path planning of robot combine harvester. *Intelligent Service Robotics*, *12*(2), 167–179. <https://doi.org/10.1007/s11370-018-00273-4>
- Rains, G. C., Faircloth, A. G., Thai, C., & Raper, R. L. (2014). Evaluation of a simple pure pursuit path-following algorithm for an autonomous, articulated-steer vehicle. *Applied Engineering in Agriculture*, *30*(3), 367–374. <https://doi.org/10.13031/aea.30.10347>
- Ramadhani, F., Pullanagari, R., Kereszturi, G., & Procter, J. (2020). Automatic mapping of rice growth stages using the integration of sentinel-2, mod13q1, and sentinel-1. *Remote Sensing*, *12*(21), 1–21. <https://doi.org/10.3390/rs12213613>

- Roldán, J. J., Cerro, J. del, Garzón-Ramos, D., Garcia-Aunon, P., Garzón, M., León, J. de, & Barrientos, A. (2018). **Robots in Agriculture: State of Art and Practical Experiences**. In *Service Robots*, 67-90. <https://doi.org/10.5772/intechopen.69874>
- Rondelli, V., Franceschetti, B., & Mengoli, D. (2022). A Review of Current and Historical Research Contributions to the Development of Ground Autonomous Vehicles for Agriculture. *Sustainability*, *14*(15), 9221. <https://doi.org/10.3390/su14159221>
- Ruiz-Larrea, A., Roldán, J. J., Garzón, M., del Cerro, J., & Barrientos, A. (2016). A UGV approach to measure the ground properties of greenhouses. In *Advances in Intelligent Systems and Computing*, *418*, 3–13. Springer Verlag. https://doi.org/10.1007/978-3-319-27149-1_1
- Sadhish Prabhu, Kannan, Indra Gandhi, Irfanuddin, & Munawir. (2018). **GPS Controlled Autonomous Bot for Unmanned Delivery**, 128-132. IEEE.
- Santos, L. C., Santos, F. N., Solteiro Pires, E. J., Valente, A., Costa, P., & Magalhaes, S. (2020). **Path planning for ground robots in agriculture: A short review**. *2020 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2020*, 61–66. <https://doi.org/10.1109/ICARSC49921.2020.9096177>
- Sarkka, O., Nieminen, T., Suuriniemi, S., & Kettunen, L. (2017). A Multi-Position Calibration Method for Consumer-Grade Accelerometers, Gyroscopes, and Magnetometers to Field Conditions. *IEEE Sensors Journal*, *17*(11), 3470–3481. <https://doi.org/10.1109/JSEN.2017.2694488>
- Shafaei, S. M., Loghavi, M., & Kamgar, S. (2018). An extensive validation of computer simulation frameworks for neural prognostication of tractor tractive efficiency. *Computers and Electronics in Agriculture*, *155*, 283–297. <https://doi.org/10.1016/j.compag.2018.10.027>
- Shafaei, S. M., Loghavi, M., & Kamgar, S. (2019). Development and implementation of a human machine interface-assisted digital instrumentation system for high precision measurement of tractor performance parameters. *Engineering in Agriculture, Environment and Food*, *12*(1), 11–23. <https://doi.org/10.1016/j.eaef.2018.08.006>
- Shafaei, S. M., Loghavi, M., & Kamgar, S. (2020a). Ascertainment of driving lead of tractor front wheels as loaded by draft force. *Measurement: Journal of the*

- International Measurement Confederation*, 165.
<https://doi.org/10.1016/j.measurement.2020.108134>
- Shafaei, S. M., Loghavi, M., & Kamgar, S. (2020b). Benchmark of an intelligent fuzzy calculator for admissible estimation of drawbar pull supplied by mechanical front wheel drive tractor. *Artificial Intelligence in Agriculture*, 4, 209–218.
<https://doi.org/10.1016/j.aiia.2020.10.001>
- Shafaei, S. M., Loghavi, M., & Kamgar, S. (2021). Fundamental realization of longitudinal slip efficiency of tractor wheels in a tillage practice. *Soil and Tillage Research*, 205. <https://doi.org/10.1016/j.still.2020.104765>
- Shao, H., Tang, R., Lei, Y., Mu, J., Guan, Y., & Xiang, Y. (2021). Rice ear counting based on image segmentation and establishment of a dataset. *Plants*, 10(8).
<https://doi.org/10.3390/plants10081625>
- Sharma, R., Kamble, S. S., Gunasekaran, A., Kumar, V., & Kumar, A. (2020). A systematic literature review on machine learning applications for sustainable agriculture supply chain performance. *Computers and Operations Research*, 119.
<https://doi.org/10.1016/j.cor.2020.104926>
- Shiotsu, F., Sakagami, N., Asagi, N., Suprpta, D. N., Agustiani, N., Nitta, Y., & Komatsuzaki, M. (2015). Initiation and dissemination of organic rice cultivation in Bali, Indonesia. *Sustainability (Switzerland)*, 7(5), 5171–5181.
<https://doi.org/10.3390/su7055171>
- Shyrokau, B., de Winter, J., Stroosma, O., Dijksterhuis, C., Loof, J., van Paassen, R., & Happee, R. (2018). The effect of steering-system linearity, simulator motion, and truck driving experience on steering of an articulated tractor-semitrailer combination. *Applied Ergonomics*, 71, 17–28.
<https://doi.org/10.1016/j.apergo.2018.03.018>
- Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). *Introduction to Autonomous Mobile Robots*.
- Singh, D., Ichiura, S., & Katahira, M. (2020). **Growth information acquisition by unmanned ground vehicle and artificial intelligence in rice**. *ASABE 2020 Annual International Meeting*. <https://doi.org/10.13031/aim.202000315>

