

Article

Fuzzy Control of Self-Balancing, Two-Wheel-Driven, SLAM-Based, Unmanned System for Agriculture 4.0 Applications

János Simon 

Department of Mechatronics and Automation, Faculty of Engineering, University of Szeged,
6720 Szeged, Hungary; simon@mk.u-szeged.hu

Abstract: This article presents a study on the fuzzy control of self-balancing, two-wheel-driven, simultaneous localization and mapping (SLAM)-based, unmanned systems for Agriculture 4.0 applications. The background highlights the need for precise and efficient navigation of unmanned vehicles in the field of agriculture. The purpose of this study is to develop a fuzzy control system that can enable self-balancing and accurate movement of unmanned vehicles in various terrains. The methods employed in this study include the design of a fuzzy control system and its implementation in a self-balancing, two-wheel-driven, SLAM-based, unmanned system. The main findings of the study show that the proposed fuzzy control system is effective in achieving accurate and stable movement of the unmanned system. The conclusions drawn from the study indicate that the use of fuzzy control systems can enhance the performance of unmanned systems in Agriculture 4.0 applications by enabling precise and efficient navigation. This study has significant implications for the development of autonomous agricultural systems, which can greatly improve efficiency and productivity in the agricultural sector. Fuzzy control was chosen due to its ability to handle uncertainty and imprecision in real-world applications.

Keywords: fuzzy control; self-balancing; SLAM; unmanned systems; Agriculture 4.0



Citation: Simon, J. Fuzzy Control of Self-Balancing, Two-Wheel-Driven, SLAM-Based, Unmanned System for Agriculture 4.0 Applications. *Machines* **2023**, *11*, 467. <https://doi.org/10.3390/machines11040467>

Academic Editor: Ning Sun

Received: 27 March 2023

Revised: 3 April 2023

Accepted: 6 April 2023

Published: 10 April 2023



Copyright: © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Agriculture 4.0 is an emerging field that aims to transform traditional farming practices into more sustainable and efficient systems by incorporating digital technologies. One of the key challenges in this field is the development of unmanned systems that can navigate autonomously and perform tasks such as crop monitoring, pesticide application, and yield estimation. However, designing an autonomous system that can move in a complex and dynamic environment such as a greenhouse is a challenging task that requires the integration of various technologies, including sensors, control systems, and machine learning algorithms. This paper focuses on the development of a self-balancing two-wheel driven unmanned system for Agriculture 4.0 applications. The system is based on SLAM algorithms, which enable the system to create a map of the environment and localize itself within it. The system is equipped with a set of sensors, including a sonar sensor, an IMU, a BME 280 sensor, and a camera, which provide data on the position, orientation, and surroundings of the system. To control the movement of the unmanned system, a fuzzy logic control system was implemented. Fuzzy logic is a powerful tool for controlling complex and nonlinear systems because it allows for the representation of imprecise or vague information. The fuzzy logic controller takes the position and orientation data provided by the sensors as input and generates a control signal to adjust the speed and direction of the system's motors as shown in Figure 1. The self-balancing feature of the system ensures that the system remains stable even on uneven terrain. The proposed system has several advantages over traditional agricultural machinery. First, it can navigate autonomously, reducing the need for human intervention and increasing efficiency. Second, it can operate in areas that are difficult or impossible for traditional

machinery to access, such as steep slopes or narrow rows. Third, it can perform tasks such as crop monitoring and yield estimation with high precision, enabling farmers to make more informed decisions about their crops. The development of autonomous unmanned systems for Agriculture 4.0 applications is a rapidly evolving field, and there is still much to be explored. However, this paper represents an important step forward in the development of these systems, and the results presented in this paper demonstrate the feasibility and effectiveness of the proposed approach. By combining SLAM algorithms, fuzzy logic control, and self-balancing mechanisms, the proposed system represents a promising direction for the development of unmanned systems for Agriculture 4.0 applications.

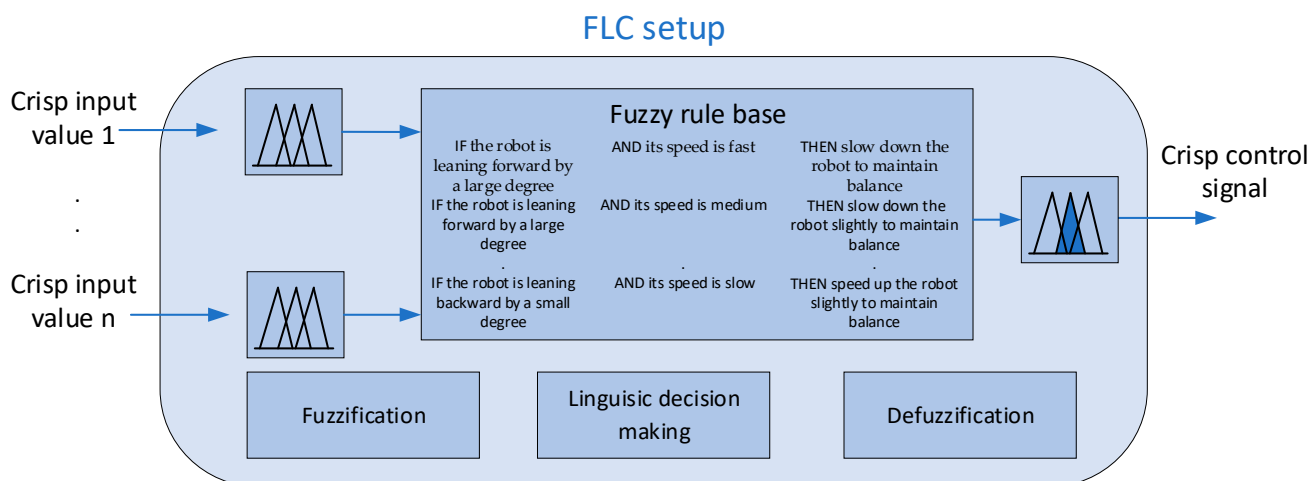


Figure 1. The fuzzy logic controller and its key components.

Two-wheel-driven, SLAM-based, unmanned systems equipped with sensors and navigation algorithms can help farmers make data-driven decisions. They can monitor crop health, detect diseases and pests, and apply fertilizers and pesticides with high precision, reducing waste and increasing yield. These systems can replace human labor in tedious and repetitive tasks, such as monitoring crop parameters, detecting diseases, and measuring environmental data. This can significantly reduce labor costs and allow farmers to focus on more skilled tasks. Autonomous mobile robots can optimize the use of resources such as water, fertilizers, and pesticides. They can detect soil moisture levels and adjust irrigation systems accordingly, reducing water waste. They can also apply fertilizers and pesticides in a targeted manner, reducing overuse and environmental impacts. They can also perform various tasks in hazardous environments, such as spraying pesticides or inspecting crops in areas with high levels of radiation or toxins. This can improve worker safety and reduce health risks. These systems can operate 24/7 and cover large areas of farmland or in greenhouses efficiently. They can work in adverse weather conditions, such as heavy rain or extreme heat, and operate in low-light conditions, such as during nighttime. Fuzzy control was chosen due to its ability to handle uncertainty and imprecision in real-world applications. The purpose of this study is to demonstrate the effectiveness of the proposed fuzzy control system in achieving accurate and stable movement of unmanned systems. The results of this study have significant implications for the development of autonomous agricultural systems, which can greatly improve efficiency and productivity in the agricultural sector.

The main contributions of this research are summarized as follows:

- A fuzzy based controller was developed and tested for two-wheel-driven unmanned system.
- A novel SLAM algorithm was developed and tested in a greenhouse environment.
- A LabVIEW—Vision-based application was developed to grade and assess the ripeness of the greenhouse fruits, in this case tomatoes.

The paper is organized as follows. Section 1.1 describes the related work. Section 2 presents the materials and method, followed by the experiment and results in Section 3. Finally, the discussion and conclusions are presented in Sections 4 and 5.

