2018

Navigation of an Autonomous Differential Drive Robot for Field Scouting in Semi-structured Environments

Douglas Freese
University of Central Florida

NAVIGATION OF AN AUTONOMOUS DIFFERENTIAL DRIVE ROBOT FOR FIELD SCOUTING IN SEMI-STRUCTURED ENVIRONMENTS

by

DOUGLAS FREESE
B.S. University of Central Florida, 2011
M.S. University of Central Florida, 2014

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Mechanical and Aerospace Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2018

Major Professor:
Yunjun Xu

ABSTRACT

In recent years, the interests of introducing autonomous robots by growers into agriculture fields are rejuvenated due to the ever-increasing labor cost and the recent declining numbers of seasonal workers. The utilization of customized, autonomous agricultural robots has a profound impact on future orchard operations by providing low cost, meticulous inspection. Different sensors have been proven proficient in agrarian navigation including the likes of GPS, inertial, magnetic, rotary encoding, time of flight as well as vision. To compensate for anticipated disturbances, variances and constraints contingent to the outdoor semi-structured environment, a differential style drive vehicle will be implemented as an easily controllable system to conduct tasks such as imaging and sampling.

In order to verify the motion control of a robot, custom-designed for strawberry fields, the task is separated into multiple phases to manage the over-bed and cross-bed operation needs. In particular, during the cross-bed segment an elevated strawberry bed will provide distance references utilized in a logic filter and tuned PID algorithm for safe and efficient travel. Due to the significant sources of uncertainty such as wheel slip and the vehicle model, nonlinear robust controllers are designed for the cross-bed motion, purely relying on vision feedback. A simple image filter algorithm was developed for strawberry row detection, in which pixels corresponding to the bed center will be tracked while the

vehicle is in controlled motion. This incorporated derivation and formulation of a bounded uncertainty parameter that will be employed in the nonlinear control. Simulation of the entire system was subsequently completed to ensure the control capability before successful validation in multiple commercial farms. It is anticipated that with the developed algorithms the authentication of fully autonomous robotic systems functioning in agricultural crops will provide heightened efficiency of needed costly services; scouting, disease detection, collection, and distribution.

Dedicated to my family and friends.

ACKNOWLEDGMENTS

I would like to first acknowledge and thank my advisor, Dr. Yunjun Xu for his guidance and insight during our project as well as all of my committee members; Dr. Aman Behal, Dr. Jeffrey Kauffman, and Dr. Kuo-Chi Lin. I would also like to thank Tom Minter, owner of Pappy's Patch and Florida Pesticide Research for all of his collaborative effort and assistance. I would also like to acknowledge the many senior design students who worked on our robot and specifically, Pablo Menendez-Aponte all of whom helped bring DDAGR to life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Precision agricultural has become a vastly expanding field over the past several decades. The concept involves utilization of mechatronic systems such as robots or semi-intelligent vehicles to improve the capable efficiency of agrarian businesses. Predominately, tasks that are routine as well as easily repeatable have been the focus of this automation. Some of these include land preparation, chemical dispersion, crop inspection and recently more difficult tasks such as harvesting. The focus of this research is the design and development of an autonomous ground vehicle that will conduct disease detection tasks.

Currently, diseases are found in agricultural crops (specifically strawberry) mostly by visual inspection. The task can be tedious, costly and time consuming and consistent repeatable results are not guaranteed. Hyperspectral technology advancements have recently offered opportunities for specimen health analysis at an earlier stage of growth for a variety of other field crops including the likes of wheat and tomatoes [1], [2]. For the application into the large strawberry domain that exists in Florida, the challenge then becomes transportation of the sensor efficiently throughout the field.

Strawberry farms can be considered semi-structured environments, where the plants grow on top of elevated patterned rows. Significant sources of disturbances such as uneven

terrain and soil densities as well as weed growth will make the control and navigation tasks difficult for any unmanned vehicle. **Chapter** 1 will describe in more detail the developed technologies that have extended into agricultural applications as well as more of the specific intricacies of strawberry farms. Next, discussion of the vehicle design highlighting the mechanics and software precedes the description of the vehicle kinematics and control approach for when the vehicle is traveling over one of the strawberry rows.

The most difficult portion of the safe travel for the autonomous system will be in the headland at the end of each row. For the localization task during this portion, machine vision was incorporated in a feedback configuration with non-linear type controllers. These control algorithms were developed, simulated and tested in the real field terrain as presented in the experimental validation chapter. Finally, reachability analysis was conducted to verify the robotic vehicle would adequately converge to the desired location at the conclusion of each control portion, with the inclusion of anticipated disturbances such as measurement noise and control input uncertainty. The pinnacle point of the research is the integrated field tests in which the vehicle travels over multiple rows safely and efficiently as presented in the later portion of **Chapter** 6. Conclusively this navigation based research has the potential to expand to virtually any agricultural province that involves elevated row structures.

CHAPTER 1

BACKGROUND

1.1 Motivation and Literature Review

Of late, robotics and automation have permeated to the field of agronomy and cultivation, popularly referred to as precision agriculture. As mechanization of critical tasks for large scale crop production becomes increasingly easier due to innovative robotic design, higher processing power, and improved reliability, overall farm efficiency is projected to increase significantly. Autonomous systems also provide the opportunity for reducing labor costs by performing repetitive tasks that can be highly susceptible to human error due to fatigue. Some such tasks involve inspection of fields for disease, chemical disbursement, land preparation, and harvesting [3], [4].