- Song, J., & Gupta, S. (2018). An Online Coverage Path Planning Algorithm. *IEEE Transactions on Robotics*, *34*(2), 526–533. <https://doi.org/10.1109/TRO.2017.2780259>
- Soylu, S., & Çarman, K. (2021). Fuzzy logic based automatic slip control system for agricultural tractors. *Journal of Terramechanics*, *95*, 25–32. <https://doi.org/10.1016/j.jterra.2021.03.001>
- Syuhada, A., Armanto, M. E., Siswanto, A., Yazid, M., & Wildayana, E. (2020). Food security and environmental sustainability on the south sumatra Wetlands, Indonesia. *Systematic Reviews in Pharmacy*, *11*(3), 457–464. <https://doi.org/10.5530/srp.2020.3.58>
- Togashi, F., Misaka, T., Löhner, R., & Obayashi, S. (2018). *Application of Ensemble Kalman Filter to Pedestrian Flow*. *Collective Dynamics*, *5*, 467–470.
- Tomera, M. (2016). **Hybrid real-time way-point controller for ships**. *2016 21st International Conference on Methods and Models in Automation and Robotics, MMAR 2016*, 630–635. <https://doi.org/10.1109/MMAR.2016.7575209>
- Vasquez-Gomez, J. I., Marciano-Melchor, M., Valentin, L., & Herrera-Lozada, J. C. (2020). Coverage Path Planning for 2D Convex Regions. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *97*(1), 81–94. <https://doi.org/10.1007/s10846-019-01024-y>
- Wang, H., Lyu, S., & Ren, Y. (2021). Paddy rice imagery dataset for panicle segmentation. *Agronomy*, *11*(8). <https://doi.org/10.3390/agronomy11081542>
- Wang, H., & Noguchi, N. (2018). Adaptive turning control for an agricultural robot tractor. *International Journal of Agricultural and Biological Engineering*, *11*(6), 113–119. <https://doi.org/10.25165/j.ijabe.20181106.3605>
- Wang, H., & Noguchi, N. (2019a). Navigation of a robot tractor using the centimeter level augmentation information via Quasi-Zenith Satellite System. *Engineering in Agriculture, Environment and Food*, *12*(4), 414–419. <https://doi.org/10.1016/j.eaef.2019.06.003>
- Wang, H., & Noguchi, N. (2019b). Navigation of a robot tractor using the centimeter level augmentation information via Quasi-Zenith Satellite System. *Engineering in*

- Agriculture, Environment and Food*, **12**(4), 414–419.
<https://doi.org/10.1016/j.eaef.2019.06.003>
- Wang, L., Lan, Y., Zhang, Y., Zhang, H., Tahir, M. N., Ou, S., Liu, X., & Chen, P. (2019). Applications and prospects of agricultural unmanned aerial vehicle obstacle avoidance technology in China. In *Sensors (Switzerland)* **19**(3). MDPI AG.
<https://doi.org/10.3390/s19030642>
- Wang, S., Sun, G., Zheng, B., & Du, Y. (2021). A crop image segmentation and extraction algorithm based on mask RCNN. *Entropy*, **23**(9).
<https://doi.org/10.3390/e23091160>
- Warden, P., & Situnayake, D. (2019). *TinyML Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O'Reilly Media.
- Wei, M., & Isler, V. (2018, May 21). **Coverage Path Planning under the Energy Constraint**. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 368-373.
- Wu, C., Chen, Z., Wang, D., Kou, Z., Cai, Y., & Yang, W. (2019). Behavior modelling and sensing for machinery operations using smartphone's sensor data: A case study of forage maize sowing. *International Journal of Agricultural and Biological Engineering*, **12**(6), 66–74. <https://doi.org/10.25165/j.ijabe.20191206.4702>
- Wu, T., & Hung, J. Y. (2017, May 10). **State estimation for a tractor-trailer system using adaptive unscented Kalman filter**. *Conference Proceedings - IEEE SOUTHEASTCON*, 1-5. <https://doi.org/10.1109/SECON.2017.7925342>
- Wulandari, Y. A., Hartadi, R., & Sunartomo, A. F. (2017). Analisis faktor-faktor yang mempengaruhi keputusan petani melakukan konversi lahan sawah dan dampaknya terhadap pendapatan petani (Studi Kasus Konversi Lahan Sawah di Kecamatan Kaliwates Kabupaten Jember). *Jurnal Agribest*, **1**(2), 152–167.
- Xiao, M. H., Zhao, J., Wang, Y. W., Zhang, H. J., Lu, Z. X., & Wei, W. H. (2018). Fuel economy of multiple conditions self-adaptive tractors with hydro-mechanical CVT. *International Journal of Agricultural and Biological Engineering*, **11**(3), 102–109. <https://doi.org/10.25165/j.ijabe.20181103.2158>
- Yakkundimath, R., Saunshi, G., Anami, B., & Palaiah, S. (2022). Classification of Rice Diseases using Convolutional Neural Network Models. *Journal of The Institution*