1.1. Related Work

In this subsection, the existing research on unmanned vehicles in agriculture is discussed, including the challenges associated with precise and efficient navigation in dynamic environments. Yang and colleagues [1] investigated an automatic motion control system for a type of wheeled inverted pendulum model commonly used to represent two-wheeled modern vehicles. Another study [2] proposed a unique approach for a Mecanum-wheeled mobile robot equipped with a cylinder that can simultaneously balance and track. The researchers first used the Newton–Euler approach to mathematically model the robot's motion. In a separate study [3], the authors presented an adaptive hierarchical sliding mode control system based on fuzzy neural networks to achieve high-precision trajectory tracking for underactuated systems. The system was viewed as several subsystems. Additionally, researchers in [4] presented Magicol, a system for indoor localization and tracking that accounts for local disturbances in the geomagnetic field by vectorizing consecutive magnetic signals and using them to shape the particle distribution in the estimation process. In another study [5], the authors analyzed the statistical properties of magnetic field (MF) measurements obtained from various smartphones in indoor environments. They found that in the absence of disturbances, MF measurements follow a Gaussian distribution with temporal stability and spatial discernibility. The aim of the research in [6] was to explore the differences that arise due to different technologies operating in an indoor space. The proposed method was validated using training and test data collected in a laboratory. In [7], the researchers proposed a novel fingerprinting-based indoor 2D positioning method for mobile robots that fused RSSI and magnetometer measurements and utilized multilayer perceptron feedforward neural networks to determine the 2D position based on both sets of data. Finally, in [8], the authors presented an improved SLAM algorithm that uses wheel encoder data from an autonomous ground vehicle to achieve robust performance in a featureless tunnel environment. The improved SLAM system integrates the wheel encoder sensor data into the FAST-LIO2 LiDAR SLAM structure using the extended Kalman filter algorithm. In [9], the study investigated the fruit and vegetable classification problem, which is an important aspect of image recognition. The authors suggested that GoogLeNet provides an optimal solution for this classification problem. In [10], the authors proposed an A* guiding deep Q-network (AG-DQN) algorithm to solve the pathfinding problem of automated guided vehicles (AGVs) in a robotic mobile fulfillment system (RMFS). The RMFS is a parts-to-picker storage system that replaces manual labor with AGVs, enhancing the efficiency of picking work in warehouses. In [11], the area of mobile robotics was examined with a focus on new development trends. These developments are based on artificial intelligence, autonomous driving, network communication, cooperative work, human–robot interfaces, interactions, emotional expressions, and perceptions. Furthermore, these trends are being applied to diverse fields such as precision agriculture, healthcare, sports, ergonomics, industry, smart transport, and service robotics. In [12], the authors presented a classification of the literature on wheeled mobile manipulation, taking into account its diversity. They discussed the interplay between deployment tasks, application arenas, and decision-making methodologies, with a view to identify future research opportunities. The paper [13] introduced a novel methodology for managing energy in systems that involve Photovoltaic (PV)/Battery/Fuel Cells (FCs). The approach employs Immersion and Invariance (I&I) theorem-based optimization rules and a proposed compensator utilizing deep learning type-2 fuzzy logic to address uncertainties. The study [14] introduced an innovative approach to observer-based fuzzy control of chaotic systems (CSs), specifically targeting a class of CSs with unknown dynamics, unknown input constraints, and unmeasurable states. To address the uncertainties, the study formulated a generalized type-2 (GT2) fuzzy logic system (FLS) for approximation purposes. Overall, this work demon-

strated a clear gap in the literature for effective control systems for unmanned vehicles in agriculture, and argue that fuzzy control is a promising approach to address this gap. By developing and testing a fuzzy control system for self-balancing, two-wheel-driven, SLAM-based unmanned systems, this research aimed to contribute to the development of autonomous agricultural systems that can improve efficiency and productivity in the agricultural sector.

2. Materials and Methods

The present study aims to develop a fuzzy control algorithm for self-balancing, two-wheel-driven, SLAM-based, unmanned systems for Agriculture 4.0 applications. The proposed control algorithm will allow the unmanned system to perform autonomous navigation and mapping tasks while maintaining a stable and balanced posture. To achieve this goal, a custom-built robotic platform based on the ESP32 microcontroller and various sensors, such as an MPU 6050 inertial measurement unit (IMU), two geared motors, and a camera module were used [3]. The robotic platform was also equipped with a motor driver and a voltage regulator to control the speed of the motors and power the sensors as depicted in Figure 2.

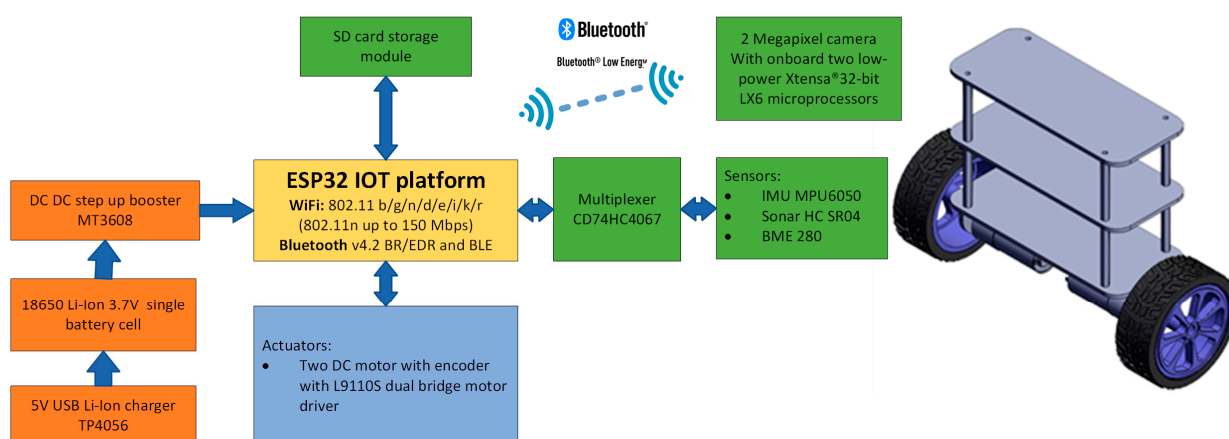


Figure 2. The block scheme and 3D CAD model of the self-balancing, two-wheel, mobile robot.

The control algorithm was designed using the Mamdani-type fuzzy logic controller (FLC) with three inputs: error angle, error rate, and acceleration. The outputs of the FLC were two control signals to adjust the speed of the two motors. The input values were acquired from the IMU, and the camera module was used for image processing to identify the environment and obstacles for the unmanned system. The proposed algorithm was tested in a real-time environment with different types of terrains, such as slopes and uneven surfaces, and the performance was evaluated based on various parameters such as stability, accuracy, and response time [15]. Overall, the proposed fuzzy control algorithm provides a reliable and efficient solution for self-balancing, two-wheel-driven, SLAM-based, unmanned systems for Agriculture 4.0 applications, which could potentially revolutionize the agriculture industry by providing a cost-effective and efficient solution for crop monitoring and management. The hardware connection relation of each module is shown in Figure 2.

The self-balancing, two-wheel, mobile robot is a complex system that can be modeled using either a transfer function or state space model [16]. A transfer function is a mathematical representation of the relationship between the input and output of a system in the frequency domain. It is expressed as the ratio of the output to the input in the Laplace domain. The transfer function of the self-balancing, two-wheel, mobile robot can

be derived using the principle of conservation of angular momentum. The transfer function is explained in [3] and can be expressed as:

$$G(s) = (K_v/R) / (s^2 + (K_v/R)s + g/L) \quad (1)$$

where s is the Laplace variable, K_v is the motor constant, R is the motor resistance, g is the gravitational acceleration, and L is the distance between the two wheels.

On the other hand, the state space model of the self-balancing, two-wheel, mobile robot describes the behavior of the system using a set of first-order differential equations. It is a mathematical model that represents the system in terms of its state variables and input variables. The state space model of the self-balancing, two-wheel, mobile robot can be expressed as:

$$\dot{x} = Ax + Bu \quad (2)$$

$$y = Cx + Du \quad (3)$$

where x is the state vector, u is the input vector, y is the output vector, A is the state matrix, B is the input matrix, C is the output matrix, and D is the direct transmission matrix. The state vector of the self-balancing, two-wheel, mobile robot can be expressed as:

$$x = [\theta, \dot{\theta}, \varphi, \dot{\varphi}]^T \quad (4)$$

where θ is the angle of the robot's body with respect to the vertical, $\dot{\theta}$ is the angular velocity of the robot's body, φ is the angle of the robot's wheels with respect to the vertical, and $\dot{\varphi}$ is the angular velocity of the robot's wheels. The input vector of the self-balancing, two-wheel, mobile robot can be expressed as:

$$u = V \quad (5)$$

where V is the voltage applied to the motor. The output vector of the self-balancing, two-wheel, mobile robot can be expressed as:

$$y = [\theta, \varphi]^T \quad (6)$$

where θ and φ are the angles of the robot's body and wheels with respect to the vertical, respectively. The state matrix, input matrix, output matrix, and direct transmission matrix of the self-balancing, two-wheel, mobile robot can be derived based on the dynamics of the model [17]. The state space model can be applied to design a controller for the self-balancing, two-wheel, mobile robot, which can be applied to stabilize the system and control its motion.