One of the most demanding tasks for an autonomous vehicle is traversal of the agricultural fields with a margin of exactness, for example, on the scale of one degree for heading and several centimeters for distances [5]. The most popular sensing devices used are GPS, machine vision, encoding, inertial and ranging derivatives, varying substantially in estimation methodology, precision, and cost. Low cost range finding has proven an accurate and feasible solution when referencing a structured row or crop. Accuracy of up to 2.4° heading

angle and 4 cm was achieved in [6] for a vehicle travelling in-between maize crop rows; while tracking in white asparagus crops was reported as ±0.5 cm in [7]. Line tracking using an enhanced carrier phase differential GPS system as reference was accurate to several centimeters for a farm tractor travelling in a bumpy field in [8]. While the above mentioned GPS technology has been prevalent for a couple decades, the cost to achieve a specific location is substantial. This is especially true for farm scenarios with restricting patterned paths, smaller travel space, and unpredictable row conditions, as is expected in strawberry fields. Fusion of multiple sensory data such as GPS, inertial and odometry, is also a viable option for tracking the vehicle path shown in [9], [10] and [11]. If a GPS signal degrades as discussed in [12], other methods of referencing such as simultaneous localization and mapping are feasible. Also mentioned is the importance of data selection and filtering for the laser scanner device utilized, due to the real test environments. Machine vision has become an increasingly popular navigation device used in agricultural environments [13], [14], and several centimeters of accuracy have been verified in [15], [16], [17] and the vehicle operating speed is up to 3 m/s in capricious farm environments including soybean, cultivated stubble as well as citrus. In [18] the developed navigation algorithm is robust enough to operate sufficiently in apple orchards with well structured tree rows as well as less structured rows with grassy terrain. Distinct image processing algorithms have been exercised for agricultural navigation e.g. the Hough Transform (HT) and the morphological and RGB image filtering methods. The key issue of machine vision is dealing with substantially varying lighting conditions [19], [20]. HT is robust to environmental disturbances and has a low

computational cost [21], [22], [23], [24], [25]. Morphological and RGB image filtering are also viable methods of detection and have proven better reliability than the HT gradient methods (wheat, soybean and maize) [6], [26] however, the required processing time is significantly higher [27], [28], [29].

However, all the aforementioned sensing devices have drawbacks if they are blindly adopted in agricultural operations. GPS precision can be limited to several meters unless a high fidelity and expensive package is used. IMUs are subjected to drifting characteristics despite lack of an input. Encoder counting methods are susceptible to error due to wheel slip, which is a common occurrence in the variable terrain types of commercial farms. Range finding techniques use physical references however they are influenced by sensor orientation [30] and can be jeopardized by dirt or leaves. Image or video sensors are sensitive to light conditions and need to be re-calibrated whenever the working environment changes.

Based on available sensing information, different controllers have been investigated to move a robotic system according to its assigned task. Proportional, integral and derivative (PID) control was the first tested technique in [31], [32]. It has extended into field testing for traveling over rows of white asparagus in [7], as well as a variety of other crops, e.g. wheat, sorghum and chickpea as in [33]. Fuzzy logic which makes use of predefined sets and discretized states to establish the control commands is successful in driving an autonomous vehicle down rows and into the headland area of emulated maize rows [34]. By combination of fuzzy logic and PID type control, the lateral offset as well as yaw rate error of an autonomous tractor is substantially reduced as shown in [35]. In the visual servo algorithm of [36], the

end of the maize crop rows are detected and compared to a GPS reference to conclude the traversal portion through the field. Afterwards, the vehicle performs an ordered sequence of motions to align with the next desired row. Another example of agrarian field navigation using vision feedback for the control is shown in a vineyard crop [37]. The Lyapunov stability was proven for the controller and the vehicle traveled between rows stably without diverging. However, the proposed method may be limited to functioning in patterned fields with large distances between rows and where the tracking error can exceed nearly twice the vehicle dimension. Nonlinear controllers have also been introduced into the agricultural domain to tackle inherent vehicle unmodeled dynamics, uncertainty and disturbances [38]. In [39], a moving horizon estimation and nonlinear model predictive control is applied to guide a tractor safely given bumpy and slippery conditions of the grassy test field. Sliding mode control has also been implemented onto a large farm tractor which was subjected to wheel slip [40] [41]. The control is stable and trajectory error is bounded despite the disturbances and terrain, similarly, the overall accuracy is sufficient for the large tractor in the given field. An analytical model is developed in [42] for improving the lateral control performance of a tractor traveling at relatively high speeds for spraying applications.

It is a common practice to split the control task into individual regions with prescribed initial conditions and restrictions based on the vehicle position and configurations [43] [44]. In [34] an Ackermann style vehicle uses fuzzy logic combined with Lidar techniques to navigate the robot down corn fields and into the headland. Vision combined with GPS is used for guidance in [33] for a robotic cart in patterned crop fields. The vehicle is capable of traveling

through the headlands to execute turns, however it does not perform row tracking during this portion. For the strategies of using different controllers in specific regions, reachability analysis is an effective method of determining autonomous vehicle location and orientation over time [44] with the objective of safely guided motion. This is because in real environments, disturbances as well as uncertainty exist for both the initial states, terrain conditions, and control inputs [45]. It was discussed in [46] that large uncertainties of the system can lead to unexpected terminal states and therefore failure of the system to remain within the desired region. A tube-based model predictive control has been investigated by restricting trajectories sequentially to tolerable regions for an off road mobile robot [47]. This method has also been introduced for an autonomous agricultural robot comprising of a tractor and coupled trailer in [48]. The navigation error was on the order of centimeters thus proving the system robustness to uncertainties and other interactions.

1.2 Semi-structured Agrarian Environment

The autonomous robot is designed to perform in commercial strawberry orchards. Florida had 10,900 Acres of strawberry farms in 2014 with an average farm size of 40 acres [49]. Utilizing the raised bed plasticulture method, commercial farms typically have long symmetric rows patterned throughout as seen in **Fig.** 1.1. The dimension of a typical strawberry orchard is the most important factor to be considered in the vehicle design. Following the raised bed growth technique [50], Florida farms typically have the specifications shown in

Figure 1.1: Sample strawberry rows in commercial farms

Table 1.1. The averages and variations listed here are established from our observations and discussions with strawberry growers at multiple farms.