- of Engineers (India): Series B*, **103**(4), 1047-1059. <https://doi.org/10.1007/s40031-021-00704-4>
- Yang, M. der, Tseng, H. H., Hsu, Y. C., Yang, C. Y., Lai, M. H., & Wu, D. H. (2021). A UAV open dataset of rice paddies for deep learning practice. *Remote Sensing*, **13**(7). <https://doi.org/10.3390/rs13071358>
- Yin, X., Wang, Y., Chen, Y., Jin, C., & Du, J. (2020). Development of autonomous navigation controller for agricultural vehicles. *International Journal of Agricultural and Biological Engineering*, **13**(4), 70–76. <https://doi.org/10.25165/j.ijabe.20201304.5470>
- Yu, Y., Zhang, K., Yang, L., & Zhang, D. (2019). Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Computers and Electronics in Agriculture*, **163**. <https://doi.org/10.1016/j.compag.2019.06.001>
- Zein, Y., Darwiche, M., & Mokhiamar, O. (2018). GPS tracking system for autonomous vehicles. *Alexandria Engineering Journal*, **57**(4), 3127–3137. <https://doi.org/10.1016/j.aej.2017.12.002>
- Zhang, C., & Kovacs, J. M. (2012). The application of small unmanned aerial systems for precision agriculture: A review. In *Precision Agriculture*, **13**(6), 693–712). <https://doi.org/10.1007/s11119-012-9274-5>
- Zhao, Z., Zhang, Y., Long, L., Lu, Z., & Shi, J. (2022). Efficient and adaptive lidar–visual–inertial odometry for agricultural unmanned ground vehicle. *International Journal of Advanced Robotic Systems*, **19**(2). <https://doi.org/10.1177/17298806221094925>
- Zhou, H., Hu, L., Luo, X., Tang, L., Du, P., Mao, T., Zhao, R., & He, J. (2020). Design and test of laser-controlled paddy field levelling-beater. *International Journal of Agricultural and Biological Engineering*, **13**(1), 57–65. <https://doi.org/10.25165/j.ijabe.20201301.4989>
- Zhu, D., Tian, C., Sun, B., & Luo, C. (2019). Complete Coverage Path Planning of Autonomous Underwater Vehicle Based on GBNN Algorithm. *Journal of Intelligent and Robotic Systems: Theory and Applications*, **94**(1), 237–249. <https://doi.org/10.1007/s10846-018-0787-7>

Zoto, J., Musci, M. A., Khaliq, A., Chiaberge, M., & Aicardi, I. (2020a). Automatic Path Planning for Unmanned Ground Vehicle Using UAV Imagery. *Advances in Intelligent Systems and Computing*, **980**, 223–230. https://doi.org/10.1007/978-3-030-19648-6_26



APPENDICES





APPENDIX A
PYTHON PATH PLANNING SOURCE CODE


```

import math
import os
import sys
from enum import IntEnum

import numpy as np
from scipy.spatial.transform import Rotation as Rot
import matplotlib.pyplot as plt
#import pdb

sys.path.append(os.path.dirname(os.path.abspath(__file__)) +
"/../Mapping")

try:
    from grid_map_lib.grid_map_lib import GridMap
except ImportError:
    raise

do_animation = True

class SweepSearcher:
    class SweepDirection(IntEnum):
        UP = 1
        DOWN = -1

    class MovingDirection(IntEnum):
        RIGHT = 1
        LEFT = -1

    def __init__(self,
                 moving_direction, sweep_direction, x_inds_goal_y,
goal_y):
        self.moving_direction = moving_direction
        self.sweep_direction = sweep_direction
        self.turning_window = []
        self.update_turning_window()
        self.x_indexes_goal_y = x_inds_goal_y
        self.goal_y = goal_y

    def move_target_grid(self, c_x_index, c_y_index, grid_map):
        n_x_index = self.moving_direction + c_x_index
        n_y_index = c_y_index

        # found safe grid
        if not grid_map.check_occupied_from_xy_index(n_x_index,
n_y_index,
occupied_val=0.5):
            return n_x_index, n_y_index
        else: # occupied
            next_c_x_index, next_c_y_index =
self.find_safe_turning_grid(
                c_x_index, c_y_index, grid_map)
            if (next_c_x_index is None) and (next_c_y_index is None):
                # moving backward

```

```

        next_c_x_index = -self.moving_direction + c_x_index
        next_c_y_index = c_y_index
        if
grid_map.check_occupied_from_xy_index(next_c_x_index,
next_c_y_index):
        # moved backward, but the grid is occupied by
obstacle
            return None, None
        else:
            # keep moving until end
            while not grid_map.check_occupied_from_xy_index(
                next_c_x_index + self.moving_direction,
                next_c_y_index, occupied_val=0.5):
                next_c_x_index += self.moving_direction
            self.swap_moving_direction()
            return next_c_x_index, next_c_y_index

def find_safe_turning_grid(self, c_x_index, c_y_index, grid_map):

    for (d_x_ind, d_y_ind) in self.turning_window:

        next_x_ind = d_x_ind + c_x_index
        next_y_ind = d_y_ind + c_y_index

        # found safe grid
        if not grid_map.check_occupied_from_xy_index(next_x_ind,
next_y_ind,
occupied_val=0.5):
            return next_x_ind, next_y_ind

        return None, None

def is_search_done(self, grid_map):
    for ix in self.x_indexes_goal_y:
        if not grid_map.check_occupied_from_xy_index(ix,
self.goal_y,
occupied_val=0.5):
            return False

        # all lower grid is occupied
        return True

def update_turning_window(self):
    # turning window definition
    # robot can move grid based on it.
    self.turning_window = [
        (self.moving_direction, 0.0),
        (self.moving_direction, self.sweep_direction),
        (0, self.sweep_direction),
        (-self.moving_direction, self.sweep_direction),
    ]

def swap_moving_direction(self):