2.1. Transfer Function

The Laplace transformation must be taken of the model equations assuming zero initial values in order to derive the needed transfer functions of the linearized system equations. The obtained Laplace transforms [18] are as follows:

$$(I + ml^2)\Phi(s)s^2 - mgl\Phi(s) = mlX(s)s^2 \quad (7)$$

$$(M + m)X(s)s^2 + bX(s)s - ml\Phi(s)s^2 = U(s) \quad (8)$$

Recall that a transfer function is a representation of the relationship between a single input and a single output at a time. To obtain the first transfer function for the output $\Phi(s)$

and an input of $U(s)$, an elimination of $X(s)$ is needed from the above equations. The first equation for $X(s)$ can be solved as follows.

$$X(s) = \left[\frac{I + ml^2}{ml} - \frac{g}{s^2} \right] \Phi(s) \quad (9)$$

Then, substitute the above into the second equation.

$$(M + m) \left[\frac{I + ml^2}{ml} - \frac{g}{s^2} \right] \Phi(s) s^2 + b \left[\frac{I + ml^2}{ml} - \frac{g}{s^2} \right] \Phi(s) s - ml \Phi(s) s^2 = U(s) \quad (10)$$

By rearranging the transfer function, the following form is defined:

$$\frac{\Phi(s)}{U(s)} = \frac{\frac{ml}{q} s^2}{s^4 + \frac{b(I+ml^2)}{q} s^3 - \frac{(M+m)mgI}{q} s^2 - \frac{bmgl}{q} s} \quad (11)$$

where,

$$q = [(M + m)(I + ml^2) - (ml)^2] \quad (12)$$

It is evident from the transfer function above that the origin contains both a pole and a zero. When they are cancelled, the transfer function changes to [19]:

$$P_{pend}(s) = \frac{\Phi(s)}{U(s)} = \frac{\frac{ml}{q} s}{s^3 + \frac{b(I+ml^2)}{q} s^2 - \frac{(M+m)mgI}{q} s - \frac{bmgl}{q}} \left[\frac{\text{rad}}{N} \right] \quad (13)$$

Second, the transfer function with the robot position $X(s)$ as the output can be derived in a similar manner to get the following:

$$P_{cart}(s) = \frac{X(s)}{U(s)} = \frac{\frac{(I+ml^2)s^2 - gml}{q}}{s^4 + \frac{b(I+ml^2)}{q} s^3 - \frac{(M+m)mgI}{q} s^2 - \frac{bmgl}{q} s} \left[\frac{m}{N} \right] \quad (14)$$

2.2. State-Space Model

If the aforementioned linearized equations of motion are reorganized into a set of first order differential equations, they can also be expressed in state-space model. The equations can then be transformed into the common matrix form, as illustrated below, because they are linear equations [18].

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgI(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u \quad (15)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \quad (16)$$

Since both the robot's position and the pendulum's position are part of the output, the C matrix will have two rows. Specifically, the robot's position is the first element of the output y and the deviation of the pendulum from its equilibrium point is the second element of y.

3. Results

The research results point to the need for such systems in the field of precision agriculture. This research includes multiple disciplinary areas. Applying modern development environments leads to clear results in the field of mobile robotics.

3.1. Fuzzy Control of Self-Balancing, Two-Wheel Robot

Fuzzy control is a method that uses fuzzy logic to control a system, accounting for uncertainty in a systematic way [20]. When applied to a self-balancing, two-wheel, Segway-type robot, fuzzy control can be used to adjust the robot's speed and direction to keep it upright and stable. To implement fuzzy control of a self-balancing, two-wheel, Segway-type robot in LabVIEW, the following steps can be acquired as depicted on Figure 3.

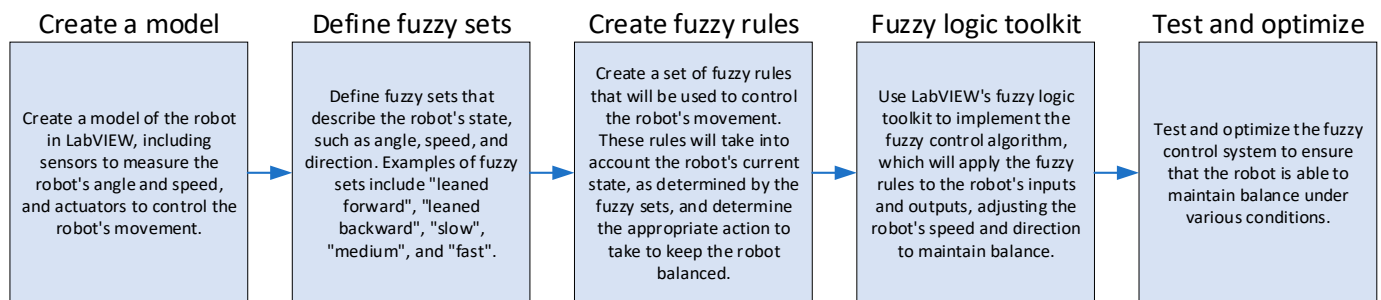


Figure 3. Steps of self-balancing two-wheel mobile robot fuzzy control development.

In the case of a self-balancing, two-wheel, Segway-type robot, the fuzzy rule base can be used to adjust the robot's speed and direction to maintain balance. Table 1 shows a full fuzzy rule base for controlling a self-balancing, two-wheel, Segway-type robot.

Table 1. Fuzzy rules of the robot.

IF	AND	THEN
If the robot is leaning forward by a large degree	its speed is fast	slow down the robot to maintain balance
If the robot is leaning forward by a large degree	its speed is medium	slow down the robot slightly to maintain balance
If the robot is leaning forward by a small degree	its speed is fast	maintain the current speed to maintain balance
If the robot is leaning forward by a small degree	its speed is medium	maintain the current speed to maintain balance
If the robot is leaning forward by a small degree	its speed is slow	speed up the robot slightly to maintain balance
If the robot is leaning backward by a large degree	its speed is fast	slow down the robot significantly to maintain balance
If the robot is leaning backward by a large degree	its speed is medium	slow down the robot moderately to maintain balance
If the robot is leaning backward by a small degree	its speed is fast	speed up the robot slightly to maintain balance
If the robot is leaning backward by a small degree	its speed is medium	maintain the current speed to maintain balance
If the robot is leaning backward by a small degree	its speed is slow	speed up the robot slightly to maintain balance

These fuzzy rules take into account the robot's angle and speed, and determine the appropriate action to take to maintain balance. By combining these rules with a fuzzy logic controller, the robot can be controlled in a way that adapts to changing conditions and maintains balance even in the presence of uncertainty. The center of gravity method is used to solve the fuzzy crisp output value, as shown in Equation (17):

$$\eta = \frac{\sum \mu_i \cdot f(\mu_i)}{\sum f(\mu_i)} \quad (17)$$

The implemented state-space model in LabVIEW environment is shown in the Figure 4 as a "State-Space" block from the "Control Design and Simulation" palette, with the state

space matrices A, B, C, and D wired to its inputs. The A matrix represents the dynamics of the system in the absence of any inputs. Its first row represents the rate of change of the angular position of the pendulum, which is related to its angular velocity. The B matrix represents how the system responds to inputs. The C matrix represents how the system produces outputs. The D matrix represents the steady-state gain of the system, which is always zero for an inverted pendulum system. These matrices are the standard notation used to represent a linear time-invariant system in state-space form and can be used to simulate and control the behavior of an inverted pendulum system using a variety of techniques, such as state feedback, observer design, and fuzzy control.

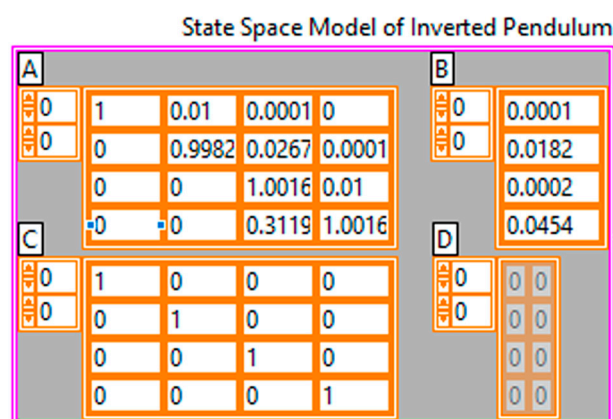


Figure 4. State-space model of the self-balancing, two-wheel-driven, mobile robot in LabVIEW block diagram.

A self-balancing, two-wheel-driven, Segway-type mobile robot can be controlled using LabVIEW by implementing a control loop that takes inputs from the robot's sensors and adjusts the motors accordingly to maintain balance as depicted in Figure 5.