Some of the technicalities of strawberry fields are as follows: (1) they are elevated row crops with definite dimensions, (2) field terrains can fluctuate substantially from soft loamy sand to firm grass patches including holes and mounds, (3) semi-structured form leads to different control objectives when over the rows and transferring between rows. Strawberry farms use repetitive closely aligned rows for ease of production therefore, the control objective while traveling over the row is comparison of the vehicle center and row center. Which is considered a translational offset error, and should not deviate substantially from zero, to prevent collision with bed walls on either side. The difference between the vehicle angle and fixed bed orientation, is referred to as heading offset error. A smaller offset angle is desired despite varying bed widths at different farms. As the strawberry rows are not perfect the distance between each may alter slightly and also for different farms based on the bed formation techniques. The control accuracy will need to be on the same order of magnitude as compared to other agricultural vehicles. Translational accuracies will be within several

Table 1.1: Strawberry Orchard Dimensions and Variances

Strawberry Row Parameters	Average Dimension (cm)	Variance +/- (cm)
Bed Width	70	5
Bed Height	20	5
Walking Path	50	10
Plant Height	18	4
Plant Diameter	20	6
Headland Length	300	50

centimeters while the robot heading angle will require limitation in each phase to several degrees. While GPS has proven successful for navigation purposes, the cost for navigation proficiency required for strawberry field dimensions is significantly higher. Furthermore, it is labor prohibitive to log the position information of each and every row and strawberry bush. Inertial type sensors must use some form of sensor fusion to negate the effects of drift. Thus, direct distance sensing utilizing low-cost range finding provides adequate information while the robot is over the row. For the second special condition in strawberry fields based on field observations, disturbances, uneven and softer terrain will degrade the tracking capability particularly odometry, therefore control techniques such as PID and fuzzy logic will be limited to when the vehicle is over a row. Vision based control typically operates at a lower frequency but will provide a higher accuracy for orientation, which is critical for aligning with the next desired row while low cost GPS will not be sufficient for control reference. Machine vision

combined with robust nonlinear control methods will deliver stable vehicle motion despite the variety of disturbances that are anticipated. Inclusively, these detailed sensors and control approaches will provide the required navigation information for performing special tasks in the semi-structured strawberry farm environment.

It is now imperative to model the field conditions that will influence the vehicle model and motion performance. These restrictions are sculpted from typical assumptions pertinent to an autonomous robotic system [51] and agronomical environment [52], thus form constraint formulae for specific regions of the farm. The assumptions mentioned here are inherited from a paper currently under review titled "Vision Based Nonlinear Robust Scouting Control in Agricultural Fields with Reachability Analyses."

Assumption 1 (A1): Intuitively the vehicle speed will be set by bounded limits for each portion and due to the low speed and vehicle mass, drag effects will be neglected, $\nu_{\min} \leq \nu \leq \nu_{\max}$ and $\omega_{\min} \leq \omega \leq \omega_{\max}$. $[\nu, \omega]^T$ are the translational and rotational control inputs respectively that will be described in detail later.

Assumption 2 (A2): The focal lengths of the camera have the following uncertainties: $f_x = (1 + \Delta_{11})\hat{f}_x$ and $f_y = (1 + \Delta_{22})\hat{f}_y$, in which $\hat{f}_x$ and $\hat{f}_y$ are calibrated nominal focal lengths of the camera. The Δ terms will be numerically established by experimentation.

Assumption 3 (A3): A strawberry bed height can be approximated as a constant due to its relatively small variance. Therefore, a planar model is adopted and as mentioned in [21] and pitching and rolling of the vehicle will assumed to be minimal.

Assumption 4 (A4): Since the safest and most power-efficient mode of over-bed travel of the robot is translation ($\omega = 0$), whenever the vehicle offset errors are within a tolerance in the over-bed motion, only a translational command will be given.

Assumption 5 (A5): Variability of field terrain will induce a range of wheel slippage during vehicle motion, which will be bounded as $\Delta\nu_{\min} \leq \Delta\nu \leq \Delta\nu_{\max}$ and $\Delta\omega_{\min} \leq \Delta\omega \leq \Delta\omega_{\max}$. Intuitively, the amount of slippage occurring during pure translation will be small compared to a pure rotation.

Assumption 6 (A6): Elevated crop row widths can vary several centimeters, thus the travel path width on each side of the row will be bounded by $\Delta RF_{row,min} \leq \Delta RF_{row} \leq \Delta\ RF_{row,max}$, as a range finder estimated variance of the strawberry bed width. Similarly, device resolution and noise from these sensors as $\Delta RF_{noise,min} \leq \Delta RF_{noise} \leq \Delta\ RF_{noise,max}$ will be estimated from variance of dynamic measurements.

Assumption 7 (A7): In the cross-bed motion, the heading angle is assumed to be very small, except during the 90°. turn. This is reasonable since the vehicle is moving in straight paths during these portions except for the turn.

All of the bounds listed for the constraints will be restricted to ensure convergence and stability within the environment yet relaxed enough considering dynamics of the mechanical system as well as physical limitations of the travel paths.

1.3 Contributions

The contributions of this study are summarized here: (1) a multi-phase control structure and algorithms are proposed based on the strawberry orchard environment and operation needs; (2) a simplified uncertainty bound is obtained as there is uncertainty associated with the focal length of the camera which could significantly reduce the conservativeness of the adopted nonlinear robust controllers; and (3) to the best knowledge of the authors, this is the first experimentally validated controller for an agricultural robot to transverse throughout a commercial strawberry field.

The rest of the dissertation is organized as follows; in **Chapter** 2 the Disease Detecting Agricultural Ground Robot (DDAGR, the name was proposed by the 2013-2014 senior design team) hardware and software is described in detail. **Chapter** 3 explains the over-bed motion control and navigation architecture while **Chapter** 4 pertains to the cross-bed control that utilizes vision based feedback in conjunction with the nonlinear controllers. **Chapter** 5 highlights the preliminary experimental results including control and measurement particularities. Reachability analysis and the combined navigation tests of **Chapter** 6 preclude the formulation for future work and conclusions in **Chapter** 7.

