sensors

MDPI

Article

Autonomous Navigation of a Center-Articulated and Hydrostatic Transmission Rover Using a Modified Pure Pursuit Algorithm in a Cotton Field

Kadeghe Fue [1,2,*], Wesley Porter [3], Edward Barnes [4], Changying Li [1] and Glen Rains [2,*]

[1] College of Engineering, University of Georgia, Athens, GA 30602, USA; cyli@uga.edu
[2] Department of Entomology, University of Georgia, Tifton, GA 31793, USA
[3] Department of Crop and Soil Sciences, University of Georgia, Tifton, GA 31793, USA; wporter@uga.edu
[4] Cotton Incorporated, Cary, NC 27513, USA; ebarnes@cottoninc.com
[*] Correspondence: kadefue@uga.edu (K.F.); grains@uga.edu (G.R.); Tel.: +1-229-386-3520 (G.R.)

Received: 1 July 2020; Accepted: 4 August 2020; Published: 7 August 2020

check for updates

Abstract: This study proposes an algorithm that controls an autonomous, multi-purpose, center-articulated hydrostatic transmission rover to navigate along crop rows. This multi-purpose rover (MPR) is being developed to harvest undefoliated cotton to expand the harvest window to up to 50 days. The rover would harvest cotton in teams by performing several passes as the bolls become ready to harvest. We propose that a small robot could make cotton production more profitable for farmers and more accessible to owners of smaller plots of land who cannot afford large tractors and harvesting equipment. The rover was localized with a low-cost Real-Time Kinematic Global Navigation Satellite System (RTK-GNSS), encoders, and Inertial Measurement Unit (IMU)s for heading. Robot Operating System (ROS)-based software was developed to harness the sensor information, localize the rover, and execute path following controls. To test the localization and modified pure-pursuit path-following controls, first, GNSS waypoints were obtained by manually steering the rover over the rows followed by the rover autonomously driving over the rows. The results showed that the robot achieved a mean absolute error (MAE) of 0.04 m, 0.06 m, and 0.09 m for the first, second and third passes of the experiment, respectively. The robot achieved an MAE of 0.06 m. When turning at the end of the row, the MAE from the RTK-GNSS-generated path was 0.24 m. The turning errors were acceptable for the open field at the end of the row. Errors while driving down the row did damage the plants by moving close to the plants' stems, and these errors likely would not impede operations designed for the MPR. Therefore, the designed rover and control algorithms are good and can be used for cotton harvesting operations.

Keywords: robotics; GNSS; cotton robotics; autonomous navigation; RTK; ROS

1. Introduction

Mechanical harvesting has helped improve crop production significantly since the mid-1900s. Before these machines were developed, crops such as cotton were primarily hand-harvested. The development of the cotton combine helped to reduce labor costs and increase production efficiency but comes with its downsides. Harvesting is accomplished using expensive machines that are massive in size and weight, which can lead to soil compaction, and are also costly and time-consuming to repair. Breakdowns during the season may expose cotton to hostile environmental conditions that can diminish the quality of the harvest. Additionally, cotton harvesting takes place only after cotton fields have been defoliated with chemical defoliants. These chemicals can degrade the land and are an added expense to production costs. The defoliation is performed approximately 50 days from the opening of the first bolls. A consequence of waiting many weeks to harvest all the cotton simultaneously, lint quality is affected profoundly by external weather and other environmental elements from the time they open to the time they are picked [1,2]. As a consequence of the current cotton management system, small acreage farmers can not afford to buy these machines or maintain them [3,4]. Furthermore, the fast aging farming community will experience a labor shortage, since many of their children are moving to urban areas [5].

Any solution that could increase the participation of small and family farmers would be well received. One alternative to current large-scale farm and machinery systems is to introduce small, multi-purpose, robotic rovers that can navigate and perform agricultural operations autonomously without the need for human–machine control. Fortunately, there is a booming industry in robotics and machine learning technologies, and robotics have been developed to solve many pertinent issues in agriculture [1,2,5–7].

However, for mobile robotic systems to be efficient, they need very effective methods to navigate fields autonomously. The autonomous navigation of robotic systems depends upon four modules: sensors, vehicle mobility, perception, and control algorithms (Figure 1). Sensors, such as RTK-GNSS, Red-Green-Blue (RGB) cameras, Stereo camera, Light Detection and Ranging (LiDAR), Sound Navigation and Ranging (SONAR), Ultrasonic, Radio-frequency identification (RFID), Inertial Measurement Unit (IMU), Laser scanner, RAdio Detection And Ranging (RADAR), encoders, thermal imaging, hyperspectral, and infrared, have been used extensively to detect fruits and plants in agricultural fields (Figure 1) [5,7–25]. Additionally, many vehicle mobility systems have been developed primarily for agricultural use, such as continuous tracks, the Ackermann four-wheel drive, center-articulated drives, legged-robots, swinging robots, omnidirectional drives, and sliding-on-the-rail robots (Figure 1). The mobility is designed to accommodate different soil and topographic conditions, open or greenhouse farming, and maneuverability requirements. Perception is created using sensor output and machine vision algorithms, such as image segmentation, hough transformation, sensor fusion, or machine learning, to obtain environmental features (Figure 1). Models that can do perception with the lower cost of computation are essential to be used in small mobile robots [26,27]. After perceiving the environment, the mobile robot performs a movement action to the next location. There are multiple control system methods used: fuzzy logic, nonlinear, proportional-integral-derivative (PID), adaptive, model-based, rear-wheel feedback, linear-quadratic regulator, model predictive control (MPC), and machine learning, such as neural networks and reinforcement learning. After assimilating a control system, several techniques must be incorporated to develop autonomous navigation in an unstructured environment, such as localization, mapping, obstacle avoidance, simultaneous localization, and Mapping (SLAM), row-following in row crops and path planning to perform complex farm operation movements [5,7–25]. Furthermore, algorithms that guide the robots to avoid obstacles should be incorporated so that the robot can work smoothly in the agricultural field while interacting with humans and other robots [28–30]. Additionally, there are proposals to use the Internet of Things and sensors to monitor and detect the operating conditions of vehicles, such as fluid leaking in a hydrostatic tractor, which can be incorporated in robotic systems too [31,32]. Additionally, all the computations to guide the robot can be done onboard to reduce network traffic and develop green communications [33]. Robot navigation in field crops has been studied well in electric vehicles with Ackerman and omnidirectional steering [34,35]. Limited work has been put forward for the center-articulated vehicle, which is a special kind of off-road vehicle that has very high traction and maneuverability and, hence, is common in agriculture, forestry, and construction industry [36]. Center-articulated vehicles can easily turn corners in a constrained environment by using two hydraulic cylinders that can change the yaw angle of the two parts of the vehicle [36]. Due to this configuration, control of the center-articulated vehicle becomes a challenge because a steering angle has a direct influence on the heading error for both parts of the vehicle and, hence, small mistakes in turning can cause large path errors [36]. Since it is working off-road, it becomes more difficult due to the rough terrain of the land. There are limited control methods for center-articulated vehicles compared to traditional vehicles [35,36]. Hence, more work should be done to improve path following of the center-articulated vehicle in real field conditions [36]. Additionally, center-articulated vehicles have the potential to be used in cotton fields because they have high traction and can be small but able to pass over high-density cotton fields easily, compared to other similar sized vehicles. In this paper, a control method to improve the path following of the center-articulated vehicles is proposed, developed, and evaluated in a real field.

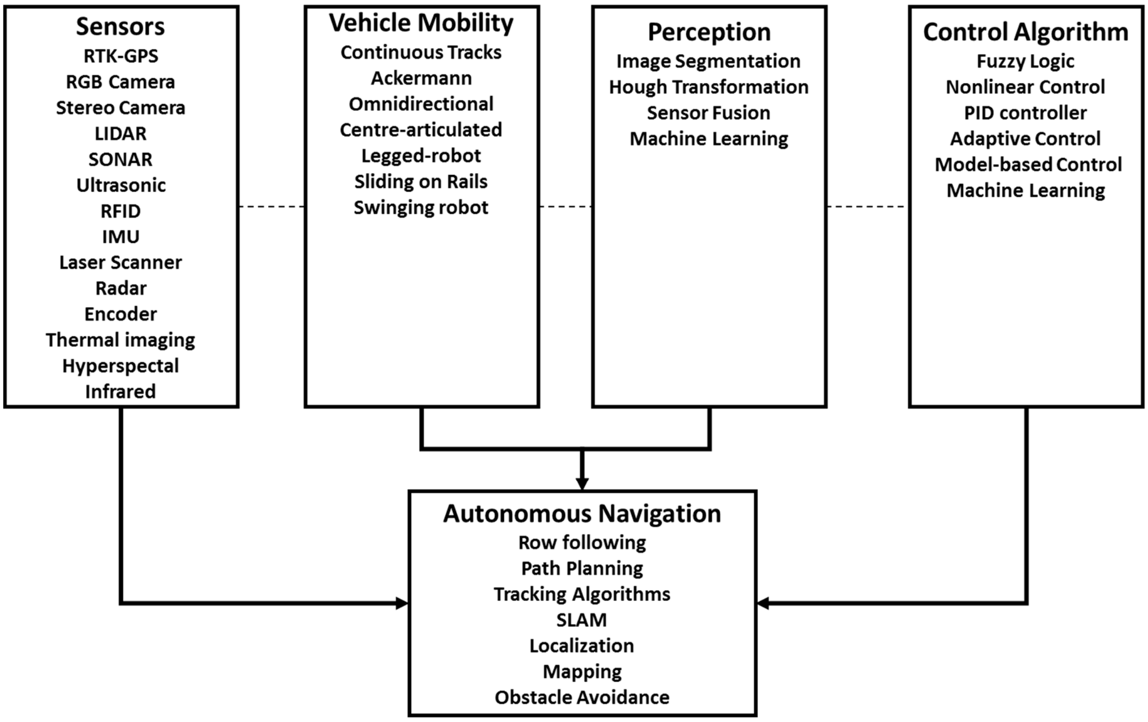

Figure 1. Autonomous navigation modules.