```

```

self.moving_direction *= -1
self.update_turning_window()

def search_start_grid(self, grid_map):
    x_inds = []
    y_ind = 0
    if self.sweep_direction == self.SweepDirection.DOWN:
        x_inds, y_ind = search_free_grid_index_at_edge_y(
            grid_map, from_upper=True)
    elif self.sweep_direction == self.SweepDirection.UP:
        x_inds, y_ind = search_free_grid_index_at_edge_y(
            grid_map, from_upper=False)

    if self.moving_direction == self.MovingDirection.RIGHT:
        return min(x_inds), y_ind
    elif self.moving_direction == self.MovingDirection.LEFT:
        return max(x_inds), y_ind

    raise ValueError("self.moving direction is invalid ")

def find_sweep_direction_and_start_position(ox, oy):
    # find sweep_direction
    max_dist = 0.0
    vec = [0.0, 0.0]
    sweep_start_pos = [0.0, 0.0]
    for i in range(len(ox) - 1):
        dx = ox[i + 1] - ox[i]
        dy = oy[i + 1] - oy[i]
        d = np.hypot(dx, dy)

        if d > max_dist:
            max_dist = d
            vec = [dx, dy]
            sweep_start_pos = [ox[i], oy[i]]

    return vec, sweep_start_pos

def convert_grid_coordinate(ox, oy, sweep_vec, sweep_start_position):
    tx = [ix - sweep_start_position[0] for ix in ox]
    ty = [iy - sweep_start_position[1] for iy in oy]
    th = math.atan2(sweep_vec[1], sweep_vec[0])
    rot = Rot.from_euler('z', th).as_matrix()[0:2, 0:2]
    converted_xy = np.stack([tx, ty]).T @ rot

    return converted_xy[:, 0], converted_xy[:, 1]

def convert_global_coordinate(x, y, sweep_vec, sweep_start_position):
    th = math.atan2(sweep_vec[1], sweep_vec[0])
    rot = Rot.from_euler('z', -th).as_matrix()[0:2, 0:2]
    converted_xy = np.stack([x, y]).T @ rot
    rx = [ix + sweep_start_position[0] for ix in converted_xy[:, 0]]
    ry = [iy + sweep_start_position[1] for iy in converted_xy[:, 1]]
    return rx, ry

```

```

def search_free_grid_index_at_edge_y(grid_map, from_upper=False):
    y_index = None
    x_indexes = []

    if from_upper:
        x_range = range(grid_map.height)[::-1]
        y_range = range(grid_map.width)[::-1]
    else:
        x_range = range(grid_map.height)
        y_range = range(grid_map.width)

    for iy in x_range:
        for ix in y_range:
            if not grid_map.check_occupied_from_xy_index(ix, iy):
                y_index = iy
                x_indexes.append(ix)
        if y_index:
            break

    return x_indexes, y_index

def setup_grid_map(ox, oy, resolution, sweep_direction,
offset_grid=10):
    width = math.ceil((max(ox) - min(ox)) / resolution) + offset_grid
    height = math.ceil((max(oy) - min(oy)) / resolution) + offset_grid
    center_x = (np.max(ox) + np.min(ox)) / 2.0
    center_y = (np.max(oy) + np.min(oy)) / 2.0

    grid_map = GridMap(width, height, resolution, center_x, center_y)
    grid_map.print_grid_map_info()
    grid_map.set_value_from_polygon(ox, oy, 1.0, inside=False)
    grid_map.expand_grid()

    x_inds_goal_y = []
    goal_y = 0
    if sweep_direction == SweepSearcher.SweepDirection.UP:
        x_inds_goal_y, goal_y = search_free_grid_index_at_edge_y(
            grid_map, from_upper=True)
    elif sweep_direction == SweepSearcher.SweepDirection.DOWN:
        x_inds_goal_y, goal_y = search_free_grid_index_at_edge_y(
            grid_map, from_upper=False)

    return grid_map, x_inds_goal_y, goal_y

def sweep_path_search(sweep_searcher, grid_map,
grid_search_animation=False):
    # search start grid
    c_x_index, c_y_index = sweep_searcher.search_start_grid(grid_map)
    if not grid_map.set_value_from_xy_index(c_x_index, c_y_index,
0.5):
        print("Cannot find start grid")
        return [], []

```

```

        x, y =
grid_map.calc_grid_central_xy_position_from_xy_index(c_x_index,
c_y_index)
        px, py = [x], [y]

        fig, ax = None, None
        if grid_search_animation:
            fig, ax = plt.subplots()
            # for stopping simulation with the esc key.
            fig.canvas.mpl_connect(
                'key_release_event',
                lambda event: [exit(0) if event.key == 'escape' else
None])

            while True:
                c_x_index, c_y_index =
sweep_searcher.move_target_grid(c_x_index,
c_y_index,
grid_map)

                if sweep_searcher.is_search_done(grid_map) or (
                    c_x_index is None or c_y_index is None):
                    print("Done")
                    break

                x, y = grid_map.calc_grid_central_xy_position_from_xy_index(
                    c_x_index, c_y_index)

                px.append(x)
                py.append(y)

                grid_map.set_value_from_xy_index(c_x_index, c_y_index, 0.5)

                if grid_search_animation:
                    grid_map.plot_grid_map(ax=ax)
                    plt.pause(1.0)

            return px, py

def planning(ox, oy, resolution,
            moving_direction=SweepSearcher.MovingDirection.RIGHT,
            sweeping_direction=SweepSearcher.SweepDirection.UP,
            ):
    sweep_vec, sweep_start_position =
find_sweep_direction_and_start_position(
        ox, oy)

    rox, roy = convert_grid_coordinate(ox, oy, sweep_vec,
                                       sweep_start_position)