CHAPTER 2

DISEASE DETECTING AGRICULTURAL GROUND VEHICLE

2.1 Motion Control Outline

To solve the motion tasks of the autonomous vehicle, splitting of control task into separate regions will prove a feasible solution. This approach is necessary due to the specific field conditions and variances. **Fig.** 2.1 displays an overview of the motion tasks and navigation methodology implemented by the autonomous system. Phases 1 consists of motion over the strawberry rows in which the vehicle will travel large distances to reach specific plants. Phase 2 indicates 90 degree in-site rotation of the vehicle as it aligns to travel in headlands while off the rows. Phases 3 is split into two sections, while the vehicle travels to an adjacent row. The first sub-phase (3.1) consist of a linear translation and subsequent specialized control to reduce the offset of the vehicle center with respect to the row center. After the intermediate rotation is performed the orientation as well as distance away from the desired row will need to be reduced. Sub-phase 3.2 will simultaneously generate linear and rotational control to produce arcing motion to reduce the orientation and distance respectively.

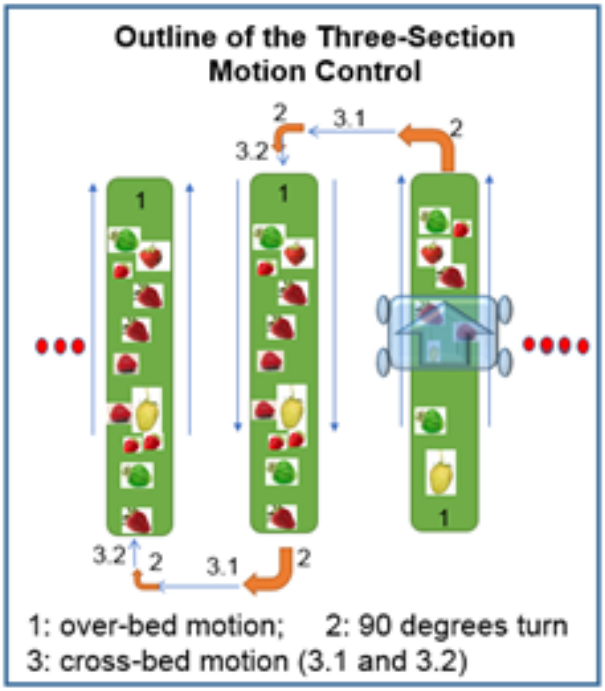


Figure 2.1: Motion control phases

2.2 Hardware, Sensors and Software

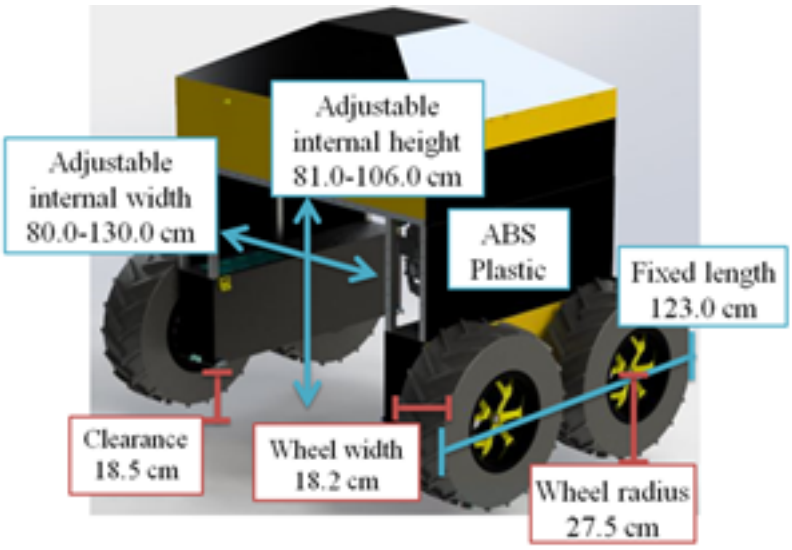
The distinct characteristics of the autonomous ground robot are summarized via excerpts from the author's previously published papers [53], [54]. It is first important to enumerate the hardware design specifications and requirements. The following list of requirements was established based upon the previously mentioned challenges and field specifications for commercial strawberry farms:

- Dimensions of the vehicle must allow it to travel over a variety of raised beds considering the mentioned variations.
- Substantial motor torque as well as large contact surface between the vehicle drive and ground is necessary to ensure significant traction in varying terrains.
- An electronics suite for data collection and integration of all the information and interfacing with the control for autonomous navigation is required.
- The vehicle must traverse the field in a timely manner to cover a reasonable area.
- Multi-spectral technology will be incorporated into the vehicle to perform the disease detection task efficiently and accurately.

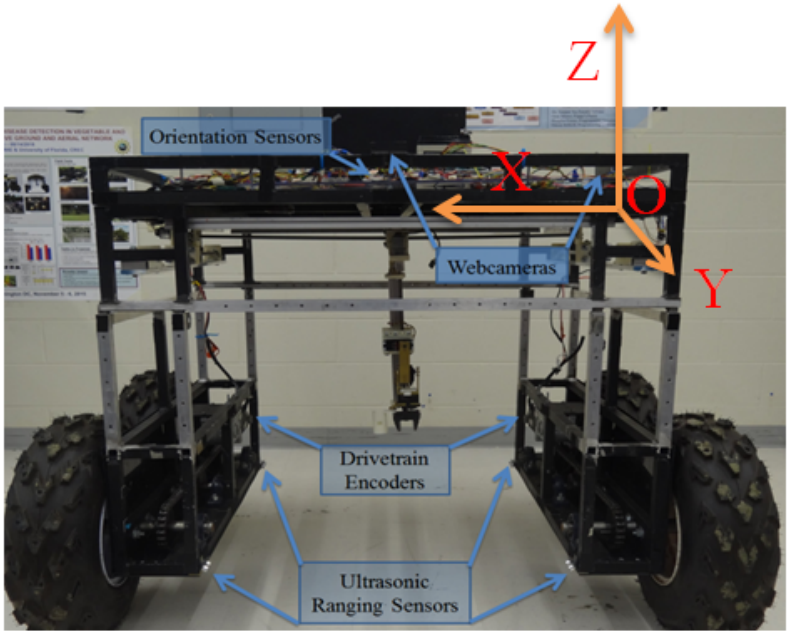
In order to accommodate the strawberry bed variances of commercial farms, the robot frame has been customized to be manually adjustable in the X and Z axes (so called a transformer), by telescoping of the frame as indicated in **Fig.** 2.2(a), as the adjustable