In this study, two IMUs, a high precision potentiometer, two encoders, a low-cost single-frequency RTK-GNSS, and the sensor fusion algorithm, Extended Kalman Filter (EKF), were utilized to perform the autonomous localization and navigation of the robot. Proportional control and a modified pure pursuit algorithm were implemented to perform autonomous cotton row following for a MPR

The development and performance of the autonomous navigation of an MPR are presented in this paper. Autonomous means the rover can navigate itself along cotton rows without the intervention of human subjects and without destroying plants or cotton bolls. High precision is required to achieve acceptable navigation without causing an economic reduction in yield. Therefore, in this study, we present two objectives:

1. Development of the navigation system of the autonomous center-articulated hydrostatic transmission MPR.
2. Evaluation of the navigation of the autonomous center-articulated hydrostatic drive MPR in a cotton field.

2. Materials and Methods

2.1. Robot Components and System Setup

The rover (Figure 2) was a custom-built four-wheel center-articulated robot (West Texas Lee Corp., Lubbock, Texas). The rover was 340 cm long with front and back parts (divided by the center of articulation) being 145 cm and 195 cm long, respectively. The rover's height and width could be adjusted to a maximum of 122 cm and 234 cm, respectively. The rover tires were 91 cm from the center of the vehicle. The rover was 212 cm wide, with a tire width of 30 cm. The four tires had a radius of 30.48 cm and a circumference of 191.51 cm. The rover had a ground clearance of 91 cm. The rover used seven sensors; two IMUs, a high precision potentiometer, two rotary encoders, and RTK-GNSS. Each front tire was connected to a rotary encoder (Koyo incremental (quadrature) TRDA-20R1N1024VD, Automationdirect.com, Atlanta, GA, USA). The two IMUs (Phidget Spatial Precision 3/3/3 High-Resolution model 1044_1B, Calgary, AB, Canada) were placed in front of the rover. The first IMU was placed 95 cm above the ground and 31 cm from the front of the vehicle. The second IMU was 132 cm above the ground and 46 cm from the front of the vehicle. The low-cost RTK-GNSS (USD 800) single-frequency receiver (EMLID Reach RS, Hong Kong, China) was placed 246 cm above the ground and 30 cm from the front of the vehicle. An embedded system (NVIDIA Jetson AGX Xavier development kit, Nvidia Corp., Santa Clara, CA, USA) was installed and used to control the rover navigation and read sensor data.

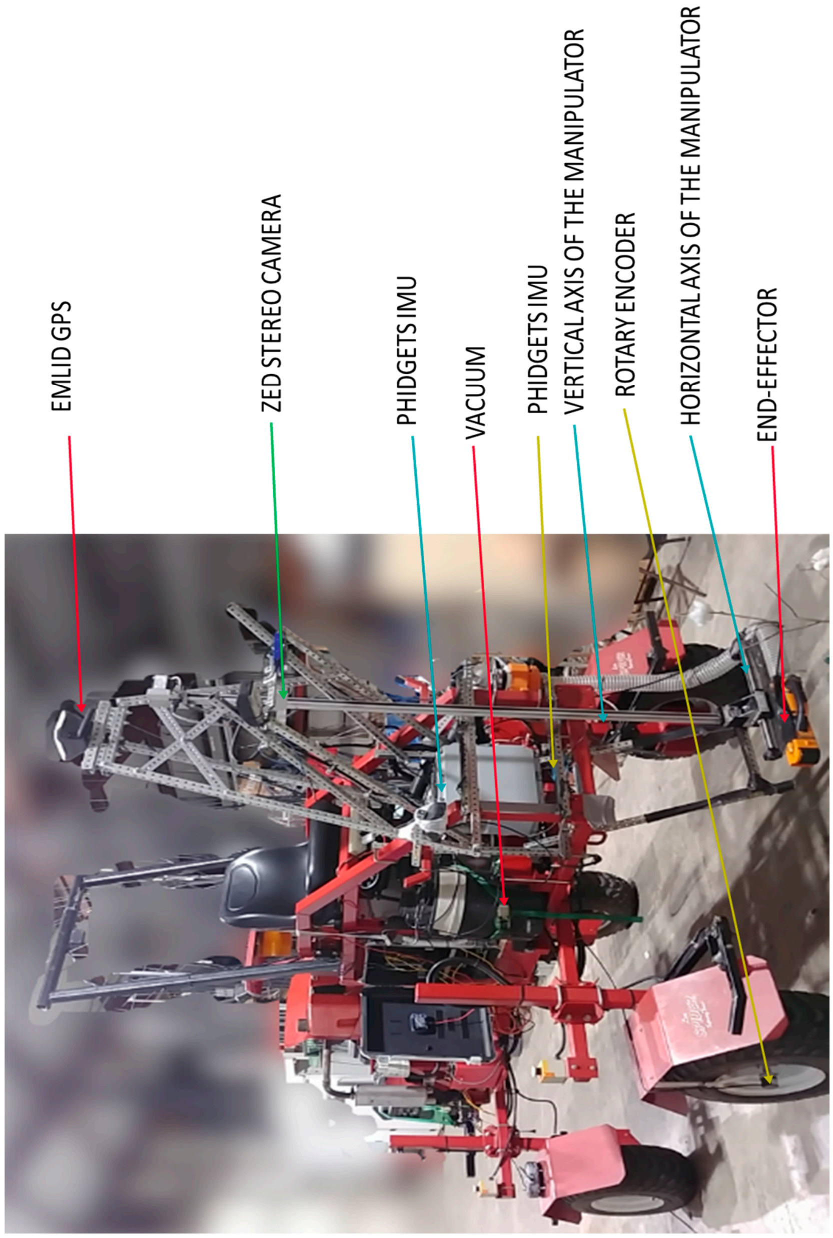

Figure 2. The red research rover with manipulator and sensors attached in front of the rover implemented for this study.

All sensors except the rotary encoders and potentiometer were connected to the embedded system via a universal serial bus (USB). The encoders were connected to the rover navigation controller (Arduino Mega 2560, Arduino LLC) using four wires: signals A and B, power, and ground, so they can register the rotation of the tires by detecting the leading edge of rising square waves. A high precision potentiometer (Vishay Spectral Single Turn, Malvern, PA) was used to report the articulation angle of the vehicle by measuring the electric potential caused by the turn of the vehicle, which was correlated to the angle. The potentiometer was connected to the rover navigation controller.

ROS (Robot Operating System), which is robotics middleware used for robot software development, was implemented to connect the embedded system (Jetson Xavier) with the rover navigation controller. The robot software was developed to communicate using ROS topics. ROS topics were named buses that the nodes (embedded system and navigation controller) used to exchange messages [37]. The sensors connected to the embedded system published the updates that were utilized by both the embedded system and navigation controller. The topics were set to communicate so that they did not know the other nodes they were communicating with [37,38]. The rover navigation controller received a signal from the embedded computer to control the rover movement, articulation, and engine throttling. All four wheels of the rover were mounted to hydraulic motors (Parker 2090B 238 cc/rev, Parker Hannifin Corp., Mayfield Heights, OH, USA) that had their rotation controlled using a linear actuator to the swashplate lever of a 14.1 cc/rev axial-piston variable rate pump (OilGear, Milwaukee, WI, USA). The swashplate angle was controlled by the rover controller that determined the placement of the linear electric servo (Robotzone HDA4, Servocity, Winfield, KS) that had a maximum movement of approximately 10.16 cm. The left/right articulation was controlled through a 4-port 3-way open-center directional control valve (DCV) connected to two 1.75 in hydraulic cylinders (LD15012014H-A, West Texas Lee Corp., Lubbock, TX, USA) and powered by a 0.45 cc/rev fixed displacement pump (Bucher Hydraulics, Klettgau, Germany) in tandem with the variable rate pump. The DCV provided hydraulic fluid to hydraulic cylinders that controlled the rover′s articulation. The rover could turn a maximum of 45 degrees with a wheelbase of 190 cm. The engine throttle was connected to an onboard Kohler Command 20HP engine (CH20S, Kohler Co., Kohler, WI, USA) with a maximum of 2500 RPM and powered the tandem variable- and fixed-rate pumps. The speed of the engine can be adjusted to higher or lower RPMs using the servo connected to the carburetor linkage that controls the amount of air into the engine cylinders. The throttle was set at approximately 2200 RPM. The front tires were connected to a rotary encoder to provide feedback on the movement of the rover along the crop rows. Left/right articulation was controlled by using relays connected to the DCV.

2.2. Real-Time Kinematic GNSS and Network Transport of Radio Technical Commission for Maritime Services (RTCM) via Internet Protocol (NTRIP)

The RTK-GNSS receiver used to acquire the global position of the rover used an NTRIP provider (eGPS Solutions, Norcross, GA, USA) to obtain differential correction through the internet using a Verizon modem (Inseego Jetpack MiFi 8800L, Verizon Wireless, New York, NY, USA) (Figure 3). The GNSS correction signal was obtained using the NTRIP signal with a mounting point within 3 kilometers of the test plot and downloaded to the RTK-GNSS through a Verizon Hotspot and wireless signal. NTRIP servers received the message from the base RTK-GNSS receivers connected to it. A data plan subscription was required to use the modem to acquire data through the internet from the eGPS base station network instead of using a local base station. The service to our NTRIP provider was registered, and a username, password, and I.P. address (mount point) to connect to the NTRIP provider via Internet Protocol were provided. Using NTRIP was advantageous because GNSS corrections were acquired without the need to set up a base station.