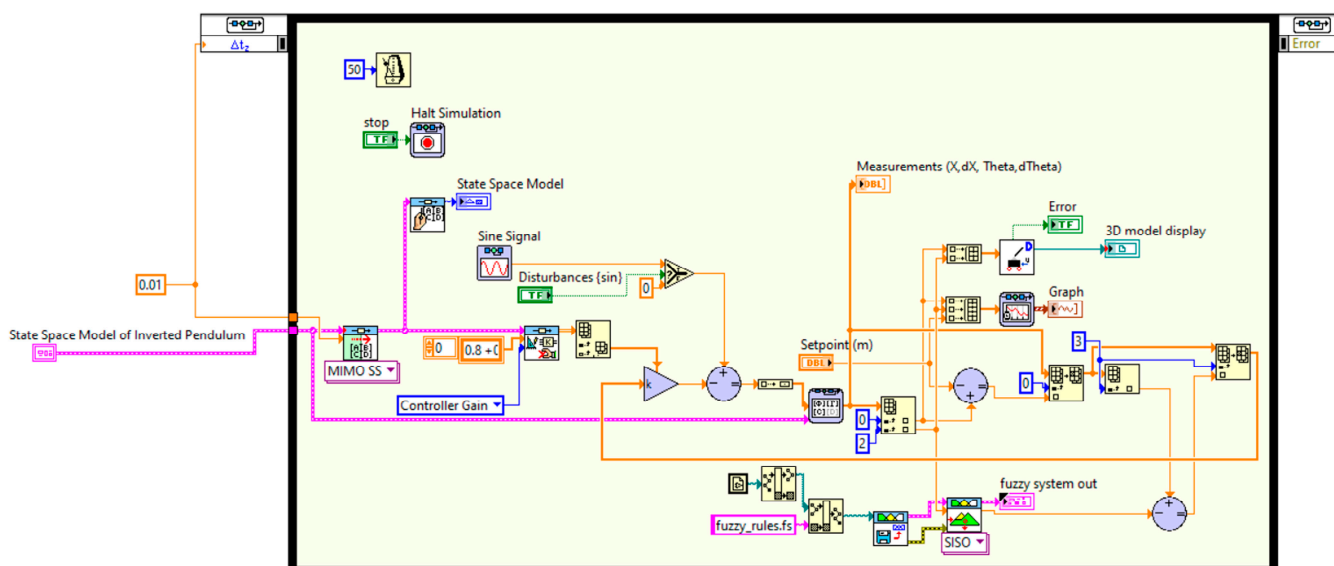


Figure 5. Block diagram of the fuzzy logic-based controller in LabVIEW.

This LabVIEW program provides a basic control loop for a self-balancing, two-wheel-driven, Segway-type mobile robot as depicted on Figure 6. With additional sensors and control algorithms, the program can be further improved to achieve more advanced behaviors, such as turning and maneuvering in different directions. In Figure 7, a solution of using a VRML model is presented in LabVIEW to display a real-time movement of the two-wheel mobile robot based on the input parameters and the controllers output. The

front panel in Figure 6 shows the real-time movement of the two-wheel mobile robot along with the control parameters and the view angle that can be changed.

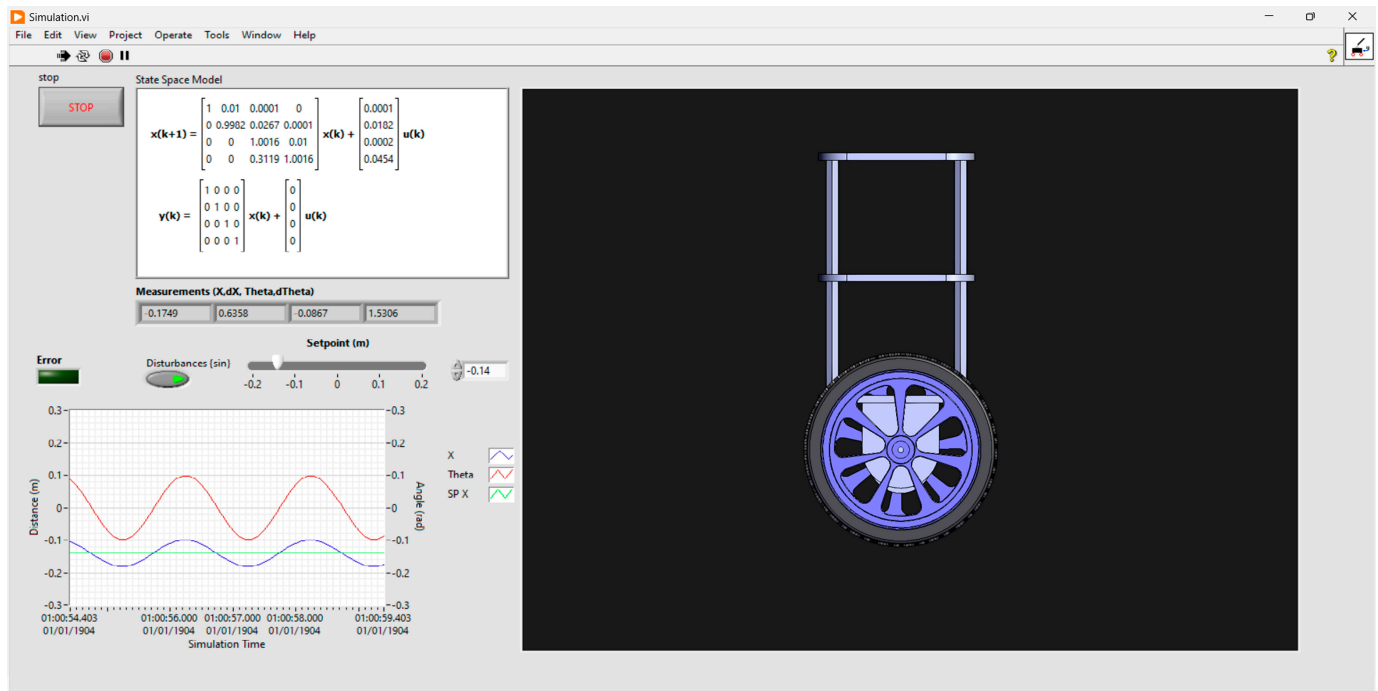


Figure 6. Front panel of the fuzzy logic-based controller with 3D model preview.

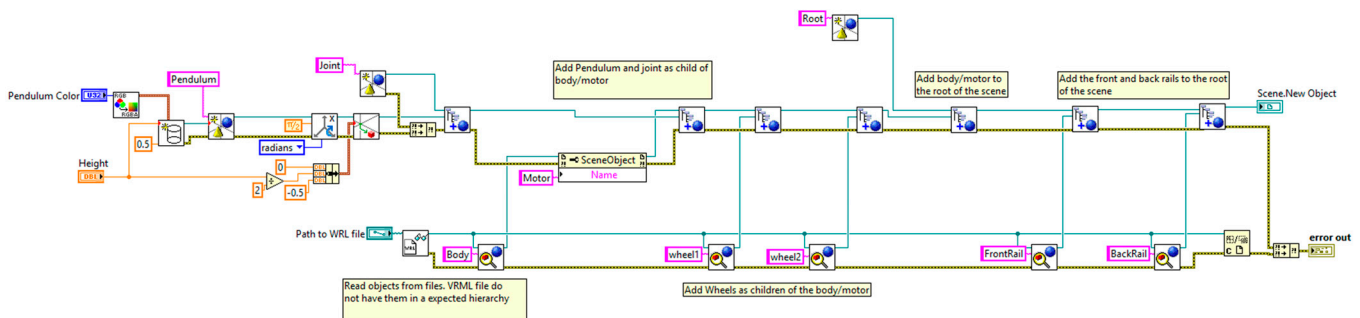


Figure 7. Block diagram of VRML model-based real-time model preview.

A real-time simulation of the segway robot using LabVIEW can be performed by modeling the robot in a 3D modeling software and exporting the model in VRML format [11]. The model can then be imported into LabVIEW using the VRML object in the User Interface Palette as can be seen in Figure 7.

The most of the relevant parameters of the two-wheel mobile robot model can be found in Table 2.

Table 2. The main components and models of the robot.

Module	Model
Chassis	Custom-built aluminum and 3D-printed PLA
Motor	DC motor
Motor driver	L9110S dual bridge
Controller	ESP32s
Sonar	HC-SR04
IMU	MPU6050
Temp./RH/Pressure sensor	BME280
Camera	ESP32 Cam

The elements and modules were chosen in such a way that they are easy to obtain and are not particularly expensive. In the design of the two-wheel mobile robot chassis, the performance of the structure directly determines the robot's moving and steering mode.

3.2. SLAM

This problem deals with the possibility of a robot autonomously starting its movement in an unknown environment, to incrementally, while moving, build a map of that space and to simultaneously determine its absolute position based on this map as can be seen on Figure 8. As the SLAM problem is considered when the robot moves in completely or partially unknown environments, then the solution of this problem is much more difficult [21]. Of all the proposed solutions to this problem so far, one of the most successful in the form of the Extended Kalman Filter (EKF) can be singled out.