dimensions. DDAGR can expand its width up to an additional 50 cm and its height up to an additional 25 cm while retaining vehicle symmetry which is critical for navigation algorithms. Contingent to the second bullet point, the robot drivetrain is designed in the differential skid-steer style with two drive banks (DG-158 24VDC motors capable of 135 RPM) and has a total of four wheels. Due to the vehicle weight and potential sinkage from soft soil, a large wheel radius of 27.5 cm as well as 18.2 cm width, is employed ensuring a larger contact surface area for the torque transfer. Shown in **Fig.** 2.2(b), Four ultrasonic range finding sensors operate at the interior frame corners of DDAGR by measuring the time delay of a wave reflected from the edge of the strawberry bed. An effectual measurement angle from the distance sensor is 15 degrees outward in which the dynamic accuracy was determined to be ± 0.5 cm for distances between 2 and 500 cm. The robot roll, pitch and yaw as well as rotation rates about the fixed inertial frame are determined by the IMU. Encoder as well as IMU data is streamed through all phases of the experiment for reference. Two pairs of mono-vision webcameras will be oriented around the upper portion of the robot frame to provide an imaging plane that is parallel to the robot frame.

The third requirement is critical for extracting data from multiple sensors and generating control instructions. Therefore, the software package is designed and developed to execute communication, information transmission, navigation, and controller command functionalities of DDAGR. The overall data flow, device wiring and hardware are depicted in **Fig.** 2.3. Software programmed in MATLAB gives commands and handles function calls, makes calculations, as well as completes processing of the data. Software developed in Arduino C



(a)



(b)

Figure 2.2: (**a**) Solidworks ® rendition of DDAGR and dimensions (**b**) Actual vehicle and sensor locations

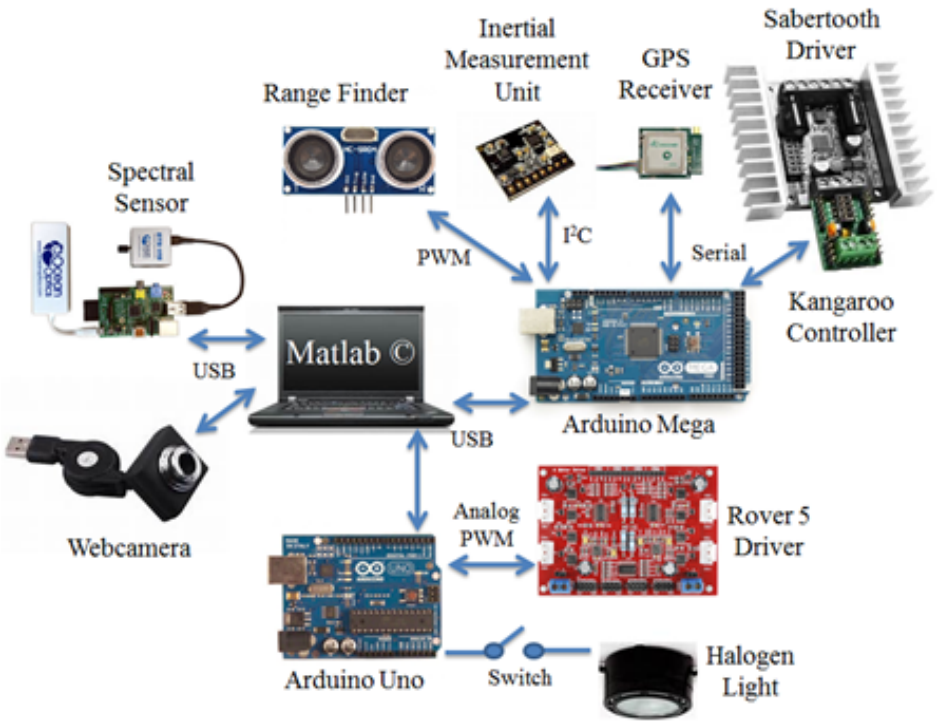


Figure 2.3: Electronics and communication methodology

is enacted to initiate and trigger all the peripheral low level devices. The laptop computer and Arduino microprocessor correspond with each other serially by Universal Asynchronous Receiver Transmitter (UART) via USB hookup. The microprocessor (Arduino hardware) auxiliary channels are then linked to sensors and actuators for operation. The assignment of the microchip is to gather, compile and transmit sensory feedback data to the laptop CPU and receive instructions or commands to implement the actuators or motors.

The fourth requirement for the agricultural robot is to travel efficiently throughout the grounds. Based on the average field size, sensor update rates, and typical speeds of manned agricultural vehicles, a desired velocity capability of 15 cm per second was pursued. Considering control calculation as well as the range finder (RF), GPS and IMU rates, the targeted average cycle time for navigation is 4 Hz during the over-bed control region and 1 Hz while transferring rows in the cross-bed vision based segment. Matlab is implemented as the control software for ease of integration of other subsystems including the communication with peripheral sensors which is accomplished through Arduino C.