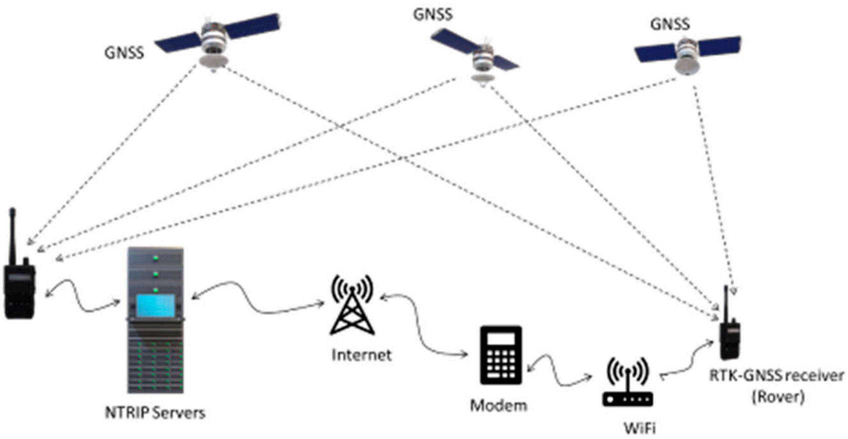

Figure 3. Context diagram of the Network Transport of RTCM via Internet Protocol (NTRIP).

2.3. Calibration of the Potentiometer, IMUs, and Encoders

The potentiometer (at point H in Figure 4) measured the articulation angle. Figure 4 shows the aerial view of the right turning center articulated rover. When the rover was straight, P_1 was parallel to P_2 and the potentiometer digital signal read 493. P_1 is the center point between front tires, while P_2 is the cnter point between the rear tires. The distance between P_1 and P_2 is Z. The length of l_1 and l_2 was 0.91 m each. To obtain the articulation angle γ, manual measurements were made and applied using the Cosine rule progression below (Equation (1));

$$Z^2 = l_1{}^2 + l_2{}^2 + 2 \times l_1 \times l_2 \times \cos\,\theta \tag{1}$$

$$\theta = \cos^{-1}\left(\frac{l_1{}^2 + l_2{}^2 - Z^2}{2 \times l_1{}^2 \times l_2{}^2}\right)$$

$$\gamma = \pi - \cos^{-1}\left(\frac{l_1{}^2 + l_2{}^2 - Z^2}{2 \times l_1{}^2 \times l_2{}^2}\right)$$

$$\gamma = \pi - \cos^{-1}\left(\frac{0.91^2 + 0.91^2 - Z^2}{2 \times 0.91^2 \times 0.91^2}\right)$$

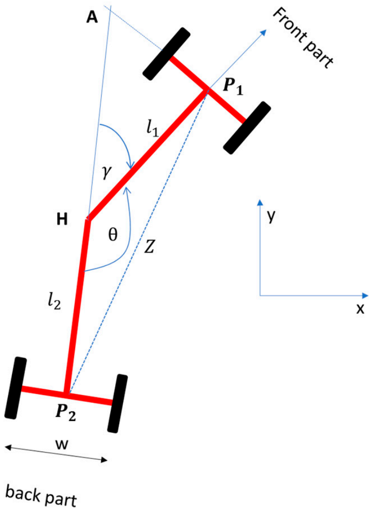

Figure 4. Potentiometer Calibration. Potentiometer calibration involves measurements of the voltage reported by the potentiometer in relation to the changing angle θ of the rover when turning left or right. Assume all the points are in a cartesian coordinate system.

To calibrate the angle, the rover was turned left or right. The angle was measured by the potentiometer, which was digitized with a 10-bit ADC, and the signal values ranged from 0 to 1023. The angle was recorded when turning left and right in 20 digital signal intervals from 493 (Figure 5a,b). Assume 493 as the center position and going left is negative while going right is positive. The angle γ was plotted together with the potentiometer signal (Figure 5). The potentiometer signal decreased when turning left and increased when turning right. The Equation obtained from the plots for the left was y = 0.190225x, and for the right was y = 0.1932075x (Figure 5). The Equations were implemented in Algorithm 1 at lines 15 to 19. The left/right Equations were slightly different due to potentiometer errors, human errors, and the slight misalignment of the vehicle.

Algorithm 1: Proportional control of the articulation angle.

Input: Angle reported by the high precision potentiometer γ_k , target angle γ_{k+1} and threshold E_t
Output: p which is equal to Kp* (γ_{k+1} - γ_k)

```
1:   Gain Kp is equal to 1
2:   p <– Kp * ($\gamma$k+1 – $\gamma$k)
3:   WHILE p > Et
4:     Declare and assign 0 to increment i
5:     Declare and assign 0 to temp
6:     WHILE i < 20
7:         Delay for one microsecond
8:         Read the analog signal from the high precision potentiometer $\gamma$k
9:         temp add the $\gamma$k to temp
10:        Increment i
11:    }
12:    Get the average temp
13:    Assign temp to $\gamma$k+1
14:    IF temp > analog signal 493
15:        $\gamma$k+1 = (temp-493) * 0.1932075;
16:    ELSE
17:        $\gamma$k+1 = (temp-493)*0.190225;
18:    END IF
19:    p <- Kp * ($\gamma$k+1 - $\gamma$k)
20:    IF p > -Et
21:        Set the left relay HIGH
22:        Set the right relay LOW
23:    ELSE IF –p < Et
24:        Set the left relay LOW
25:        Set the right relay HIGH
26:    ELSE
27:        Set the left relay LOW
28:        Set the right relay LOW
29:    END IF
30:  END WHILE
31:  Return all the error
```

Both IMUs were calibrated as advised by the manufacturer's users guide (Phidgets Inc., Calgary, CA, USA). The magnetic error correction was done by the compass calibrator software downloaded from the Phidgets website. The two IMUs were placed at two different locations on the vehicle, as described in the section "Robot components and System Setup." The IMU was calibrated by connecting the IMU to the embedded computer, which had the Phidget compass calibration program installed. The magnetic field estimated value for Tifton, Georgia, was 0.47459 T obtained online from the NOAA website (http://www.ngdc.noaa.gov/geomag-web/#igrfwmm). After entering the magnetic field value, the program was started, and the rover was driven in a circle behind the Engineering Annex fields (31.475340N, 83.528968W) in Tifton, Georgia, to generate the calibrated compass parameters. After the calibration, the IMUs were then used for localization and navigation experiments.

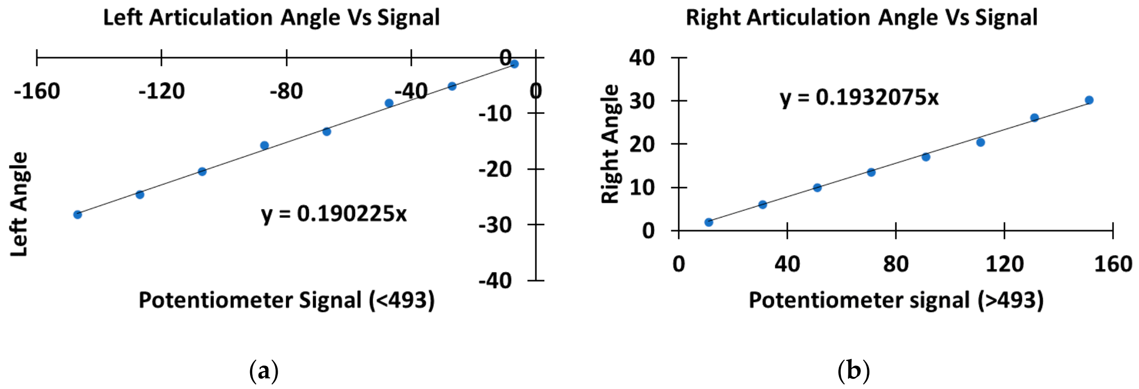

Figure 5. The calibration of the potentiometer and articulation angle. The left image (**a**) presents the relationship of the left articulation angle versus the potentiometer signal. The left image (**b**) shows the relationship of the right articulation angle versus the potentiometer signal.

Encoder calibration was conducted by finding the circumference of the rover wheels and then converting the signal of the encoder to distance for each encoder count. The rotary encoders used a 10-bit Analog-to-digital converter. To make sure that the encoders were accurately calibrated, the tires of the rover were rotated 360°, and the count of the encoders increased from 0 to 1023. Since the circumference of the tire was 1915.1 mm, the distance per count (resolution) was 1915.1/1024 = 1.87 mm.

2.4. Robot Navigation Systems

The navigation system consisted of the embedded development kit and the rover navigation controller. The rover used two algorithms to control navigation: modified pure pursuit and Proportional control [6,24,39–41]. The system used a predefined path of the GNSS signal to pass over the rows. The path was obtained by recording the rover path as it was manually driven down cotton rows (Figure 6). The predefined path was then used by the rover to navigate autonomously.

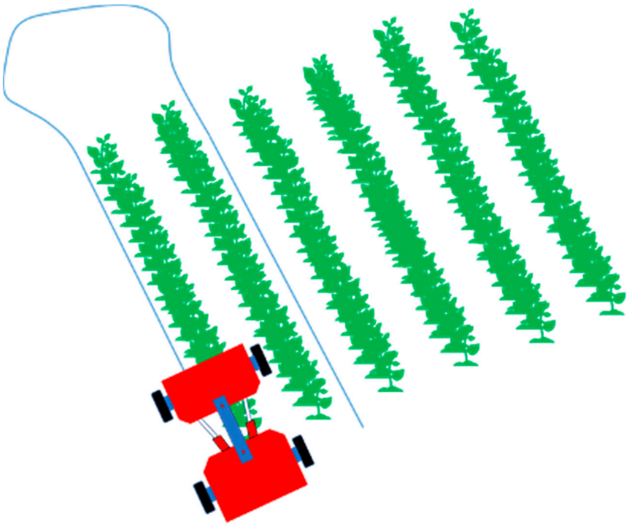

Figure 6. The rover driving along the cotton rows. Blue line is the path recorded by the rover after finishing one lap.

Since the rover used six sensors (two IMUs, potentiometer, two encoders, and RTK-GNSS) to navigate (Figure 7), the Extended Kalman Filter was implemented for simultaneous localization and navigation [25,42–44]. Sensor fusion was achieved by using the open-source ROS library "Robot localization," which provided sensor fusion and nonlinear state estimation for IMUs, encoders, and GNSS. The IMUs published two ROS topics (imu_link1/data and imu_link2/data), encoders published wheel odometry (/wheel_odom), and RTK-GNSS published /gps/fix signal [43]. The EKF localization (Figure 7) used the nav_sat_transform library to integrate fixed data from the RTK-GNSS [44]. Basically, "navsat_transform_node" required three sources of information: The robot's current pose estimate in its world frame, an earth-referenced heading, and a geographic coordinate expressed as a latitude/longitude pair (with optional altitude) (http://docs.ros.org/).