```

```

    grid_map, x_inds_goal_y, goal_y = setup_grid_map(rox, roy,
resolution,
sweeping_direction)

    sweep_searcher = SweepSearcher(moving_direction,
sweeping_direction,
                                x_inds_goal_y, goal_y)

    px, py = sweep_path_search(sweep_searcher, grid_map)

    rx, ry = convert_global_coordinate(px, py, sweep_vec,
sweep_start_position)

    print("Path length:", len(rx))

    return rx, ry

def planning_animation(ox, oy, resolution): # pragma: no cover
    px, py = planning(ox, oy, resolution)

    # animation
    if do_animation:
        for ipx, ipy in zip(px, py):
            plt.cla()
            # for stopping simulation with the esc key.
            plt.gcf().canvas.mpl_connect(
                'key_release_event',
                lambda event: [exit(0) if event.key == 'escape' else
None])

            plt.plot(ox, oy, "-xb")
            plt.plot(px, py, "-r")
            plt.plot(ipx, ipy, "or")
            plt.axis("equal")
            plt.grid(True)
            plt.pause(0.1)

        plt.cla()
        plt.plot(ox, oy, "-xb")
        plt.plot(px, py, "-r")
        plt.axis("equal")
        plt.grid(True)
        plt.pause(0.1)
        plt.close()

#pdb.set_trace()
def main(): # pragma: no cover
    print("start!!")

    ox = [0.0, 20.0, 50.0, 100.0, 80.0, 40.0, 0.0]
    oy = [0.0, -40.0, -200.0, 60.0, 40.0, 100.0, 0.0]
    resolution = 5.0
    planning_animation(ox, oy, resolution)

    ox = [0.0, 20.0, 50.0, 100.0, 130.0, 40.0, 0.0]

```

```

oy = [0.0, -40.0, -200.0, 60.0, 100.0, 100.0, 0.0]
resolution = 5.0
planning_animation(ox, oy, resolution)

ox = [0.0, 20.0, 50.0, 100.0, 130.0, 40.0, 0.0]
oy = [0.0, -20.0, 0.0, 30.0, 60.0, 80.0, 0.0]
resolution = 5.0
planning_animation(ox, oy, resolution)

ox = [0.0, 50.0, 50.0, 0.0, 0.0]
oy = [0.0, 0.0, 30.0, 30.0, 0.0]
resolution = 1.3
planning_animation(ox, oy, resolution)

ox = [0.0, 20.0, 50.0, 200.0, 130.0, 40.0, 0.0]
oy = [0.0, -80.0, 0.0, 30.0, 60.0, 80.0, 0.0]
resolution = 5.0
planning_animation(ox, oy, resolution)

if do_animation:
    plt.show()
print("done!!")

if __name__ == '__main__':
    main()

```







APPENDIX B
PYTHON GENERATE GRID MAP MATPLOTLIB

```

import matplotlib.pyplot as plt
import numpy as np

class GridMap:
    """
    GridMap class
    """

    def __init__(self, width, height, resolution,
                 center_x, center_y, init_val=0.0):
        """__init__

        :param width: number of grid for width
        :param height: number of grid for height
        :param resolution: grid resolution [m]
        :param center_x: center x position [m]
        :param center_y: center y position [m]
        :param init_val: initial value for all grid
        """
        self.width = width
        self.height = height
        self.resolution = resolution
        self.center_x = center_x
        self.center_y = center_y

        self.left_lower_x = self.center_x - self.width / 2.0 *
self.resolution
        self.left_lower_y = self.center_y - self.height / 2.0 *
self.resolution

        self.ndata = self.width * self.height
        self.data = [init_val] * self.ndata

    def get_value_from_xy_index(self, x_ind, y_ind):
        """get_value_from_xy_index

        when the index is out of grid map area, return None

        :param x_ind: x index
        :param y_ind: y index
        """

        grid_ind = self.calc_grid_index_from_xy_index(x_ind, y_ind)

        if 0 <= grid_ind < self.ndata:
            return self.data[grid_ind]
        else:
            return None

    def get_xy_index_from_xy_pos(self, x_pos, y_pos):
        """get_xy_index_from_xy_pos

        :param x_pos: x position [m]
        :param y_pos: y position [m]
        """

```

```

x_ind = self.calc_xy_index_from_position(
    x_pos, self.left_lower_x, self.width)
y_ind = self.calc_xy_index_from_position(
    y_pos, self.left_lower_y, self.height)

return x_ind, y_ind

def set_value_from_xy_pos(self, x_pos, y_pos, val):
    """set_value_from_xy_pos

return bool flag, which means setting value is succeeded or
not

:param x_pos: x position [m]
:param y_pos: y position [m]
:param val: grid value
"""

x_ind, y_ind = self.get_xy_index_from_xy_pos(x_pos, y_pos)

if (not x_ind) or (not y_ind):
    return False # NG

flag = self.set_value_from_xy_index(x_ind, y_ind, val)

return flag

def set_value_from_xy_index(self, x_ind, y_ind, val):
    """set_value_from_xy_index

return bool flag, which means setting value is succeeded or
not

:param x_ind: x index
:param y_ind: y index
:param val: grid value
"""

if (x_ind is None) or (y_ind is None):
    return False, False

grid_ind = int(y_ind * self.width + x_ind)

if 0 <= grid_ind < self.ndata:
    self.data[grid_ind] = val
    return True # OK
else:
    return False # NG

def set_value_from_polygon(self, pol_x, pol_y, val, inside=True):
    """set_value_from_polygon

Setting value inside or outside polygon

:param pol_x: x position list for a polygon
:param pol_y: y position list for a polygon

```

```

:param val: grid value
:param inside: setting data inside or outside
"""

# making ring polygon
if (pol_x[0] != pol_x[-1]) or (pol_y[0] != pol_y[-1]):
    pol_x.append(pol_x[0])
    pol_y.append(pol_y[0])

# setting value for all grid
for x_ind in range(self.width):
    for y_ind in range(self.height):
        x_pos, y_pos =
self.calc_grid_central_xy_position_from_xy_index(
    x_ind, y_ind)

        flag = self.check_inside_polygon(x_pos, y_pos, pol_x,
pol_y)

        if flag is inside:
            self.set_value_from_xy_index(x_ind, y_ind, val)

def calc_grid_index_from_xy_index(self, x_ind, y_ind):
    grid_ind = int(y_ind * self.width + x_ind)
    return grid_ind

def calc_grid_central_xy_position_from_xy_index(self, x_ind,
y_ind):
    x_pos = self.calc_grid_central_xy_position_from_index(
        x_ind, self.left_lower_x)
    y_pos = self.calc_grid_central_xy_position_from_index(
        y_ind, self.left_lower_y)

    return x_pos, y_pos

def calc_grid_central_xy_position_from_index(self, index,
lower_pos):
    return lower_pos + index * self.resolution + self.resolution /
2.0

def calc_xy_index_from_position(self, pos, lower_pos, max_index):
    ind = int(np.floor((pos - lower_pos) / self.resolution))
    if 0 <= ind <= max_index:
        return ind
    else:
        return None

def check_occupied_from_xy_index(self, xind, yind,
occupied_val=1.0):

    val = self.get_value_from_xy_index(xind, yind)

    if val is None or val >= occupied_val:
        return True
    else:
        return False