Although not the focus of this dissertation is important to justify the robot design and purpose. As discussed in [55] crop loss in strawberry can be as high as fifty percent, even in well managed farms. The overall objective and concept is to introduce early stage detection and identification of these infections before visual inspection is feasible, and subsequently prevent propagation of the disease [56]. A spectral sensor is a device that measures electromagnetic radiation reflection from surfaces, it can work in both the visual as well as infra-red region and will produce response plots of the reflectance amplitude with respect

to wavelengths. As discussed in [57] distinct amplitude variances in both the visual and near infra-red spectrum are discernible for fungal based infections in wheat crops allowing for classification of the plant as healthy, asymptomatic or symptomatic. This research has expanded to applications such as strawberry where Anthracnose, one of the primary concerns in central Florida, was detected with an accuracy of 80% even in the early incubation stages [58]. It is therefore ideal to utilize this methodology as a preventative measure in commercial strawberry farms and the task now becomes safe transport and implementation of the spectral sensors.

CHAPTER 3

OVER-BED MOTION CONTROL

3.1 Differential Drive Robot Kinematics

The information presented here is adapted from the paper currently under review titled "Vision Based Nonlinear Robust Scouting Control in Agricultural Fields with Reachability Analyses." The model from [59], [60] is adopted as the governing equation for DDAGR, a differential drive style skid steer mobile robot. All four wheels have an equal radius, and are on fixed direction axles with respect to the robot frame. Inherently, this will result in only rotational motion of the wheel around the axle and not along it. The coordinate information of DDAGR is shown in **Fig**. 3.1. The global coordinate is referred as x_g and y_g, and the z axis is formed by the right hand rule. The heading angle of DDAGR is represented by θ . The robot local reference frame is denoted x_r and y_r, which rotates from the global coordinate with an angle of e_θ referred to as the heading error. The distance between the tread centerlines of the wheel banks in the y_r axis is L. The distance of DDAGR center to the bed center, considered the translational offset error, is denoted by e_y and is calculated based on the RF readings. The rotational angle between the DDAGR local coordinate with respect to the global coordinate, e_θ, is calculated using the geometric relation similar to that

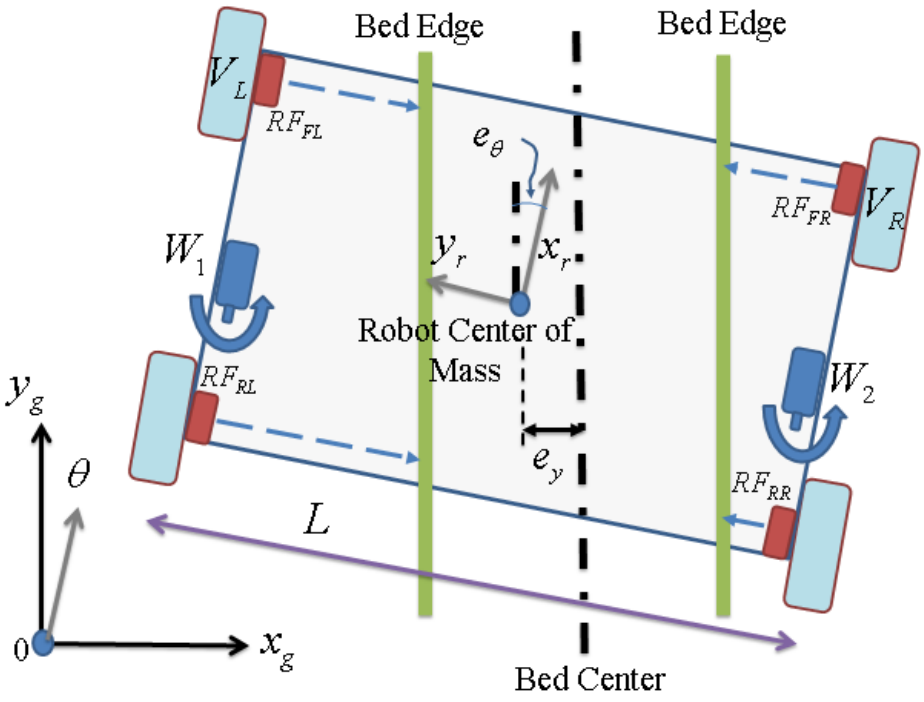


Figure 3.1: DDAGR coordinates, kinematics and navigation design

in [7] as

$$e_\theta = 0.5\left[\tan^{-1}\left(\frac{|RF_{FR} - RF_{RR}|}{L}\right) + \tan^{-1}\left(\frac{|RF_{FL} - RF_{RL}|}{L}\right)\right] \tag{3.1.1}$$

in which the subscripts FR, RR, FL, and RL denote the RF in the front right, rear right, front left, and rear left, respectively. It is worth noting that the absolute value in 3.1.1 ensures a positive angle and therefore no cancellation from each pair. Consequently the direction is established from comparison of the right vs the left range finder reading. The distance of the vehicle center to the bed is calculated by

$$d = \frac{RF_{FR} - RF_{FL}}{2} + \frac{RF_{RR} - RF_{RL}}{2} \tag{3.1.2}$$

This equation applies to both the forward or reverse travel direction, as it is a simple step to invert the control command if traveling in reverse as opposed to redefining the navigation structure. The offset of the vehicle center to the bed centerline, e_y , is then computed by

$$e_y = d\cos e_\theta \tag{3.1.3}$$

Both of the offset error terms from equations 3.1.1 and 3.1.3 will be utilized in the over-bed controller. The drivetrain outputs motor angular velocities $W_j,\ j = 1, 2$ as shown in the figure, which are converted to the wheel velocities ($V_i,\ i = 1, 2, 3, 4$) using the wheel radius R_i and the reduction gear ratio η as

$$\begin{aligned} &V_i = R_i W_j/\eta,\ \ i = 1, 2, 3, 4; j = 1, 2 \\ &V_1 = V_3 = V_R;\ \ V_2 = V_4 = V_L \end{aligned} \tag{3.1.4}$$