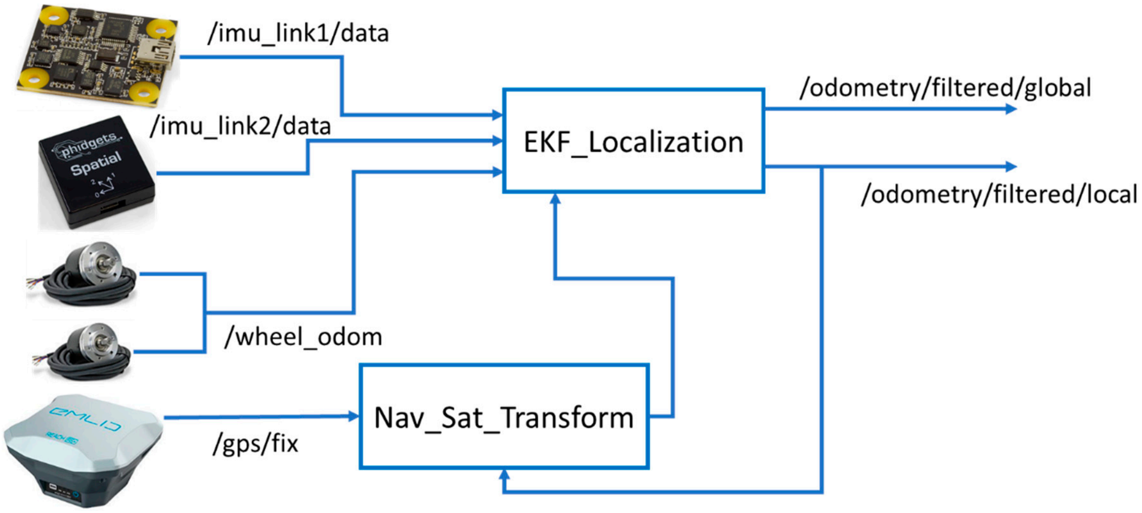

Figure 7. Simultaneous localization and navigation of the rover using dual Extended Kalman Filter (dual EKF).

EKF localization was implemented (Figure 8) as a dual EKF method that involved running two EKF's concurrently. The state and model could be estimated from the noisy observations of the IMU, wheel odometry, and GPS fixed signal.

$$x_{k+1} = F(x_k, u_k, w) + v_k \tag{2}$$

$$y_k = H(x_k, w) + n_k$$

We considered the nonlinear problem in which system states (x_k) and model parameters (w) were simultaneously estimated from the observed noisy signals (y_k). The process noise (v_k) drives the dynamical system, while the observed exogenous input noises (v_k) and observation noise (n_k) were obtained (Equation (2)). In Figure 8, two EKFs were set to run concurrently using current model estimates w_k; an EKF state filter calculated the new state in every step [42]. The lower EKF, which takes the previous state estimate as the input and output next state, estimates the next state using the current model estimate $\hat{w}_{k-1}$, while the upper EKF estimates the weight using the current state estimate $\hat{x}_{k-1}$ Then, it calculated the fresh weights of the current state estimate x_k. The model structures **F** and **H** are multilayer neural networks, while **w** are the weights [42]. In this case, the model used IMUs and encoders to generate localization estimates and then added RTK-GNSS to make global localization estimates of the rover position.

After getting the state estimates of the rover, a modified pure pursuit and PID were used to control the rover's navigation.

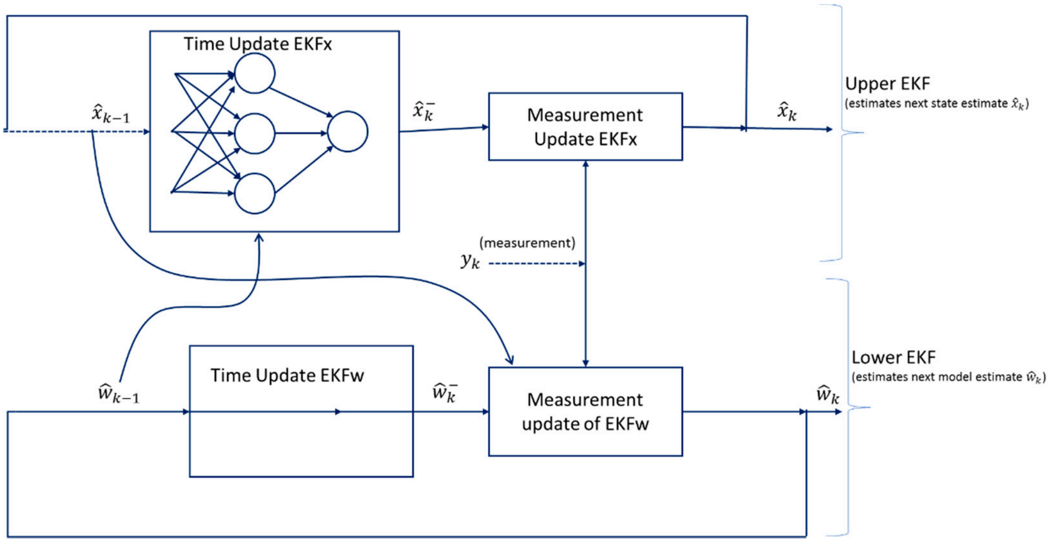

Figure 8. The dual extended Kalman filter. The method utilizes two concurrently running EKFs. State estimates are done by the top EKF using $\hat{w}_{k-1}$ for the time update, while weight estimates are generated using the bottom EKF that does measurement updates using $\hat{x}_{k-1}$.

2.5. Modified Pure Pursuit

Pure pursuit (P.P.) is a technique that computes the current vehicle position relative to a goal and then determines the curvature that would bring the vehicle back to the predefined or designated path. P.P. chooses the goal that is some distance in front of the rover. It looks ahead and determines the articulation of the tires to get into the path. The look-ahead distance changes depending on the curvature of the path and speed of the vehicle.

The rover achieved pure pursuit tracking by following six steps; determine the current location of the rover, find the path point closest to the rover, find the goal location, transform the goal location to rover coordinates, calculate the curvature, and request the rover to set the articulation to that curvature and then update the vehicle's position [24,39].

Figure 9 illustrates the center-articulated rover with all the sensors at the front, including GNSS steering to pursue the point (x,y) (goal point) at a distance L. L is also known as look-ahead distance. The radius of turning is r, while d is the horizontal distance of the goal from the center of the turning circle, and s is the horizontal distance of the rover to the goal. The articulation angle is γ, while Pe is the Path error. At the same time, s, which is equal to x, is the relative distance of the rover to the goal point.

$$\begin{gathered} \mathrm{x} + \mathrm{d} = \mathrm{r} \\ \mathrm{x} = \mathrm{s} \\ \mathrm{x}^2 + \mathrm{y}^2 = \mathrm{L}^2 \end{gathered} \tag{3}$$

Using the relationship of x, y, r, and L in Figure 9, the curvature C can be derived. If x+d = r, and $\mathrm{d}^2 + \mathrm{y}^2 = \mathrm{r}^2$, r can be found by computing its relationship with x, y coordinates, $\mathrm{r} = (\mathrm{x}^2 + \mathrm{y}^2)/(2\mathrm{x})$ [6]. Then, the turning radius r becomes:

$$\mathrm{r} = \frac{\mathrm{L}^2}{2\mathrm{x}} \tag{4}$$

Hence, the curvature **C,** which is 1/r, is given as:

$$\mathrm{C} = \frac{2\mathrm{x}}{\mathrm{L}^2} \tag{5}$$

Consider Figure 10, which shows the geometry of the turning rover [45].

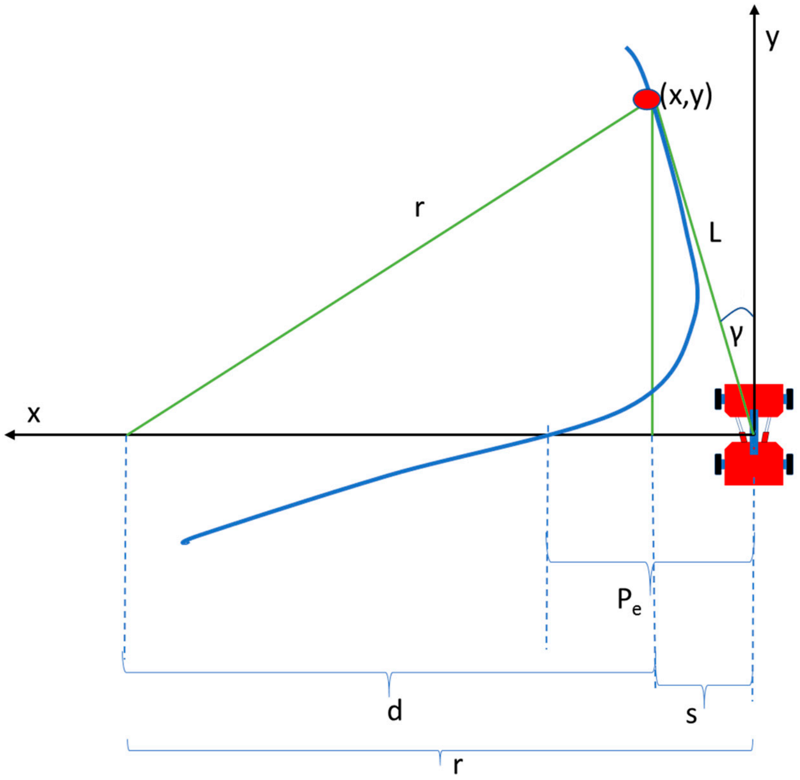

Figure 9. The geometry of the Pure Pursuit algorithm. Blueline is the designated path to pursue while red point (x,y) is the goal point. Black lines represent cartesian coordinates axis (y and x) while the green lines show the geometry of turning.