```

```

def expand_grid(self):
    xinds, yinds = [], []

    for ix in range(self.width):
        for iy in range(self.height):
            if self.check_occupied_from_xy_index(ix, iy):
                xinds.append(ix)
                yinds.append(iy)

    for (ix, iy) in zip(xinds, yinds):
        self.set_value_from_xy_index(ix + 1, iy, val=1.0)
        self.set_value_from_xy_index(ix, iy + 1, val=1.0)
        self.set_value_from_xy_index(ix + 1, iy + 1, val=1.0)
        self.set_value_from_xy_index(ix - 1, iy, val=1.0)
        self.set_value_from_xy_index(ix, iy - 1, val=1.0)
        self.set_value_from_xy_index(ix - 1, iy - 1, val=1.0)

    @staticmethod
    def check_inside_polygon(iox, ioy, x, y):

        npoint = len(x) - 1
        inside = False
        for i1 in range(npoint):
            i2 = (i1 + 1) % (npoint + 1)

            if x[i1] >= x[i2]:
                min_x, max_x = x[i2], x[i1]
            else:
                min_x, max_x = x[i1], x[i2]
            if not min_x < iox < max_x:
                continue

            tmp1 = (y[i2] - y[i1]) / (x[i2] - x[i1])
            if (y[i1] + tmp1 * (iox - x[i1]) - ioy) > 0.0:
                inside = not inside

        return inside

    def print_grid_map_info(self):
        print("width:", self.width)
        print("height:", self.height)
        print("resolution:", self.resolution)
        print("center_x:", self.center_x)
        print("center_y:", self.center_y)
        print("left_lower_x:", self.left_lower_x)
        print("left_lower_y:", self.left_lower_y)
        print("ndata:", self.ndata)

    def plot_grid_map(self, ax=None):

        grid_data = np.reshape(np.array(self.data), (self.height,
self.width))
        if not ax:
            fig, ax = plt.subplots()

```

```

        heat_map = ax.pcolor(grid_data, cmap="Blues", vmin=0.0,
vmax=1.0)
        plt.axis("equal")
        # plt.show()

        return heat_map

def test_polygon_set():
    ox = [0.0, 20.0, 50.0, 100.0, 130.0, 40.0]
    oy = [0.0, -20.0, 0.0, 30.0, 60.0, 80.0]

    grid_map = GridMap(600, 290, 0.7, 60.0, 30.5)

    grid_map.set_value_from_polygon(ox, oy, 1.0, inside=False)

    grid_map.plot_grid_map()

    plt.axis("equal")
    plt.grid(True)

def test_position_set():
    grid_map = GridMap(100, 120, 0.5, 10.0, -0.5)

    grid_map.set_value_from_xy_pos(10.1, -1.1, 1.0)
    grid_map.set_value_from_xy_pos(10.1, -0.1, 1.0)
    grid_map.set_value_from_xy_pos(10.1, 1.1, 1.0)
    grid_map.set_value_from_xy_pos(11.1, 0.1, 1.0)
    grid_map.set_value_from_xy_pos(10.1, 0.1, 1.0)
    grid_map.set_value_from_xy_pos(9.1, 0.1, 1.0)

    grid_map.plot_grid_map()

def main():
    print("start!!")

    test_position_set()
    test_polygon_set()

    plt.show()

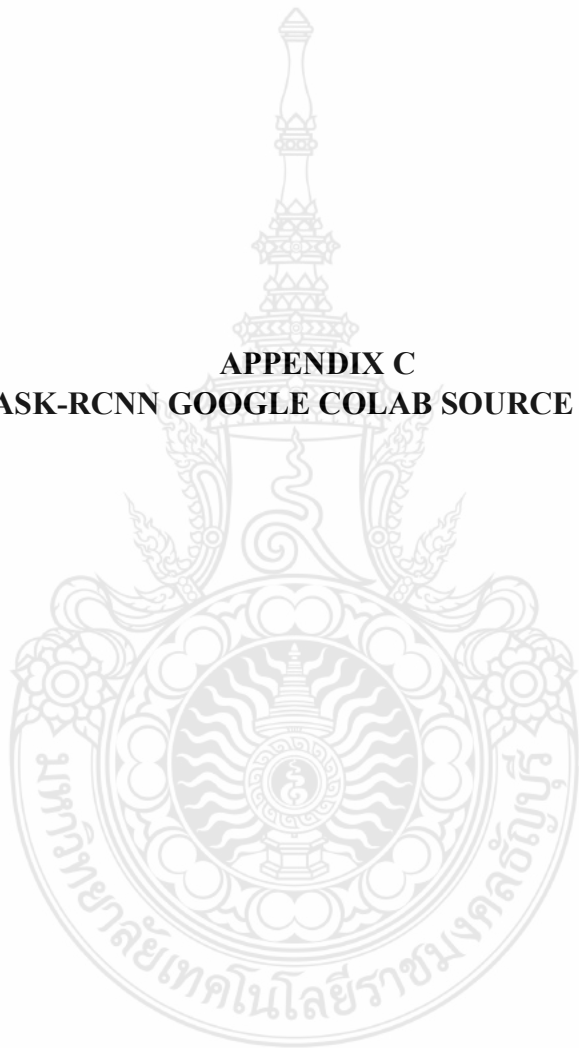
    print("done!!")

if __name__ == '__main__':
    main()