The velocity of each wheel bank is combined to produce the overall translational and rotational velocity with respect to the vehicle center of mass (COM) and are defined as ν and ω

respectively, therefore

$$\begin{aligned} v &= (V_R + V_L)/2 \\ \omega &= (V_R - V_L)/L \\ u &= [v, \omega]^T \end{aligned} \tag{3.1.5}$$

Naturally, V_R and V_L are the right and left wheel bank velocities, in which the subscripts R and L denote the wheels on the left and right sides of DDAGR. The kinematic model for the drivetrain based on [61] can be written as 3.1.6

$$\begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \tag{3.1.6}$$

Here $\dot{x}_g(t) \in \Re^1$ and $\dot{y}_g(t) \in \Re^1$ are the differentiated cartesian coordinates of the robot in the global frame while $\dot{\theta}(t) \in \Re^1$ depicts the derivative of the orientation. The combined control vector $u(t) \in \Re^2$ is defined as $u = [v, \omega]^T$ and is C^1 .

The possible motions for the drivetrain are translation, in-site rotation, and logarithmic spiral [62]. The linear translation is performed by commanding the same motor speed for both wheel banks and in the same direction. By instructing one of the wheel banks to rotate at a larger or smaller rate than the opposing bank, the logarithmic spiraling motion is achieved. The in-site rotation operation is fulfilled by rotating the banks in opposite directions at matched rotation rates, which will occur at the end of a row, or if the robot needs to fix its orientation.

3.2 Over-bed Algorithm

For the motion of the vehicle while over the strawberry rows (Phase 1) a desired travel distance down the row is first established as: $e_x = x_d - x_a$. Where $\{e_x, x_d, x_a\} \in \Re^1$, e_x is the travel distance error, x_d is the desired travel distance (preprogrammed or GPS), x_a is the encoder measured travel distance. The translational velocity gain is computed as: $v = k_{p_x} e_x$ From tuning k_{p_x} was set equal to 1. When the velocity exceeds a defined value, it is set to a limit before employed in the control as $\nu > \nu_{\text{lim}} \rightarrow \nu = \nu_{\text{lim}}$ The turn velocity ω is created from a simple PID structure [7], [32].

$$\omega = k_p e + k_i \int_{t_0}^{t} e dt + k_d \dot{e} \tag{3.2.1}$$

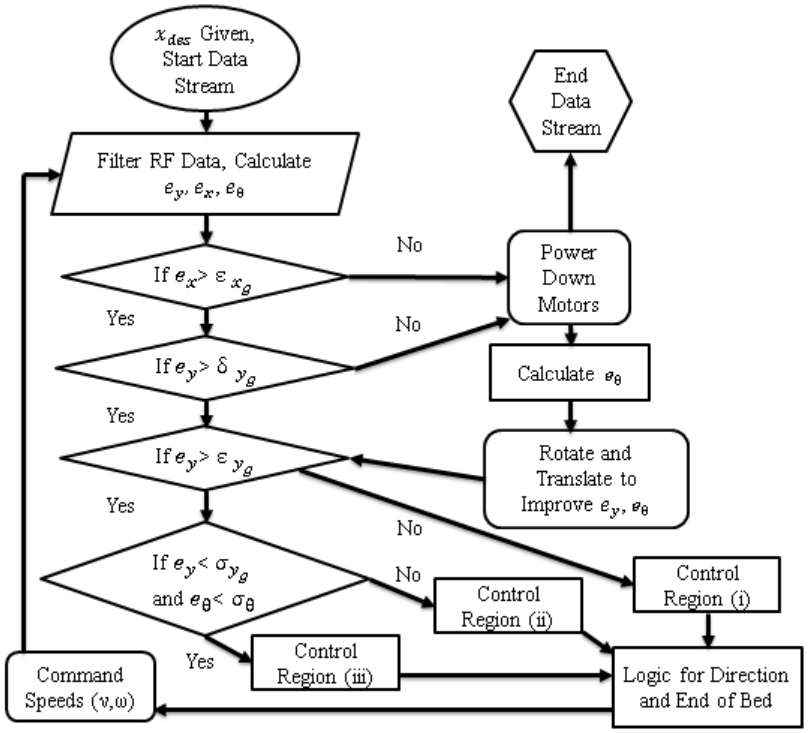
Where an integral term is generated by utilizing the Euler method. The tunable proportional, integral and derivative gains are k_p k_i and k_d respectively. The error term e will be constructed from e_θ and e_y from equations 3.1.1 and 3.1.3 as well as user defined weights w_θ and w_y.

$$e = w_\theta e_\theta + w_y e_y \tag{3.2.2}$$

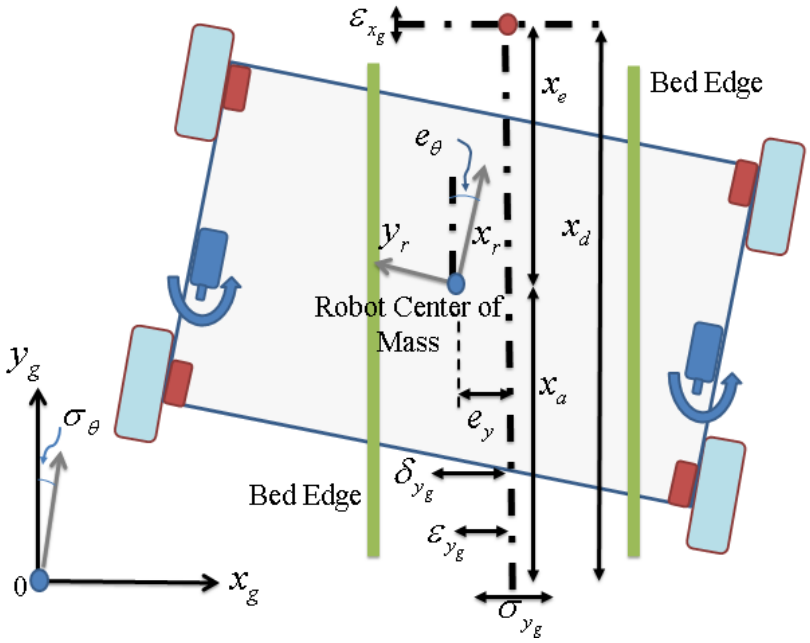
Feasibility of a weighting method to produce the error term for the controller is exhibited in [44]. The weights are necessary as the controller is designed to function given the variability between field dimensions at different farms, and e_θ or e_y will be more critical for specific fields.