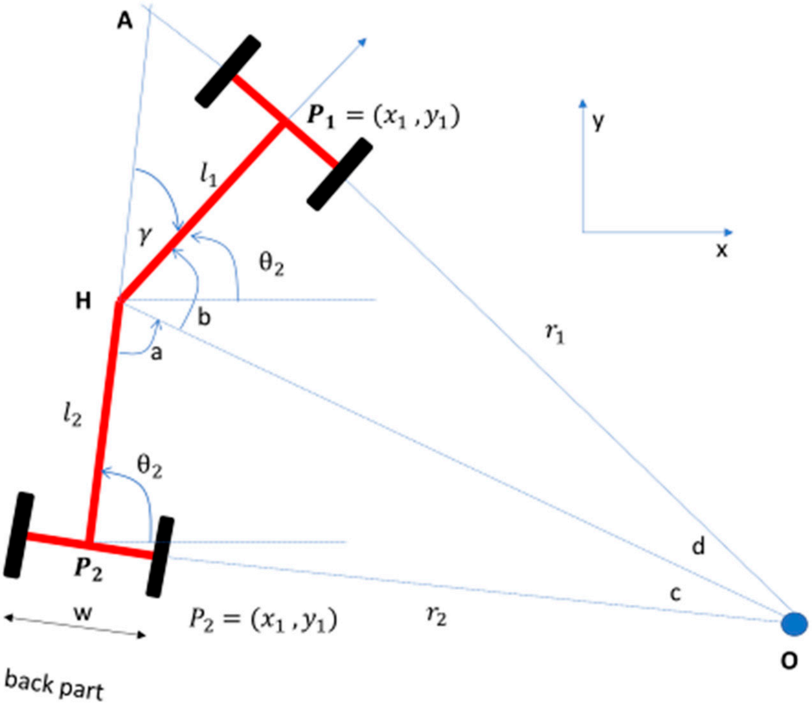

Figure 10. The geometry of the center-articulated rover while turning. $\mathbf{r}_1$ is the distance from the origin of the turning circle O [45].

At the point H, a + b + γ = 180°. So, b + d = 90° and a + c = 9°. Therefore, by substituting constants makes γ = c + d. γ is the articulation angle while θ is the heading angle.

Now, consider the right-triangle ΔOP_2A,

$$l_2 + \frac{l_1}{\cos\ \gamma} = r_2 \tan\ \gamma \tag{6}$$

$$(r_1 + l_1 \tan\ \gamma) \times \sin\ \gamma = l_2 + \frac{l_1}{\cos\ \gamma}$$

Then, by simplifying Equation (6), the turning radii, r_1 and r_2, can be found:

$$r_1 = \frac{l_1 \cos\ \gamma\ +\ l_2}{\sin\ \gamma} \tag{7}$$

$$r_2 = \frac{l_2 \cos\ \gamma\ +\ l_1}{\sin\ \gamma}$$

Since the rover was an approximately symmetric vehicle, the distance from the center (L_f) to back tires or front tires was 91 cm. hence, we can simplify the radii Equation above,

$$r\ =\ r_1 =\ r_2 = \frac{L_f \cos\ \gamma\ +\ L_f}{\sin\ \gamma} \tag{8}$$

Then, using trigonometry rules, sin 2a = 2sin a cos a and cos 2a = $2\cos^2 a$–1

$$\frac{r}{L_f} = \frac{\cos\ \gamma\ + 1}{\sin\ \gamma} = \frac{2\ \cos^2\frac{\gamma}{2}}{2\ \sin\frac{\gamma}{2}\ \cos\frac{\gamma}{2}}$$

Hence, by simplifying the trigonometry,

$$\tan\ \gamma\ = \frac{L_f}{r} \tag{9}$$

Then, insert Equation (4), which makes:

$$\tan\ \gamma\ = 2 \times\ L_f \times \left(\frac{x}{L}\right)\frac{1}{L} \tag{10}$$

$$\tan\ \gamma\ = 2 \times\ L_f \times (\sin\ \theta)\frac{1}{L}$$

$$\gamma\ = 2 \times\ \tan^{-1}\left(\frac{2 \times\ L_f \times \sin\ \theta}{L}\right)$$

So, Equation (10) shows the relationship of articulation angle γ to look-ahead distance L, half-length of the rover L_f, and heading angle θ. Consider that the vehicle at state (x_k, y_k) is articulating to the next goal (x_{k+1}, y_{k+1}). Next, the heading angle θ_{k+1}in relation to the current heading angle θ_k can be found by:

$$\theta_{k+1} = \theta_k -\ \tan^{-1}\frac{x_{k+1} -\ x_k}{y_{k+1} -\ y_k} \tag{11}$$

Because the rover has a slow turning action when moving, the horizontal distance of the vehicle from the goal should be increased by a factor "K" (See Equation (12)). The closest distance of the rover to the path is the path error (P_e). P_e is found by calculating the perpendicular distance of the vehicle to the designated path. The next position of the path (x_{k+1}) was modified by adding the factor K multiplied by P_e, becoming the modified Pure Pursuit algorithm of the center-articulated rover (Equation (12)). The value K is a rover-dependent factor that should be obtained by testing and experimentation to achieve the best path tracking.

$$x_{k+1} = x_{k+1} + K \times P_e \tag{12}$$

2.6. Proportional Control of the Articulation Angle

The rover turned to the target articulation angle γ, as described in Equation (10) by using proportional control. The current angle γ_k and required target angle γ_{k+1} was used to find the error that was used to control the movement (Figure 11). The gain, Kp, used was set to 1. The articulation angle was controlled by hydraulic cylinder linear actuators. The actuators were connected to two relays. Proportional control was used since the actuators were only controlled by on/off relays, which limited the ability to control actuator speed. The two relays were connected to the navigation controller digital pin 7 for the left actuator and digital pin 8 for the right actuator. The relays used to control the linear actuator were Single Pole Double Throw (SPDT) relays. SPDT relays provided a capability to control the linear actuators by switching into three different connections: normally closed, normally open, and common [46]. So, if the rover turned left, the left actuator retracted while the right actuator extended until a desired left articulation angle was achieved. Additionally, if the rover turned right, the right actuator retracted while the left actuator extended until a desired right articulation angle was achieved. It means that the circuit opened for the right actuator and closed for the left one. When the angle required was attained, it switched to common.

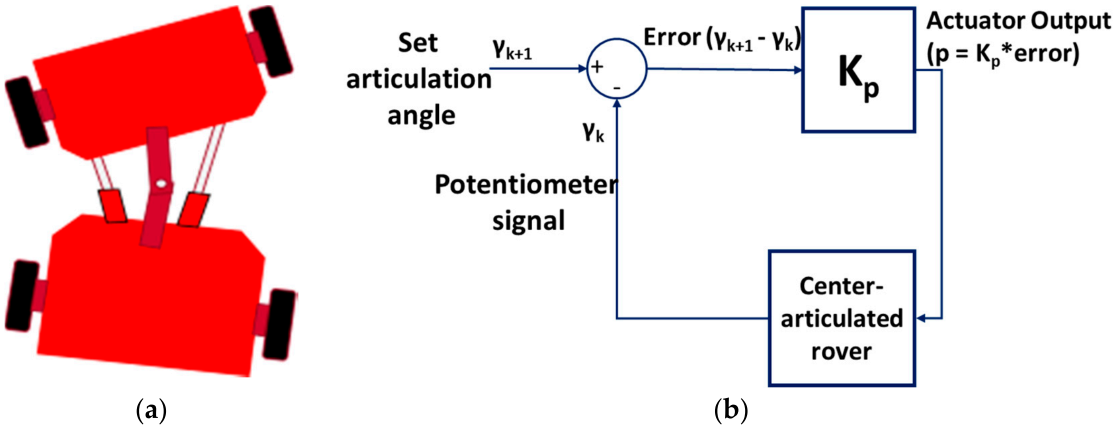

Figure 11. Proportional control of the articulation angle. (**a**) The rover when turning left; (**b**) the proportional control to achieve a targeted articulation angle.

2.7. Proportional Control of the Speed of the Rover

The speed of the rover was controlled by using the proportional controller (Figure 12). The controller had a gain of 22 and a target speed of 1.2 m/s. A PID controller was not implemented because we were not targeting precise speed control; hence, proportional control was enough. The controller calculated error from the difference between target speed and the speed estimated by the EKF from encoders, IMU, and GNSS readings. The acceleration of the rover was controlled by extending and retracting the linear actuator, which sets the swashplate angle. By changing the angle of the swashplate, hydraulic fluid flow rate to the hydraulic motors, turning the wheels was changed and effectively changed the speed of the rover. The vehicle remained stationary when the angle was set at 90°. When the angle of the swashplate changed from 90° to 60°, the vehicle moved backwards while increasing the speed to the minimum backwards speed. Additionally, when the angle of the swashplate changed from 90° to 120°, the vehicle moved forward with the increase in the maximum forward speed. However, the change in actuator movement required the swashplate angle to be more than 108° to move forward, or less than 80° to move backward, created by an inherent deadband in the pump performance. The deadband was caused by wear (leakage) in the hydraulic system and the mechanical compliance of connectors. The neutral position was held until the actuator extended or retracted by more than 2.5 cm.