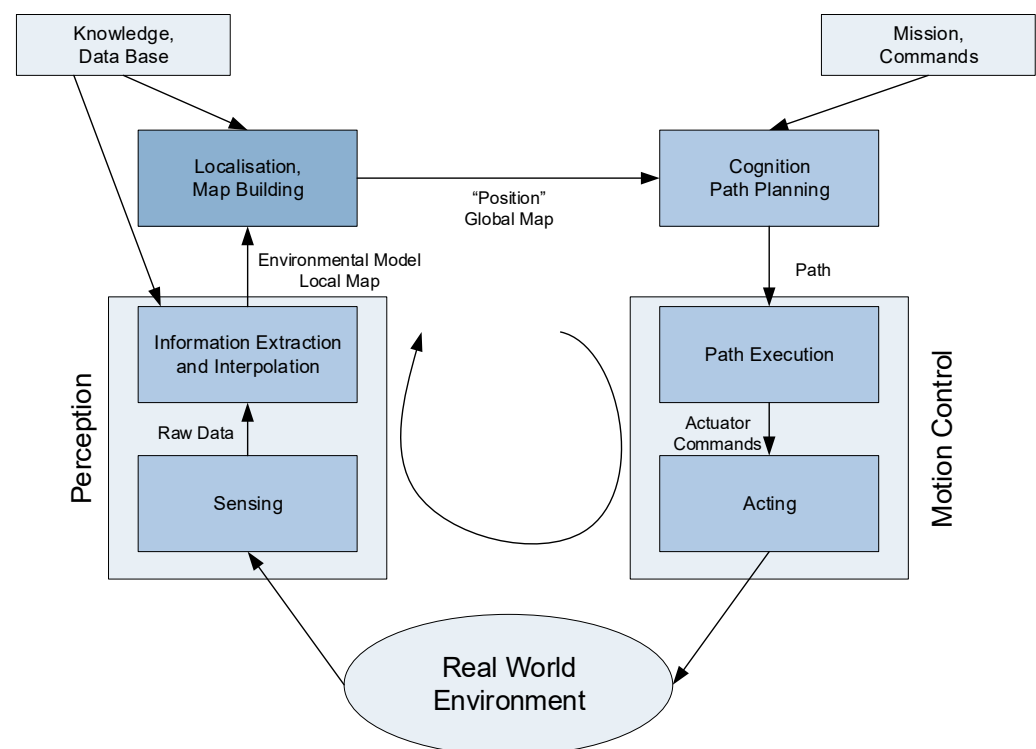


Figure 8. Simplified SLAM-based control scheme of the two-wheel mobile robot.

Ultrasonic sensors (or sonars) are distance sensors. They measure distance by emitting a short-duration sound signal at an ultrasonic frequency, and measure the time from when the signal is emitted until the reflected wave (echo) returns back to the sensor. The measured time is proportional to twice the distance to the nearest obstacle in the range of the sensor, and from this, the distance to the nearest obstacle is easily obtained. The two-wheel mobile robot is equipped with three ultrasonic sensors type HC-SR04 (left, right, front) to support SLAM.

Assuming that the mobile robot moves in some closed space, in which it is possible to determine the absolute position of the robot in the whole or most often in some parts of the space with external vision sensors, the solution to the problem of localization and space mapping can be greatly facilitated. However, information only about the robot's location is not sufficient in most applications [22]. If external sensors that can determine the absolute location of the robot are used, the location of the robot in relation to the reference coordinate system can be obtained, but that information does not define the environment in which the robot is located, which, especially if inhabited by people, can be dynamic and changeable. In addition, the mobile robot may be obscured in some positions, so control based on external sensors is impossible. If several robots are present in the environment

and they can determine their relative position in relation to other robots, then based on that information and the information received from external sensors, the absolute position of a robot that is not visible by external sensors can be determined.

Navigation, in the context of mobile robots, can be considered a combination of three fundamental operations of mobile robots: localization, path planning, and map creation, i.e., interpretation. Since localization and mapping have already been previously processed, this part will give an overview of algorithms for path planning. There are two “types” of navigation: global and local. Global navigation is considered to be navigation in a known space (that is, space for which there is a prepared map). With global navigation, the robot can, using path planning algorithms (e.g., Dijkstra, A*), plan in advance the path to the goal without colliding with obstacles [23]. On the other hand, local navigation has no knowledge of the space in which the robot moves and is therefore limited to a small space around the robot. By using local navigation, the robot usually only avoids obstacles that it detects using sensors. In most applications, global and local navigation are used together, where the global navigation algorithm determines the optimal path for the robot to reach its destination, and the local navigation guides it there, avoiding obstacles that appear and are not visible on the map. The two-wheel mobile robot is controlled to start from the starting point as shown in Figure 9 and start to build the map.

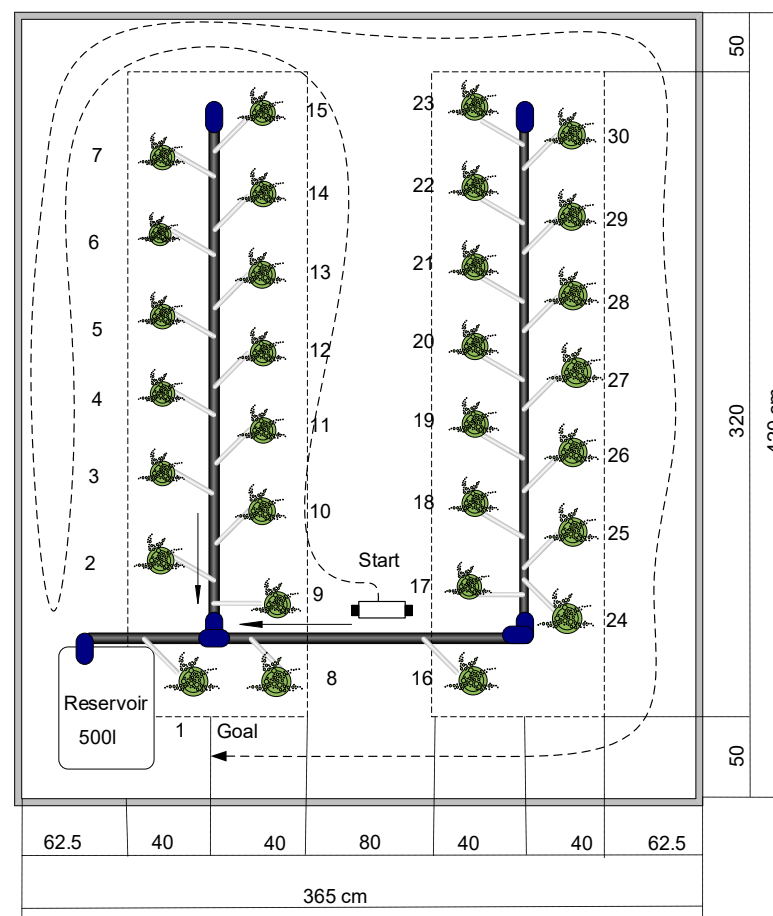


Figure 9. Navigation strategy of the two-wheel mobile robot in a greenhouse environment.

The methods of controlling mobile robots can be classified into three categories based on the method of collecting information about the position and the environment:

- Control based on information obtained from internal sensors on the robot;
- Control based on information obtained from external sensors, where robots are controlled by a computer that represents the brain of all robots;

- Control based on information obtained from internal and external sensors, which represents a hybrid solution of the previous two solutions, and includes all the advantages of both.

3.3. Computer Vision and Classification

Vision is probably the “most powerful” human sense, which is rich in information about the outside world, so appropriate sensors were developed to imitate human vision for machines. Vision sensors are digital and video cameras, which, apart from mobile robots, are also used in various other everyday applications. The interpretation of what the robot sees using the cameras, i.e., extracting information from the image captured by the camera, is obtained by processing and analyzing the images [24]. Computer vision and classification of green and red fruit in a greenhouse environment using a self-balancing, two-wheel-driven, mobile robot can be a very useful application in Agriculture 4.0. The robot can be equipped with a camera and a computer vision algorithm to identify and classify the color of the fruits as depicted in Figure 10 [25].