```



APPENDIX C
MASK-RCNN GOOGLE COLAB SOURCE CODE




```

colab.research.google.com/13hvy1Z9F92aRODuD4WVYt6DmqKD_xdf
low poly isometric... Jupyter Notebook... Embedded Program... 0% Scale Posited... ... Table of Contents 2...

SideWalk.ipynb
File Edit View Insert Runtime Tools Help ML Assistant...
+ Code + Text
RAM 100% CPU 100%
Editing

# Checking GPU
!nvlsim-smi

Fri Oct 28 09:35:46 2022

+-----+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.7     |
+-----+-----+-----+
| GPU Name      Persistence-M| Bus-Id        Disp-A   | Volatile Uncorr. ECC | | |
| Fan  Temp  Perf    Pwr:Usage/Cap|  Memory-Usage | GPU-Util  Compute M. |
|               |              |          |   |               |
+-----+-----+-----+
| 0  Tesla T4           Off       | 00000000:00:04:0 Off    |      0         |
| N/A   64C    P8     13W / 70W     |  0MiB / 15360MiB |    0%      Default  |
+-----+-----+-----+

Processes:
  GPU  CPU  CT  PID  Type  Process name      GPU Memory
  ID   ID                               Usage
+-----+-----+-----+
| No running processes found |
+-----+-----+-----+

# Getting access from google Drive
from google.colab import drive
drive.mount('/content/drive')
!ls -ls /content/drive/My Drive/

Mounted at /content/drive/

# cd /content/drive/MyDrive/Sidewalks
!cd /content/drive/MyDrive/Sidewalks

# Install dependencies
!pip install --upgrade kfp==2.10.0
# !wget https://pytorch.org/assets/files/torchvision-1.10.0-cp37-cp37m-manylinux1_x86_64.whl (2.9 MB)
# !tar -xvf torchvision-1.10.0-cp37-cp37m-manylinux1_x86_64.whl (15.7 MB)
Collecting kfp==2.10.0
  Downloading kfp-2.10.0-cp37-cp37m-manylinux1_x86_64.whl (2.9 MB)
  Requirement already satisfied: numpy>=1.7 in /usr/local/lib/python3.7/dist-packages (from kfp==2.10.0) (1.19.5)
  Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from kfp==2.10.0) (1.16.0)
  Installing collected packages: kfp
  Attempting uninstall: kfp
    Found existing installation: kfp 1.1.0
    Uninstalling kfp-1.1.0:
      Successfully uninstalled kfp-1.1.0
    Successfully installed kfp-2.10.0

!pip install 'kfp==2.10.0' --force-reinstall

Collecting kfp==2.10.0
  Using cached kfp-2.10.0-cp37-cp37m-manylinux1_x86_64.whl (2.9 MB)
Collecting numpy>=1.7
  Using cached numpy-1.21.5-cp37-cp37m-manylinux2.12_x86_64_musllinux_2.01_x86_64.whl (15.7 MB)
Collecting six
  Using cached six-1.16.0-py2.py3-none-any.whl (11 kb)
Installing collected packages: six, numpy, kfp
  Attempting uninstall: six
    Found existing installation: six 1.16.0
    Uninstalling six-1.16.0:
      Successfully uninstalled six-1.16.0
  Attempting uninstall: numpy
    Found existing installation: numpy 1.21.5
    Uninstalling numpy-1.21.5:
      Successfully uninstalled numpy-1.21.5
  Attempting uninstall: kfp
    Found existing installation: kfp 2.10.0
    Uninstalling kfp-2.10.0:
      Successfully uninstalled kfp-2.10.0
  Successfully installed kfp-2.10.0 numpy-1.21.5 six-1.16.0

ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This is because in the future the dependency resolver will take into account all the packages that are installed. This behavior is the result of the following dependency conflicts.
  torch 1.10.2 requires torchvision, which is not installed.
  torchvision 0.11.3 requires torchvision==0.11.3, but you have torchvision 0.11.2 which is incompatible.
  kfp==2.10.0 requires google-cloud-storage, but you have google-cloud-storage 1.39.0 which is incompatible.
  google-cloud-storage 1.39.0 requires google-auth==2.15.0, but you have google-auth 2.15.1 which is incompatible.
  google-auth 2.15.1 requires google-auth-httplib2==0.2.0, but you have google-auth-httplib2 0.2.0 which is incompatible.
  google-auth 2.15.1 requires google-auth-oauthlib==0.4.6, but you have google-auth-oauthlib 0.4.6 which is incompatible.
  Successfully installed kfp-2.10.0 numpy-1.21.5 six-1.16.0

# change location in main.py
!cat main.py

import sys
sys.path.insert(0, '/content/drive/MyDrive/1Idwz180/Mark-8794/ercc')
from ercc import *
from ercc import utils
from ercc import model as modellib
from ercc import visualize

!python3 main.py

# Ercc can edit in main.py

dataset_train = load_image_dataset(os.path.join('/content/drive/MyDrive/Sidewalks/annotations_train.json'), '/content/drive/MyDrive/Sidewalks/dataset_train')
dataset_val = load_image_dataset(os.path.join('/content/drive/MyDrive/Sidewalks/annotations_val.json'), '/content/drive/MyDrive/Sidewalks/dataset_val')
class_number = dataset_train.count_classes()

print('Train: %d' % len(dataset_train.image_ids))
print('Validation: %d' % len(dataset_val.image_ids))
print('Classes: {}'.format(class_number))


Annotation: {os.path: /content/drive/MyDrive/Sidewalks/annotations_train.json
Train: 863
Validation: 187
Classes: 1}