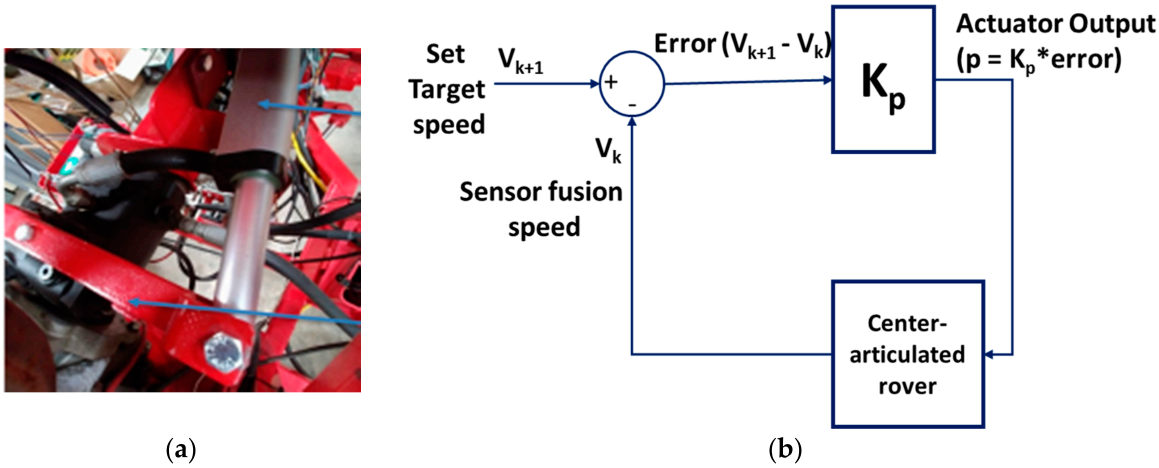

Figure 12. (**a**) The arrow points to the red arm that controls the swashplate angle. Another arrow points to the grey linear actuator controlled by the navigation controller (**b**) The proportional control diagram of the speed of the rover.

2.8. Waypoints Collection and Cubic Spline Interpolation of the Waypoints

The rover was driven to obtain the waypoints at the UGA grounds behind the engineering annex located at (31.475340 N, 83.528968 W) and the Entomology farm (31.472985 N, 83.531228 W) near Bunny Run Rd. in Tifton Georgia. The rover recorded the waypoints using RTK-GNSS at the rate of 5 Hz. Since the rate 5 Hz provided very few data points, the algorithm to interpolate the points using a cubic spline interpolation method was developed. The points were changed to UTM data points. Both fields were located at Zone 17R.

Cubic spline interpolation was done by assuming the data points were connected by a line whose Equation was a polynomial degree of three [47]. It was assumed that the datapoints given were [x_i, y_i], and no two points xi were equal to each other, and the x_i was in sequence such that $x_0 < x_1 < x_2 < \ldots < x_n$. The Spline function $S(x_i) = y_i$. For each subinterval [$x_{i-1} < x < x_i$], the cubic function is given as $C_i = a_i + b_i x + c_i x^2 + d_i x^3$. So, for every subinterval, the Spline function S(x) can be assumed as:

$$S(x) = \begin{cases} C_1(x), & x_0 < x < x_1 \\ C_2(x), & x_1 < x < x_2 \\ & \vdots \\ C_i(x), & x_{i-1} < x < x_i \\ & \vdots \\ C_n(x), & x_{n-1} < x < x_n \end{cases}$$

Algorithm 2 is used to calculate the value of a to calculate the values of a_i, b_i, c_i and d_i for every interval of the dataset. More values make it easy to calculate the relative position of the rover to the target path perpendicularly.

Algorithm 2: Cubic Spline Algorithm to estimate subinterval of UTM waypoints data intervals.

```
Input: $x_0, x_1, x_2, \ldots, x_n$; $a_0 = f(x_0)$, $a_1 = f(x_1)$, $a_2 = f(x_2)$, …… $a_n = f(x_n)$
Output: ai, bi, ci, di for j = 0,1,2, … ..,n-1
Assign $P_0 = 0$
Assign $Q_0 = 0$
Assign $R_0 = 0$
FOR j = 0 TO n-1
    Set the interval difference hi <- $x_{i+1}$ - $x_i$
    Set $\alpha_1 = (3/h_j)*(a_{j+1} - a_j) - (3*(a_j - a_{j-1})/h_{j-1})$
END FOR
FOR j = 1 TO n-1
    $P_j = 2*(x_{j+1} - x_{j-1}) - h_{j-1}Q_{j-1}$
    $Q_j = h_j / L_j$
    $R_j = (\alpha_j - h_{j -1}*R_{j-1})/P_j$
END FOR
Assign $P_n = 1$
Assign $R_n = 0$
Assign $c_n = 0$
FOR i = n-1 TO 0
//get the remaining values of the cubic spline b,c, and d
    $c_i = Z_i - Q_i * c_{i+1}$
    $b_i = (a_{i+1} - a_i)/h_i - h_i*(c_{i+1} + 2*c_i)/3$
    $d_i = (c_{i+1} - c_i)/3*h_i$
Return all the values of ai, bi, ci, di
```

2.9. Preliminary Experiment

Preliminary experiments were conducted on 10th October 2019 at the UGA grounds to study the navigation behavior of the rover, when parameters such as ROS update rates, look ahead, and path error were altered. These tests served to calibrate the system to perform well in the field. The preliminary experiment involved four tests:

1. A fast ROS rate was set to 10 Hz;
2. A short look ahead was set at 1 m;
3. Path error was set to 0 which means K x $P_e = 0$;
4. An optimal condition was set (long look-ahead is 3 m, path error is 1.5 times path error and slow ROS rate at 1 Hz).

The experiment was conducted by setting the rover to follow the prescribed path. The predefined path was obtained by moving the rover manually and collecting the GNSS waypoints. Later, the rover was set to autonomous mode to follow the pre-planned path so that behavior and characteristics could be observed and tuned.

2.10. Field Experiment

The field experiment was conducted at the Horticulture hill farm (31.472985N, 83.531228W) near Bunny Run Road in Tifton, Georgia, after establishing the calibrated parameters of the rover. The field (Figure 13) was planted on 19th June 2019 using a tractor (Massey Fergurson MF2635 tractor, AGCO, Duluth, GA, USA) and a 2-row planter (Monosem planter, Monosem Inc, Edwardsville, KS, USA). The cotton seeds (Delta DP1851B3XF, Delta & Pine Land Company of Mississippi, Scott, MS, USA) were planted every two-rows and skipped two rows. The rows were 36-inch (91.44 cm) wide, and the seed spacing was 4-inch (10.16 cm). The cotton field was undefoliated, and most of the cotton bolls were open already at the time of the experiment. Three tests were conducted for navigation on 18.5 m rows on 21st October 2019. The path was obtained by driving the rover over one of the two rows of cotton plants and collecting the waypoints (Figure 6). The experiments were conducted by setting the rover to follow the predefined path autonomously.

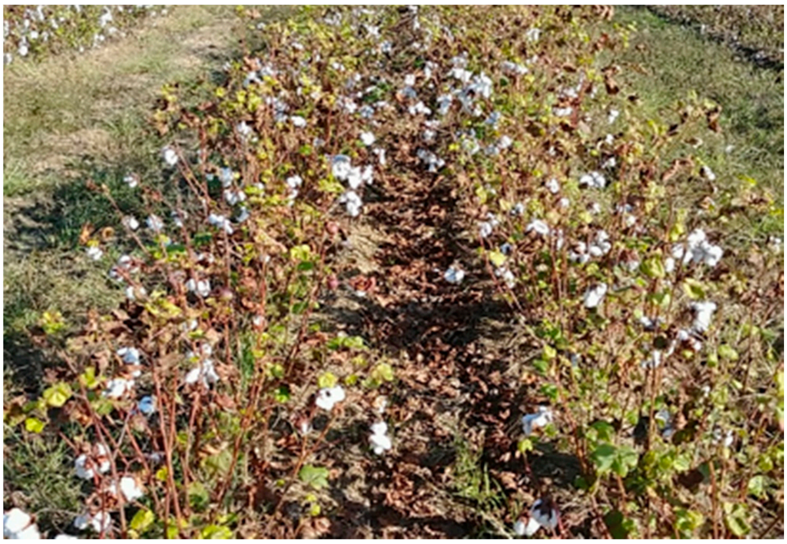

Figure 13. The appearance of the cotton at the time of the field experiment.

3. Results and Discussions

3.1. Preliminary Experiment

The results of the preliminary experiments show that the rover navigation tracking was negatively affected when ROS update rates were increased, or no path error correction was applied, and when very short look-ahead was used (Figure 14). The pure pursuit algorithm has a goal to make sure that the rover regains the designated path by articulation when it loses where it was on the predefined waypoints. Fast ROS update rates affected system performance since the mechanical responses of the machine were slow compared to the update rate provided by the controller. This short time between the new input reading of the system meant that the reaction time of the vehicle to the control signal was slow, and the vehicle reaction was always lagging behind the control decision, causing the rover to lose tracking control. The ROS rate update was a significant parameter if it was not set right. The performance was obtained with short look-ahead distances. With a short look ahead distance, the rover tried to move quickly to regain the path it has lost. However, this action caused the rover to overshoot the path and oscillate along the prescribed path.

With no path error corrections (when K was Zero), the rover could never converge to the path over time. Figure 14c shows how the vehicle was not able to converge to the path, which means the error was consistently maintained. To avoid this behavior, modified pure pursuit increased the error of the system by 1.5 using Equation (12) to force the system to converge to the path. If the error was significant, the system amplified the error forcing the rover to act aggressively and quickly. When the error was small, the system acted slowly because the amplification of the error also became small (Refer to Equation (12)).

3.2. Field Experiments

Figure 15 shows the navigation path traces, as recorded by the GNSS for the third experiment. The rover performed well visually. The third pass can clearly show that the rover moves out of the path during the end-of-row turning. Contributing factors were wheel slipping as the rover attempted to turn compounded by control signal updates requiring turning too quick for the rover to respond and follow the designated path.