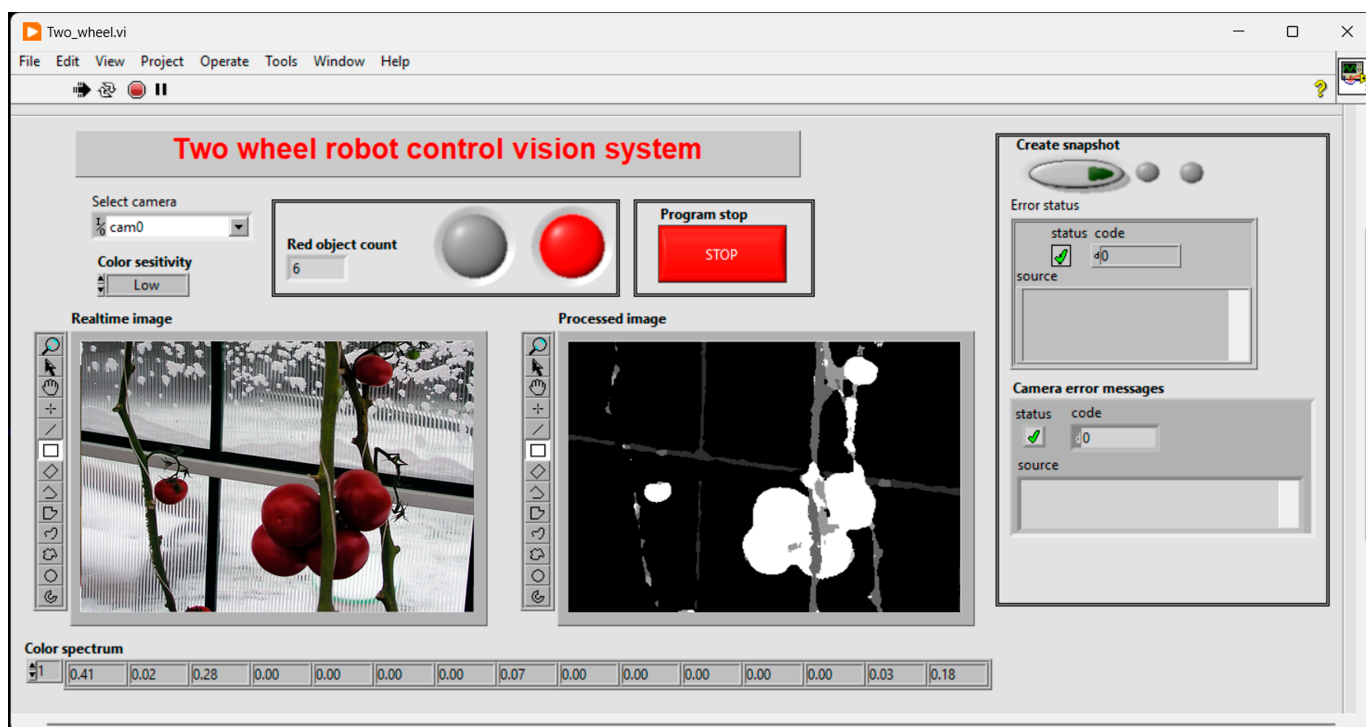


Figure 10. Two-wheel robot vision/classification front panel.

The algorithm can be trained using machine learning techniques to accurately detect and classify green and red fruits. The robot can then move around the greenhouse, scanning and classifying fruits on its way. The robot can also collect data on the location and number of fruits, which can be used to optimize the harvesting process [25]. The self-balancing, two-wheel-driven, mobile robot can be controlled using a remote or an autonomous system. With the autonomous system, the robot can move around the greenhouse on its own, without any human intervention. This can save time and increase efficiency in the harvesting process [26]. Overall, the application of computer vision and classification of green and red fruit using a self-balancing, two-wheel-driven, mobile robot can be a very useful and innovative solution for Agriculture 4.0. It can help farmers to optimize the harvesting process, reduce labor costs, and increase yields. Figure 11 shows the Vision subsystem implemented in the LabVIEW environment.

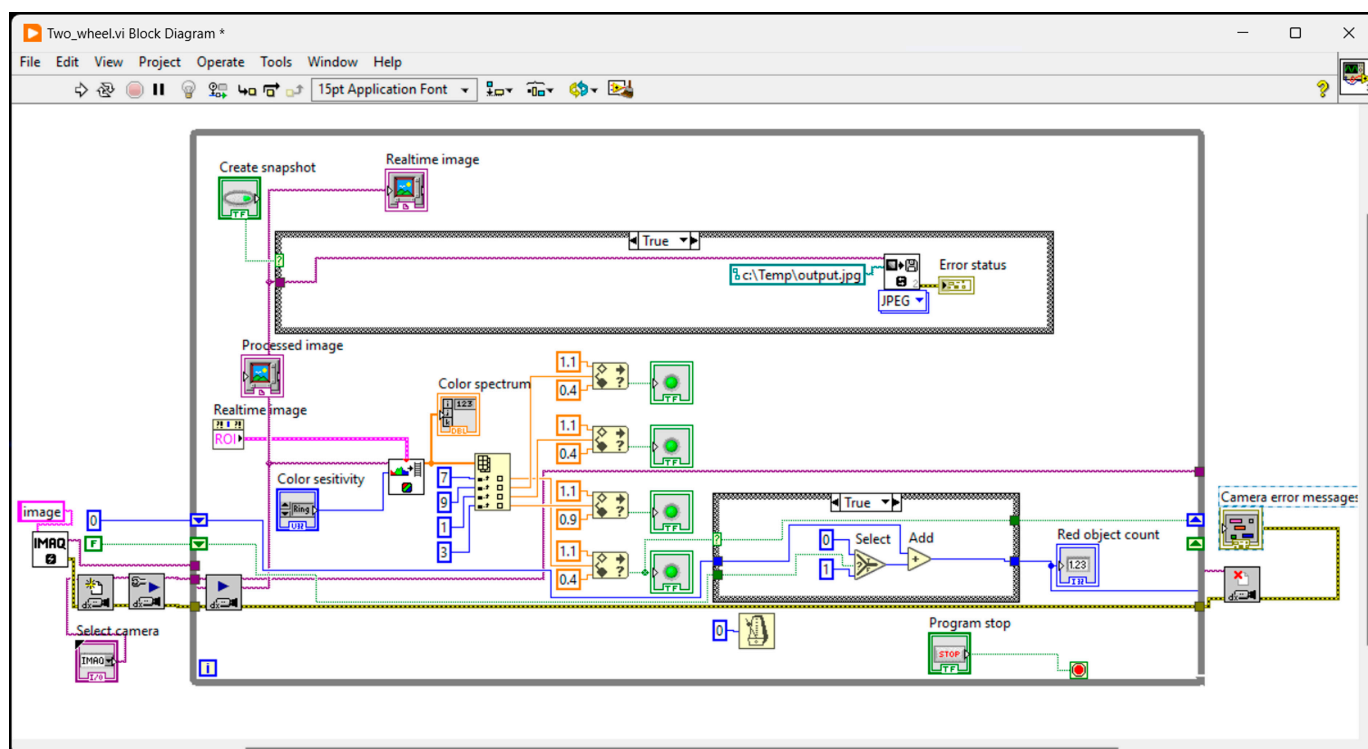


Figure 11. Two-wheel robot vision/classification code in LabVIEW.

The LabVIEW Vision toolbox is a powerful solution for applications that require real-time image processing and analysis. The LabVIEW Vision toolbox provides a wide range of tools and functions for image processing, such as filtering, edge detection, feature extraction, and object recognition [27]. To classify green and red fruits using the LabVIEW Vision toolbox, the system needs to be trained to recognize the color of the fruit. This can be done by using a set of training images of green and red fruits, and then using machine learning algorithms, such as neural networks or support vector machines, to classify the images based on their color. Once the system is trained, it can be used to classify the color of the fruits in real-time [28]. This can be done by acquiring an image of the fruit using a camera, preprocessing the image to enhance its quality, and then analyzing the image to identify the color of the fruit. The LabVIEW Vision toolbox provides a range of functions to perform these tasks, such as image acquisition, image enhancement, and color analysis. The classification results can be displayed on a monitor or sent to a control system for further processing [29]. The system can also be configured to trigger an alarm or send a notification when a certain number of fruits of a certain color are detected. In summary, the LabVIEW Vision toolbox can be used to classify green and red fruits based on their color. The system can be trained using machine learning algorithms and can perform real-time image processing and analysis. This can be a useful solution for fruit sorting and grading applications, where accurate color classification is important.

4. Discussion

This section deals with the results obtained from the simulation experiment using a sinusoidal constant disturbance and response using manual control. A resulting map of an unknown environment was generated and the vision system was tested. Figure 12 indicates the behavior of the controller, which was realized in the LabVIEW environment, to a constant disturbance in the form of a sinusoidal signal.

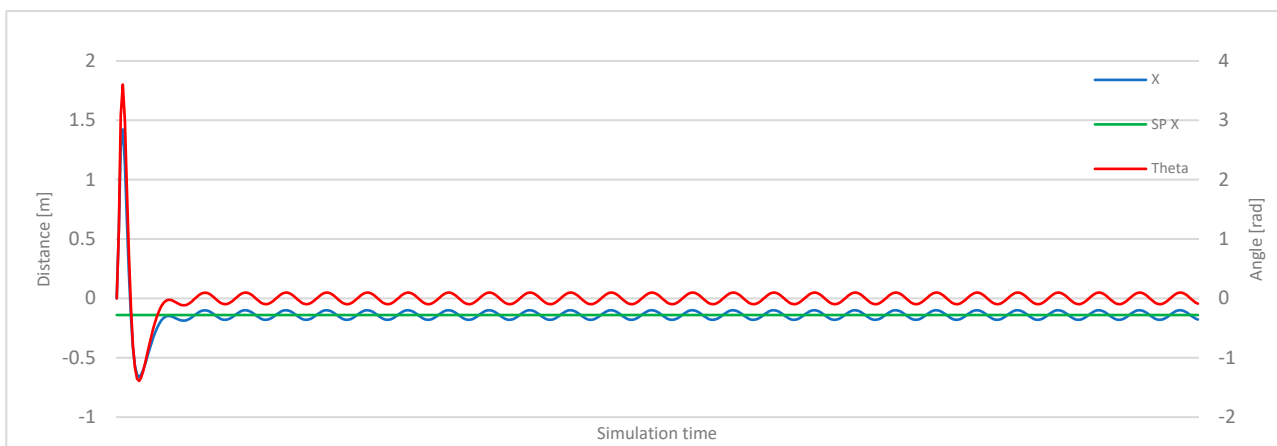


Figure 12. Controller response to sinusoidal disturbance.