```



```
WARNING:tensorflow:From /tensorflow-1.15.2/python3.7/keras/backend/tensorflow_backend.py:422: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.
WARNING:tensorflow:From /tensorflow-1.15.2/python3.7/keras/backend/tensorflow_backend.py:431: The name tf.is_variable_initialized is deprecated. Please use tf.compat.v1.is_variable_initialized instead.
WARNING:tensorflow:From /tensorflow-1.15.2/python3.7/keras/backend/tensorflow_backend.py:430: The name tf.variables_initializer is deprecated. Please use tf.compat.v1.variables_initializer instead.
WARNING:tensorflow:From /tensorflow-1.15.2/python3.7/keras/callbacks/tensorboard_v1.py:200: The name tf.summary.merge_all is deprecated. Please use tf.compat.v1.summary.merge_all instead.
WARNING:tensorflow:From /tensorflow-1.15.2/python3.7/keras/callbacks/tensorboard_v1.py:200: The name tf.summary.FileWriter is deprecated. Please use tf.compat.v1.summary.FileWriter instead.
Epoch 1/5
500/500 [=====] - 109s 2s/step - loss: 0.0432 - val_loss: 0.1160
WARNING:tensorflow:From /tensorflow-1.15.2/python3.7/keras/callbacks/tensorboard_v1.py:144: The name tf.Summary is deprecated. Please use tf.compat.v1.Summary instead.
Epoch 2/5
500/500 [=====] - 1020s 2s/step - loss: 0.4994 - val_loss: 0.5034
Epoch 3/5
500/500 [=====] - 1010s 2s/step - loss: 0.6297 - val_loss: 0.3626
Epoch 4/5
500/500 [=====] - 1011s 2s/step - loss: 0.5654 - val_loss: 0.3045
Epoch 5/5
500/500 [=====] - 1011s 2s/step - loss: 0.3359 - val_loss: 0.2063

| | test_model, inference_config = load_test_model(1)
| |
| | csrcnn_model.PushRCNN object at 0x7f79facd4210:
| | /content/gdrive/MyDrive/S10waks/Mask_RCNN/logs/object302302371246/mask_rcnn_object_0005.h5
| | Loading weights from task /content/gdrive/MyDrive/S10waks/Mask_RCNN/logs/object302302371246/mask_rcnn_object_0005.h5
| | Re-starting from epoch 5.

Test random_image(test_model, dataset_val, inference_config)
original_image shape: (512, 512, 3) min: 0.00000 max: 255.00000 dtype: float32
Trained model results:
Processing 1 images
image shape: (512, 512, 3) min: 0.00000 max: 255.00000 dtype: float32
masked_image shape: (1, 512, 512, 3) min: -127.74000 max: 131.21000 dtype: float64
image_mask shape: (1, 14) min: 0.00000 max: 512.00000 dtype: float64
anchors shape: (1, 5672, 4) min: -0.70049 max: 1.58325 dtype: float32
WARNING:tensorflow:From /tensorflow-1.15.2/python3.7/keras/backend/tensorflow_backend.py:431: The name tf.is_variable_initialized is deprecated. Please use tf.compat.v1.is_variable_initialized instead.
WARNING:tensorflow:From /tensorflow-1.15.2/python3.7/keras/backend/tensorflow_backend.py:430: The name tf.variables_initializer is deprecated. Please use tf.compat.v1.variables_initializer instead.
1/1 [=====] - 5s 5s/step
Annotation

```

APPENDIX D
INTERNATIONAL PUBLICATIONS



Crisnapati PN, Maneetham D. RIFIS: A Novel Rice Field Sidewalk Detection Dataset for Walk-Behind Hand Tractor. Data-MDPI. 2022 Sep 25;7(10):135. <https://doi.org/10.3390/data7100135> (Scopus Q2)

Crisnapati PN, Maneetham D, Triandini E. Trolls: a novel low-cost controlling system platform for walk-behind tractor. International Journal of Electrical and Computer Engineering (IJECE). 2023 Feb 1;13(1):842. <http://doi.org/10.11591/ijece.v13i1.pp842-858> (Scopus Q2)

Crisnapati PN, Maneetham D, Triandini E. Eleven Degree of Freedom Humanoid Upper Body Robot SIBO. Information Technology International Seminar (ITIS 2022) IEEE (Scopus)



Biography



Name – Surname	Padma Nyoman Crisnapati
Date of Birth	January 17, 1988
Address	39 Rangsit, Thanyaburi District, Chang Wat Pathum Thani 12110
Education	<p>Doctoral Degree in Mechatronics Engineering (D. Eng.) – Rajamangala University of Technology Thanyaburi, from 2020- 2023. GPA: 4,00.</p> <p>Master Degree in Computer Science (M. Kom.) - Ganesha University of Education, from 2016 – 2018. GPA: 3,88.</p> <p>Master Degree in Learning Technologies (M. Pd.) - Ganesha University of Education, from 2009 – 2011. GPA: 3,52.</p> <p>Bachelor Degree in Informatics Engineering (S. Kom.) - Sepuluh Nopember Institute of Technology, from 2005 - 2009. GPA: 3,12.</p>
Experience Work	<p>ITB STIKOM Bali Lecturer (2014- 2020)</p> <p>ITB STIKOM Bali Head Department (2016- 2020)</p> <p>Ganesha University of Education Lecturer (2009-2015)</p> <p>Denpasar Industrial Training Center Competency Assessor (2015-2020)</p>
Telephone Number	+66991857582/ +6287762999652
Email Address	crisnapati@gmail.com padma_c@mail.rmutt.ac.th