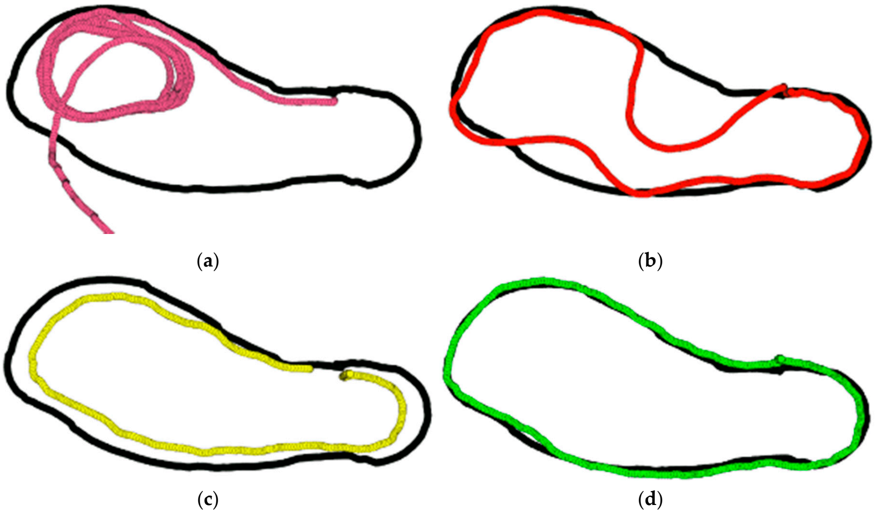

Figure 14. Performance of the rover when the (**a**) ROS updates were high (10 Hz), (**b**) Short look ahead distance (1 m), (**c**) no path error correction was applied and (**d**) successful path tracking when look-ahead was 3 m, K was 1.5, and ROS updates were at 1 Hz. Blacklines are the predefined waypoints, while colored lines represent the rover passes for each condition from 1 to 4.

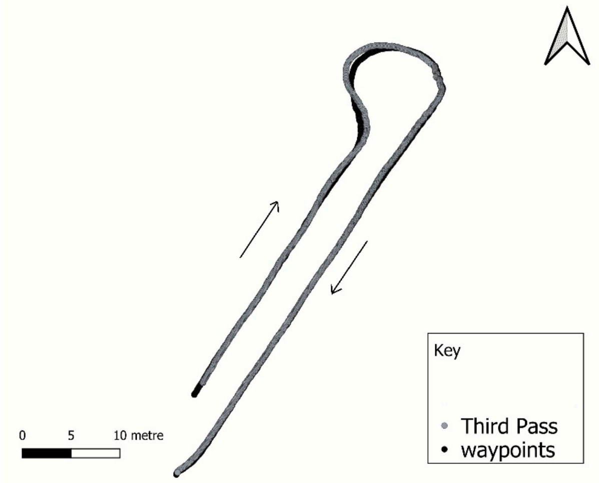

Figure 15. Path tracking of the prescribed path (black pass). The gray pass is the GPS generated path of the rover when following the rows using the prescribed path (black). The gray path trace is the third navigation pass experiment. The arrows show the passing direction of the rover.

The rover was able to follow the path and return to enter the next cotton row. The absolute error distribution was determined to characterize rover navigation behavior. The rover performed well, as Figure 16 shows most of the path errors were less than 15 cm (0.15 m).

Figure 16 and Table 1 show that the rover performance was adequate as it was safely navigating along the rows without driving over the plants. Most of the errors were less than 10 cm from the prescribed path. The rover converged back to the predefined path when there was an error, except during the turning maneuver, which provided a significant challenge to the rover. The first and second tests were excellent, but the third one had trouble turning in the muddy end-row that caused the wheels to slide. The MAE of 0.042 m, 0.062 m and 0.091 m for the first, second, and third pass, respectively, show that the system performance was acceptable. However, the MAE was significant when turning as it ranged at 0.233 m, 0.227 m and 0.244 m for the first, second, and third pass, respectively (Figure 17). Slippage had effects only in the third pass when turning caused errors to increase to 0.115 m in the second pass. Additionally, the results show a modified simple pursuit algorithm was struggling to compensate for errors in the third pass. However, the magnitude of the errors measured when turning did not inhibit overall performance for harvesting.

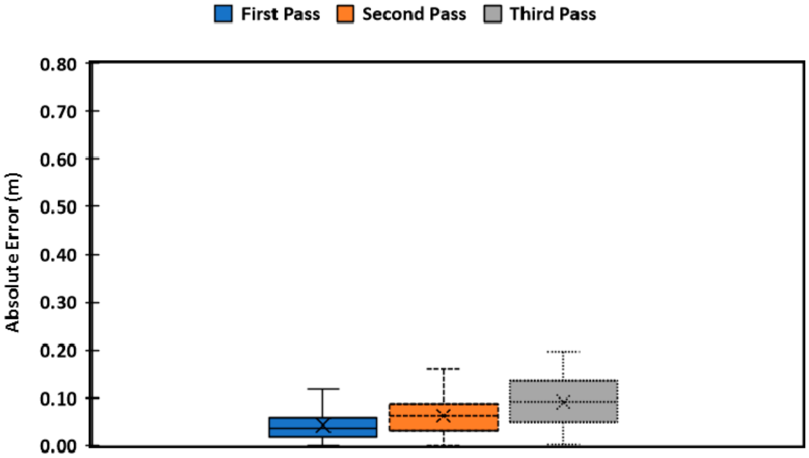

Figure 16. The average of absolute errors (A.E.) in the first, second, and third non-turning passes (blue, orange, and gray boxes, respectively). The boxes present the first quartile to the third quartile of A.E., while the whiskers show maximum values and minimum values of each of the passes. The "x" shows the mean absolute error (MAE) for each of the passes.

Table 1. The mean absolute error and standard deviations of the three passes along the cotton rows.

Mean ± Std. Dev (m)	1st Pass	2nd Pass	3rd Pass	Overall
1St Row	0.048 ± 0.036	0.048 ± 0.035	0.066 ± 0.0046	0.053 ± 0.041
Turning	0.233 ± 0.198	0.227 ± 0.211	0.244 ± 0.211	0.235 ± 0.206
2nd Row	0.036 ± 0.024	0.081 ± 0.028	0.115 ± 0.043	0.070 ± 0.046
Overall (1st and 2nd Rows)	0.042 ± 0.032	0.062 ± 0.036	0.091 ± 0.053	0.061 ± 0.044

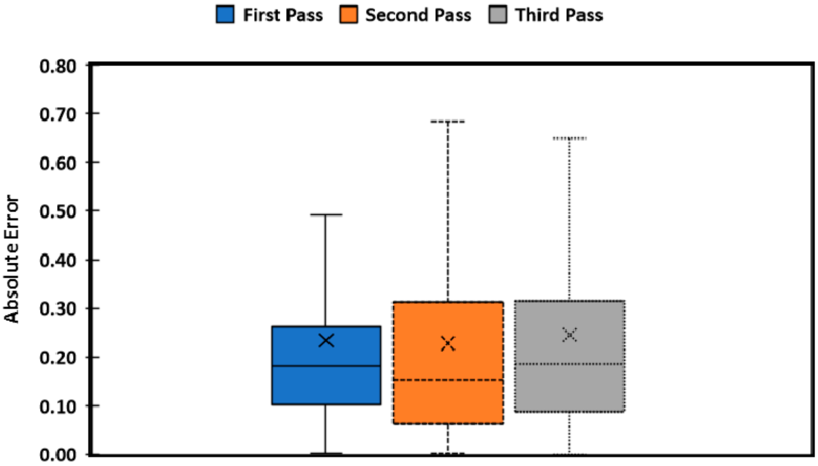

Figure 17. The average of absolute errors (A.E.) when turning for the first, second, and third passes (blue, orange, and gray boxes, respectively). The rover had no significant difference in turning performance. The boxes present the first quartile to the third quartile of A.E., while the whiskers show maximum values and minimum values of each of the passes. The "x" shows the mean absolute error (MAE) for each of the passes.

4. Conclusions

An autonomous navigation algorithm using a modified pure-pursuit algorithm for an MPR was designed, developed, and tested in this study. The results showed that the rover could autonomously navigate safely along the rows of cotton, turn around, and enter a second row. Results from the preliminary testing and field testing showed that an affordable single-frequency RTK-GPS could be used with other sensors and a sensor fusion technique to achieve acceptable navigation accuracy. There was an increase in errors when the rover performed turns that did not impede the rover's ability to enter the next row. As a result, the MPR can follow rows and operate autonomously to perform any number of tasks when a predefined path in GNSS coordinates is available or created. In the future, small, intelligent, multi-purpose vehicles that can autonomously navigate on the field could provide paths and prescriptions for spraying, planting, scouting, or harvesting. Small rovers would eventually need to operate in teams to cover larger acreages with rover-to-rover communication to create built-in task optimization.

Author Contributions: Conceptualization, K.F. and G.R.; Methodology, K.F. and G.R.; formal analysis, K.F.; software, K.F.; Writing—Original Draft Preparation, K.F.; Writing—Review and Editing, K.F., W.P, C.L. and G.R.; supervision, K.F., W.P, C.L. and G.R.; Project Administration, W.P., E.B. and G.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Cotton Incorporated, Agency Assignment Number 17-038.

Acknowledgments: The authors are very thankful for project support from Cotton Incorporated, Agency Assignment #17-038. We also thank Ricky Fletcher and Gary Burnham for their helpfulness and technical support in building the rover platform and collection of data. Additionally, we thank Logan Moran for the Field Testing of the rover.