Controlling a mobile robot is a problem that is solved by determining velocities and angular velocities (or forces and torques in the case of using a dynamic robot model) that will bring the robot into a given configuration, enabling it to follow a given trajectory or more generally, to perform a given task with a certain performance. The robot can be controlled by guidance (open-loop) and regulation [7]. Guidance can be used to follow a given trajectory by dividing it into segments, usually straight lines and circular segments. The guidance problem is solved by pre-calculating a smooth path from the initial to the given final configuration (see in Figure 13). This method of management has numerous disadvantages [30]. The biggest one is that it is often difficult to determine a feasible path to a given configuration, taking into account the kinematic and dynamic limitations of the robot, and the inability of the robot to adapt if some dynamic change occurs in the environment.

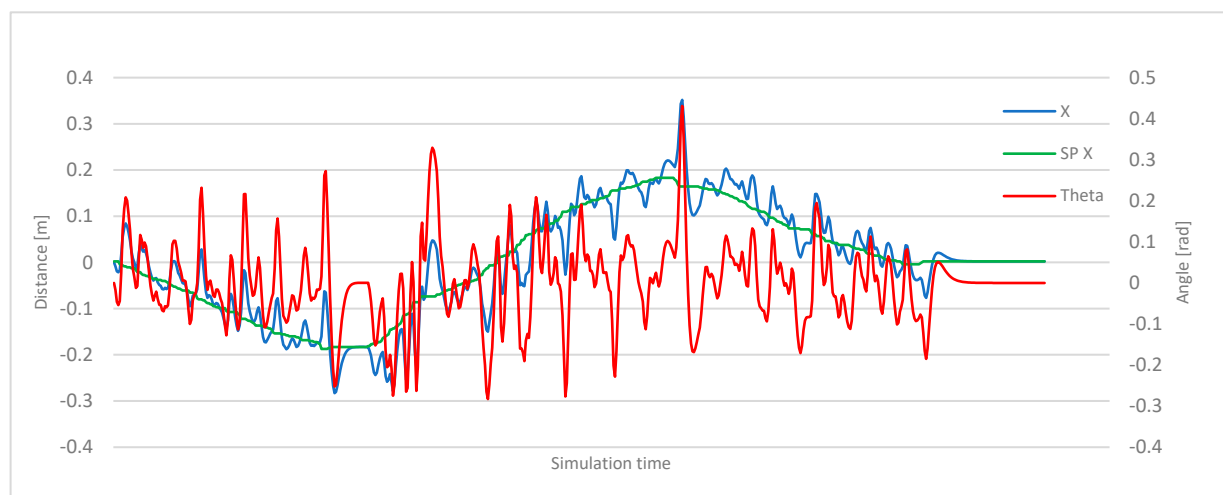


Figure 13. Controller response using manual control.

In 2015, the IEEE 1873-2015 standard was adopted for displaying two-dimensional maps for use in robotics. An XML record of local maps is defined, which can be topological, grid-based, and geometric. Although the map record in XML format takes up more memory space than other records, its main advantage is that it is “readable” and easy to debug and remove errors, if any. EKF is used as part of the methods for purposes such as SLAM, a method that simultaneously maps an unknown space and localizes a robot within that space [31,32]. The colored part of the image on Figure 14 represents the obstacle while the non-colored area is freely accessible by the two wheeled mobile robot.

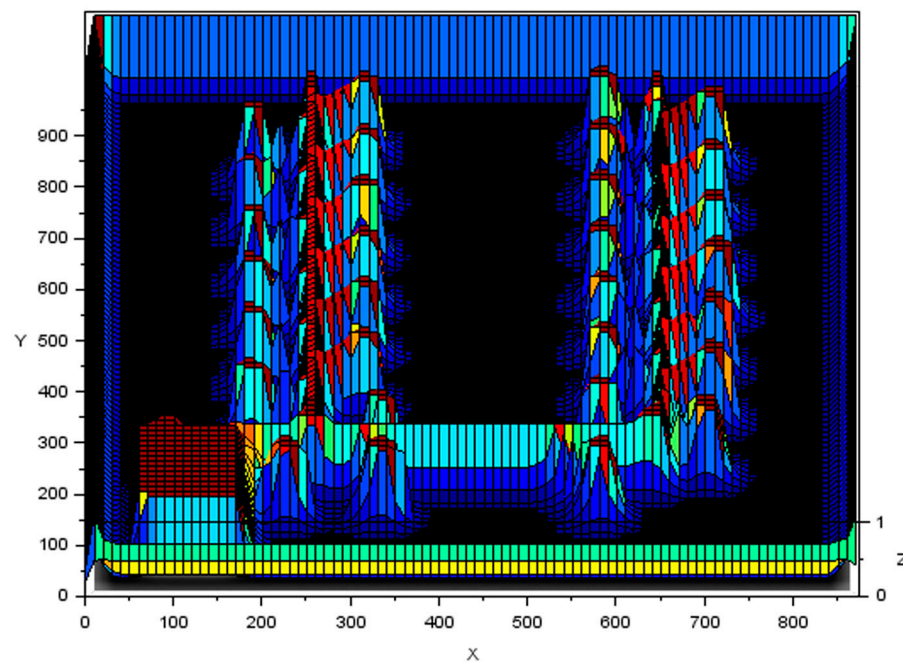


Figure 14. Generated map of the two-wheel mobile robot in a greenhouse environment.

SLAM is a process by which a robot creates a map of the environment and simultaneously uses that map to obtain its location. The trajectory of the robotic platform and the locations of objects (obstacles) in space are not known in advance. There are two formulations of the SLAM problem. The first is the online SLAM problem, where the current a posteriori probability is estimated. The second formulation is a complete (full) SLAM problem, which differs from the previous one in that it requires an a posteriori probability estimate for robot configurations during the entire trajectory.

The purpose of image segmentation is to simplify the image, dividing it into sub-regions or sets of pixels to make the content more understandable and easier for computer analysis [33]. In other words, the image of a complex scene is transformed by a segmentation process in order to extract the elements essential for a specific application and to remove irrelevant details from the image. The goal is therefore to separate objects of interest from the background by assigning a description to each pixel of the image so that pixels with the same description share the same specific visual characteristic as depicted in Figure 15.



Figure 15. Original and segmented image from the vision subsystem.

In precision agriculture, fast and robust object recognition is necessary, as well as flexible adaptation of the robot to the environment. This problem is approached in this paper by applying 2D vision systems to a two-wheel mobile robot, which is trained to check the fruit maturity in a greenhouse environment. The camera was fixed and mounted so that

it observes the entire scene. In order to recognize objects, algorithms of frame detection, edge detection, and matching with previously known shapes and colors were used.

This research is part of a larger project; the future research involves integrating these results into the larger system and implementing the proposed solutions in real-world Agriculture 4.0 applications. This may involve further testing, optimization, and validation of the proposed methods, as well as addressing any limitations or challenges that may arise during implementation. The research is focused on developing new methods and techniques; the future path may involve further refinement and development of the proposed methods, as well as evaluating their effectiveness and comparing them to other existing approaches.

5. Conclusions

In conclusion, the development of a self-balancing, two-wheel-driven, SLAM-based, unmanned system for Agriculture 4.0 applications has implemented a fuzzy-based two-wheel stability control, a SLAM-based map generation method, and a vision-based fruit classification algorithm. The use of these systems can reduce the manual labor required to monitor and control crops, thereby increasing efficiency and productivity in the agriculture industry. However, one of the major challenges faced in the development of these systems is maintaining their balance, especially when they navigate over uneven terrain. To overcome this challenge, fuzzy control was employed as a viable control technique to ensure that the system remains balanced. Fuzzy control is a type of control system that uses rules and reasoning based on fuzzy logic to make decisions. It has been used in several applications, including robotics and automation, due to its ability to handle complex and uncertain systems. The implementation of fuzzy control in the self-balancing, two-wheel-driven, SLAM-based, unmanned systems for Agriculture 4.0 applications resulted in the improved performance, stability, and accuracy of the system. The fuzzy control system uses feedback from the sensors on the system to adjust the speed and direction of the wheels, ensuring that the system maintains its balance. This has improved the accuracy and efficiency of the system's movement, thereby reducing the risk of damage to crops and soil. Moreover, the use of SLAM-based mapping has further improved the performance of these unmanned systems. SLAM allows the system to create and update a map of its environment in real-time, thereby improving the system's ability to navigate over uneven terrain. The integration of SLAM and fuzzy control further improved the accuracy and efficiency of these unmanned systems, making them an essential tool for precision agriculture. In summary, the use of self-balancing, two-wheel-driven, SLAM-based, unmanned systems for Agriculture 4.0 applications has revolutionized the way farming is performed. However, maintaining the balance of these systems has been a significant challenge. The integration of fuzzy control and SLAM-based mapping has improved the accuracy, stability, and efficiency of these systems, making them an essential tool for precision agriculture. Further research and development in this area are necessary to improve the performance and capability of these systems for future agriculture applications.