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

References

1. Hayes, L. Those Cotton Picking Robots. 2017. Available online: http://georgia.growingamerica.com/features/2017/08/those-cotton-picking-robots (accessed on 29 August 2019).
2. Fue, K.G.; Porter, W.M.; Rains, G.C. Real-Time 3D Measurement of Cotton Boll Positions Using Machine Vision Under Field Conditions. In Proceedings of the 2018 BWCC, San Antonio, TX, USA, 3–5 January 2018; pp. 43–54.
3. UGA Cotton News. *2019 Georgia Cotton Production Guide*; U.E. Team, Ed.; UGA Extension Team: Tifton, GA, USA, 2019.
4. USDA/NASS. 2017 State Agriculture Overview for Georgia. 2018. Available online: https://www.nass.usda.gov/Quick_Stats/Ag_Overview/stateOverview.php?state=GEORGIA (accessed on 29 August 2019).
5. Duckett, T.; Pearson, S.; Blackmore, S.; Grieve, B.; Chen, W.-H.; Cielniak, G.; Cleaversmith, J.; Dai, J.; Davis, S.; Fox, C.; et al. Agricultural robotics: The future of robotic agriculture. *arXiv* **2018**, arXiv:1806.06762.
6. Rains, G.C.; Faircloth, A.G.; Thai, C.; Raper, R.L. Evaluation of a simple pure pursuit path-following algorithm for an autonomous, articulated-steer vehicle. *Appl. Eng. Agric.* **2014**, *30*, 367–374.
7. Fue, K.G.; Porter, W.M.; Barnes, E.M.; Rains, G.C. An Extensive Review of Mobile Agricultural Robotics for Field Operations: Focus on Cotton Harvesting. *Agric. Eng.* **2020**, *2*, 10. [CrossRef]
8. Reiser, D.; Sehsah, E.-S.; Bumann, O.; Morhard, J.; Griepentrog, H.W. Development of an Autonomous Electric Robot Implement for Intra-Row Weeding in Vineyards. *Agriculture* **2019**, *9*, 18. [CrossRef]
9. Auat Cheein, F.; Steiner, G.; Perez Paina, G.; Carelli, R. Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection. *Comput. Electron. Agric.* **2011**, *78*, 195–207. [CrossRef]
10. Cheein, F.A.A.; Carelli, R.; Cruz, C.D.L.; Bastos-Filho, T.F. SLAM-based turning strategy in restricted environments for car-like mobile robots. In Proceedings of the 2010 IEEE International Conference on Industrial Technology, Vina del Mar, Chile, 14–17 March 2010.
11. Ouadah, N.; Ourak, L.; Boudjema, F. Car-Like Mobile Robot Oriented Positioning by Fuzzy Controllers. *Int. J. Adv. Robot. Syst.* **2008**, *5*, 25. [CrossRef]
12. Higuti, V.A.H.; Velasquez, A.E.B.; Magalhaes, D.V.; Becker, M.; Chowdhary, G. Under canopy light detection and ranging-based autonomous navigation. *J. Field Robot.* **2019**, *36*, 547–567. [CrossRef]

13. Xue, J.; Zhang, L.; Grift, T.E. Variable field-of-view machine vision based row guidance of an agricultural robot. *Comput. Electron. Agric.* **2012**, *84*, 85–91. [CrossRef]
14. Farzan, S.; Hu, A.-P.; Davies, E.; Rogers, J. Modeling and control of brachiating robots traversing flexible cables. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018.
15. Grimstad, L.; From, P.J. The Thorvald II Agricultural Robotic System. *Robotics* **2017**, *6*, 24. [CrossRef]
16. Xiong, Y.; Peng, C.; Grimstad, L.; From, P.J.; Isler, V. Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper. *Comput. Electron. Agric.* **2019**, *157*, 392–402. [CrossRef]
17. Ramin Shamshiri, R.; Weltzien, C.; Hameed, I.A.; Yule, I.J.; Grift, T.E.; Balasundram, S.K.; Pitonakova, L.; Ahmad, D.; Chowdhary, G. Research and development in agricultural robotics: A perspective of digital farming. *Int. J. Agric. Biol. Eng.* **2018**, *11*, 1–11. [CrossRef]
18. Ball, D.; Upcroft, B.; Wyeth, G.; Corke, P.; English, A.; Ross, P.; Patten, T.; Fitch, R.; Sukkarieh, S.; Bate, A. Vision-based obstacle detection and navigation for an agricultural robot. *J. Field Robot.* **2016**, *33*, 1107–1130. [CrossRef]
19. Kayacan, E.; Kayacan, E.; Chen, I.M.; Ramon, H.; Saeys, W. On the comparison of model-based and model-free controllers in guidance, navigation and control of agricultural vehicles. In *Type-2 Fuzzy Logic and Systems*; John, R., Hagras, H., Castillo, O., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; pp. 49–73.
20. Tu, X.; Gai, J.; Tang, L. Robust navigation control of a 4WD/4WS agricultural robotic vehicle. *Comput. Electron. Agric.* **2019**, *164*, 104892. [CrossRef]
21. Boubin, J.; Chumley, J.; Stewart, C.; Khanal, S. Autonomic computing challenges in fully autonomous precision agriculture. In Proceedings of the 2019 IEEE International Conference on Autonomic Computing (ICAC), Umea, Sweden, 16–20 June 2019.
22. Wang, H.; Noguchi, N. Adaptive turning control for an agricultural robot tractor. *Int. J. Agric. Biol. Eng.* **2018**, *11*, 113–119. [CrossRef]
23. Liakos, G.K.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine Learning in Agriculture: A Review. *Sensors* **2018**, *18*, 2674. [CrossRef] [PubMed]
24. Coulter, R.C. *Implementation of the Pure Pursuit Path Tracking Algorithm*; Technical Report; Defense Technical Information Center: Fort Belvoir, VA, USA, 1 January 1992.
25. Backman, J.; Oksanen, T.; Visala, A. Navigation system for agricultural machines: Nonlinear model predictive path tracking. *Comput. Electron. Agric.* **2012**, *82*, 32–43. [CrossRef]
26. Gupta, N.; Khosravy, M.; Gupta, S.; Dey, N.; Crespo, R.G. Lightweight Artificial Intelligence Technology for Health Diagnosis of Agriculture Vehicles: Parallel Evolving Artificial Neural Networks by Genetic Algorithm. *Int. J. Parallel Program* **2020**, 1–26. [CrossRef]
27. Fulop, A.-O.; Tamas, L. Lessons learned from lightweight CNN based object recognition for mobile robots. In Proceedings of the 2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 24–26 May 2018.
28. Chennoufi, M.; Bendella, F.; Bouzid, M. Multi-agent simulation collision avoidance of complex system: Application to evacuation crowd behavior. *Int. J. Ambient. Comput. Intell.* **2018**, *9*, 43–59.
29. Vougioukas, S.G. A distributed control framework for motion coordination of teams of autonomous agricultural vehicles. *Biosyst. Eng.* **2012**, *113*, 284–297. [CrossRef]
30. Jasiński, M.; Mączak, J.; Szulim, P.; Radkowski, S. Autonomous agricultural robot-collision avoidance methods overview. *Zesz. Nauk. Inst. PojazdóW* **2016**, *2*, 37–44.
31. Gupta, S.; Khosravy, M.; Gupta, N.; Darbari, H.; Patel, N. Hydraulic system onboard monitoring and fault diagnostic in agricultural machine. *Braz. Arch. Biol. Technol.* **2019**, *62*, 1–15. [CrossRef]
32. Gupta, S.; Khosravy, M.; Gupta, N.; Darbari, H. In-field failure assessment of tractor hydraulic system operation via pseudospectrum of acoustic measurements. *Turk. J. Electr. Eng. Comput. Sci.* **2019**, *27*, 2718–2729. [CrossRef]
33. Gupta, N.; Khosravy, M.; Patel, N.; Dey, N.; Gupta, S.; Darbari, H.; Crespo, R.G. Economic data analytic AI technique on IoT edge devices for health monitoring of agriculture machines. *Appl. Intell.* **2020**, 1–27. [CrossRef]
34. Carpio, R.F.; Potena, C.; Maiolini, J.; Ulivi, G.; Rosselló, N.B.; Garone, E.; Gasparri, A. A Navigation Architecture for Ackermann Vehicles in Precision Farming. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1103–1110. [CrossRef]

35. Gao, X.; Li, J.; Fan, L.; Zhou, Q.; Yin, K.; Wang, J.; Song, C.; Huang, L.; Wang, Z. Review of Wheeled Mobile Robots' Navigation Problems and Application Prospects in Agriculture. *IEEE Access* **2018**, *6*, 49248–49268. [CrossRef]
36. Gao, Y.; Cao, D.; Shen, Y. Path-following control by dynamic virtual terrain field for articulated steer vehicles. *Veh. Syst. Dyn.* **2019**, 1–25. [CrossRef]
37. Koubâa, A. *Robot Operating System (ROS)*; Springer: Berlin/Heidelberg, Germany, 2017.
38. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009.
39. Samuel, M.; Hussein, M.; Mohamad, M.B. A review of some pure-pursuit based path tracking techniques for control of autonomous vehicle. *Int. J. Comput. Appl.* **2016**, *135*, 35–38. [CrossRef]
40. Bergerman, M.; Billingsley, J.; Reid, J.; van Henten, E. Robotics in Agriculture and Forestry. In *Springer Handbook of Robotics*; Siciliano, B., Khatib, O., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 1065–1077.
41. Botterill, T.; Paulin, S.; Green, R.; Williams, S.; Lin, J.; Saxton, V.; Mills, S.; Chen, X.; Corbett-Davies, S. A Robot System for Pruning Grape Vines. *J. Field Robot.* **2017**, *34*, 1100–1122. [CrossRef]
42. Wan, E.A.; Nelson, A.T. Dual extended Kalman filter methods. *Kalman Filter. Neural Networks* **2001**, *123*, 123–173.
43. Moore, T.; Stouch, D. A generalized extended kalman filter implementation for the robot operating system. In *Intelligent Autonomous Systems*; Strand, M., Dillmann, R., Menegatti, E., Ghidoni, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 335–348.
44. Post, M.A.; Bianco, A.; Yan, X.T. Autonomous navigation with ROS for a mobile robot in agricultural fields. In Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Madrid, Spain, 26–28 July 2017.
45. Corke, P.I.; Ridley, P. Steering kinematics for a center-articulated mobile robot. *IEEE Trans. Robot. Autom.* **2001**, *17*, 215–218. [CrossRef]
46. Oberhammer, J.; Tang, M.; Liu, A.-Q.; Stemme, G. Mechanically tri-stable, true single-pole-double-throw (SPDT) switches. *J. Micromech. Microeng.* **2006**, *16*, 2251. [CrossRef]
47. McKinley, S.; Levine, M. Cubic spline interpolation. *Coll. Redwoods* **1998**, *45*, 1049–1060.