This research is based on simulations or modeling, and the validity of the results depends on the accuracy of the model and the assumptions made during the simulation. It is important to compare the results with the existing literature and validate the findings using additional experiments or simulations. The simulation results and early experimental results are promising. The realization of the experimental setup is in progress and the results will be published in the future.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declare no conflict of interest.

References

1. Yang, C.; Li, Z.; Cui, R.; Xu, B. Neural network-based motion control of an underactuated wheeled inverted pendulum model. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 2004–2016. [CrossRef] [PubMed]
2. Yue, M.; Wang, S.; Sun, J.Z. Simultaneous balancing and trajectory tracking control for two-wheeled inverted pendulum vehicles: A composite control approach. *Neurocomputing* **2016**, *191*, 44–54. [CrossRef]
3. Hwang, C.L.; Chiang, C.C.; Yeh, Y.W. Adaptive fuzzy hierarchical sliding-mode control for the trajectory tracking of uncertain underactuated nonlinear dynamic systems. *IEEE Trans. Fuzzy Syst.* **2014**, *22*, 286–299. [CrossRef]
4. Shu, Y.; Bo, C.; Shen, G.; Zhao, C.; Li, L.; Zhao, F. Magicol: Indoor Localization Using Pervasive Magnetic Field and Opportunistic WiFi Sensing. *IEEE J. Sel. Areas Commun.* **2015**, *33*, 1443–1457. [CrossRef]
5. Ouyang, G.; Abed-Meraim, K. Analysis of Magnetic Field Measurements for Indoor Positioning. *Sensors* **2022**, *22*, 4014. [CrossRef] [PubMed]
6. Csik, D.; Odry, A.; Sarcevic, P. Comparison of RSSI-Based Fingerprinting Methods for Indoor Localization. In Proceedings of the IEEE International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 15–17 September 2022. [CrossRef]
7. Sarcevic, P.; Csik, D.; Odry, A. Indoor 2D Positioning Method for Mobile Robots Based on the Fusion of RSSI and Magnetometer Fingerprints. *Sensors* **2023**, *23*, 1855. [CrossRef] [PubMed]
8. Filip, I.; Pyo, J.; Lee, M.; Joe, H. LiDAR SLAM with a Wheel Encoder in a Featureless Tunnel Environment. *Electronics* **2023**, *12*, 1002. [CrossRef]
9. Samodro, M.; Puriyanto, R.; Caesarendra, W. Artificial Potential Field Path Planning Algorithm in Differential Drive Mobile Robot Platform for Dynamic Environment. *Int. J. Robot. Control. Syst.* **2023**, *3*, 161–170. [CrossRef]
10. Lei, L.; Ning, Z.; Yi, Z.; Yangjun, S. A* guiding DQN algorithm for automated guided vehicle pathfinding problem of robotic mobile fulfillment systems. *Comput. Ind. Eng.* **2023**, *178*, 109112. [CrossRef]
11. Rubio, F.; Valero, F.; Llopis-Albert, C. A review of mobile robots: Concepts, methods, theoretical framework, and applications. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419839596. [CrossRef]
12. Thakar, S.; Srinivasan, S.; Al-Hussaini, S.; Bhatt, P.M.; Rajendran, P.; Jung Yoon, Y.; Dhanaraj, N.; Malhan, R.K.; Schmid, M.; Krovi, V.N.; et al. A Survey of Wheeled Mobile Manipulation: A Decision-Making Perspective. *ASME J. Mech. Robot.* **2023**, *15*, 020801. [CrossRef]
13. Sabzalian, M.H.; Alattas, K.A.; Aredes, M.; Alanazi, A.K.; Abo-Dief, H.M.; Mohammadzadeh, A.; Fekih, A. A New Immersion and Invariance Control and Stable Deep Learning Fuzzy Approach for Power/Voltage Control Problem. *IEEE Access* **2021**, *10*, 68–81. [CrossRef]
14. Sabzalian, M.H.; Mohammadzadeh, A.; Rathinasamy, S.; Zhang, W. A developed observer-based type-2 fuzzy control for chaotic systems. *Int. J. Syst. Sci.* **2021**. [CrossRef]
15. Rastgar, H.; Naeimi, H.; Agheli, M. Characterization, validation, and stability analysis of maximized reachable workspace of radially symmetric hexapod machines. *Mech. Mach. Theory* **2019**, *137*, 315–335. [CrossRef]
16. Yang, T.; Cabani, A.; Chafouk, H. A Survey of Recent Indoor Localization Scenarios and Methodologies. *Sensors* **2021**, *21*, 8086. [CrossRef] [PubMed]
17. Filipenko, M.; Afanasyev, I. Comparison of Various SLAM Systems for Mobile Robot in an Indoor Environment. In Proceedings of the 9th IEEE International Conference on Intelligent Systems, Madeira, Portugal, 25–27 September 2018; pp. 400–407. [CrossRef]
18. Available online: <https://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum§ion=ControlStateSpace> (accessed on 26 March 2023).
19. Liu, J.; Wang, X. *Advanced Sliding Mode Control for Mechanical Systems: Design, Analysis and Matlab Simulation*; Springer: Berlin/Heidelberg, Germany, 2011; ISBN 978-3-642-20906-2. [CrossRef]
20. Wang, G.; Ding, L.; Gao, H.; Deng, Z.; Liu, Z.; Yu, H. Minimizing the Energy Consumption for a Hexapod Robot Based on Optimal Force Distribution. *IEEE Access* **2019**, *8*, 5393–5406. [CrossRef]
21. Campos, C.; Elvira, R.; Gomez, J.J.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [CrossRef]
22. Júnior, G.P.C.; Rezende, A.M.; Miranda, V.R.; Fernandes, R.; Azpúrua, H.; Neto, A.A.; Pessin, G.; Freitas, G.M. EKF-LOAM: An Adaptive Fusion of LiDAR SLAM With Wheel Odometry and Inertial Data for Confined Spaces With Few Geometric Features. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 1458–1471. [CrossRef]
23. Jang, H.; Kim, T.Y.; Lee, Y.C.; Song, Y.H.; Choi, H.R. Autonomous Navigation of In-Pipe Inspection Robot Using Contact Sensor Modules. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 4665–4674. [CrossRef]
24. Piao, J.C.; Kim, S.D. Real-Time Visual-Inertial SLAM Based on Adaptive Keyframe Selection for Mobile AR Applications. *IEEE Trans. Multimed.* **2019**, *21*, 2827–2836. [CrossRef]
25. Ni, X.; Li, C.; Jiang, H.; Takeda, F. Three-dimensional photogrammetry with deep learning instance segmentation to extract berry fruit harvestability traits. *ISPRS J. Photogramm. Remote Sens.* **2021**, *171*, 297–309. [CrossRef]
26. Mukhiddinov, M.; Muminov, A.; Cho, J. Improved Classification Approach for Fruits and Vegetables Freshness Based on Deep Learning. *Sensors* **2022**, *22*, 8192. [CrossRef] [PubMed]
27. Rizzo, M.; Marcuzzo, M.; Zangari, A.; Gasparetto, A.; Albarelli, A. Fruit ripeness classification: A survey. *Artif. Intell. Agric.* **2023**, *7*, 44–57. [CrossRef]

28. Castro, W.; Oblitas, J.; De-La-Torre, M.; Cotrina, C.; Bazán, K.; Avila-George, H. Classification of cape gooseberry fruit according to its level of ripeness using machine learning techniques and different color spaces. *IEEE Access* **2019**, *7*, 27389–27400. [[CrossRef](#)]
29. Zhu, L.; Spachos, P.; Pensini, E.; Plataniotis, K.N. Deep learning and machine vision for food processing: A survey. *Curr. Res. Food Sci.* **2021**, *4*, 233–249. [[CrossRef](#)] [[PubMed](#)]
30. Yuesheng, F.; Jian, S.; Fuxiang, X.; Yang, B.; Xiang, Z.; Peng, G.; Zhengtao, W.; Shengqiao, X. Circular Fruit and Vegetable Classification Based on Optimized GoogLeNet. *IEEE Access* **2021**, *9*, 113599–113611. [[CrossRef](#)]
31. Chiu, Y.C.; Chen, S.; Lin, J.F. Study of an autonomous fruit picking robot system in greenhouses. *Eng. Agric. Environ. Food* **2013**, *6*, 92–98. [[CrossRef](#)]
32. Hu, C.; Liu, X.; Pan, Z.; Li, P. Automatic detection of single ripe tomato on plant combining faster R-CNN and intuitionistic fuzzy set. *IEEE Access* **2019**, *7*, 154683–154696. [[CrossRef](#)]
33. Yue, X.Q.; Shang, Z.Y.; Yang, J.Y.; Huang, L.; Wang, Y.Q. A smart data-driven rapid method to recognize the strawberry maturity. *Inf. Process. Agric.* **2020**, *7*, 575–584. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.