Control was implemented in the following algorithm structure observing the assumptions of Section 1.2. 1) ω_{max} was imposed for vehicle safety considering A1, 2) measurement errors will contribute to state uncertainty ΔRF while environmental disturbances generate

control input error $\Delta\nu, \Delta\omega$ which reflects A5-A6. 3) If the vehicle is adequately aligned with the strawberry row it is most efficient to drive straight, ie. $\omega = 0$ as mentioned by A4. In the algorithm, all of the inequalities are user defined tolerances that may vary for specific field circumstances. Initially, a desired drive distance is supplied as x_d and the translational velocity ν is set to the speed limit. The overall algorithm can be seen in **Fig**. 3.2(a) depicted as a flow chart such as in [63]. Range finder data is then filtered using a zero-order hold filter and smoothed to remove erroneous readings. Both the distance errors e_x and e_y are continuously checked to determine when to turn off the motors. If e_x is within the tolerance ϵ_x , the vehicle has traveled far enough down the row to its desired location. As a safety feature, if the robot is very close to a bed wall, within the halt tolerance i.e., $e_y \leq \delta_y$ it needs to stop, reduce the heading error e_θ and then move away from the row before continuing. If e_y is not outside of the turn tolerance ϵ_y , the vehicle is too close to a bed wall, then it needs to turn. Contingent to A1, the turn rate ω will be fixed to a constant value and is referred to as Control Region i in the diagram. Subsequently, if e_y and e_θ are outside of an alignment tolerance σ_y and σ_θ , respectively, justified by assumptions A5 and A6, the PID calculated control command 3.2.1 will be implemented as referred to be Control Region ii. Alternatively, if the robot is within these bounds and respecting Assumption 4, the turn rate will be set to 0 resulting in a constant speed translation or Control Region iii. Ordering of the control regions is presumed based on the assumptions. It is important to note that the PID gains will be tuned conservatively enough to compensate for disturbances however the vehicle may still approach the bed wall too close and therefore the turn rate maximum is ideal



(a)



(b)

Figure 3.2: (a) Over-bed motion control algorithm (b) Labelled parameters

and necessary. Simple logic checks are enacted to establish travel direction e.g. (forward or reverse) as well as clockwise or counterclockwise to adjust the offset errors. The algorithm also contains the capability of detecting when the robot has reached the end of a row, and will attempt to reduce e_θ before leaving the row. Finally, the calculated command speeds are sent to the controller and the loop is repeated. When the controller has reduced the x error offset below the defined threshold, the vehicle has either traveled far enough down the row and reached the desired destination or is at the end of the bed.

To properly asses the stability of the entire control system, at the conclusion of an over bed section the final states must be measured and verified as settled within the mandatory range.

Remark 1. It is therefore required that $e_{\theta,\min} \leq e_\theta \leq e_{\theta,\max}$ as well as $e_{y,\min} \leq e_y \leq e_{y,\max}$ for this over-bed control region. These final state parameters will require modification at different farms for varying strawberry row dimensions and accordingly adjustment of the vehicle dimensions.

CHAPTER 4

CROSS-BED MOTION CONTROL

4.1 Image Processing

This Chapter pertains to content presented by the author at the 2017 Dyncamic Systems and Control Conference [54]. Also included is material from a paper currently under review titled "Vision Based Nonlinear Robust Scouting Control in Agricultural Fields with Reachability Analyses" For control phases 3.1 and 3.2, information from images collected by webcameras mounted to the autonomous robot will generate navigation references. Conducive to utilizing camera images as a means of course-plotting, feature detection must be accurately employed [23], [28]. A control objective such as tracking is also necessary and will govern the vehicle motion based on the interest points extracted from images. An algorithm was developed specifically for operation in strawberry orchards based on the semi-structured environment as well as the distinct colors of the plants and elevated rows. The first step is capturing an image of the desired strawberry row. Next the image is cropped to reduce the processing time and localized to a region of interest [26] [64].

The image is filtered by comparing the pixel intensity of each individual channel (RGB) to a determined threshold. Then based on the comparison a single channel pixel is

set to either black or white, commonly referred to as binarization [65]. Area and centroid calculations are made from the Matlab® Image Processing toolbox on the filtered image. As multiple rows may be visible in any given image, the returned pixel coordinates are processed to match with the desired row the robot is to align with. The method is discussed in more details in the authors work [54]. Finally, the pixel values are passed to be used in the control derivations. The steps involved in the centerline detection are illustrated in **Fig**. 4.1.

4.2 Camera Model

Using the image information for vehicle guidance necessitates formulation of the camera model to collect and reconstruct output pixels. It will first be critical to simulate the controllers, by formation of the full camera matrix which can be used to convert cartesian coordinates of a detected object or point to pixel coordinates and vice versa. The position bias of the camera from the robot center of gravity is denoted by $b = \begin{bmatrix} b_x & b_y & b_z \end{bmatrix}^T$ as shown in **Fig**. 4.2, ($b_x = 0$) The location of the row centerline points (red circles) in the world frame are assumed to be $x_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T$, $i = 1, 2$ where z_i will be taken as a constant estimated from field data and is valid by consideration of Assumption 3. These terms form the first portion of the extrinsic matrix identifying the camera location in the world coordinate system. It is assumed that there are uncertainties associated with the camera

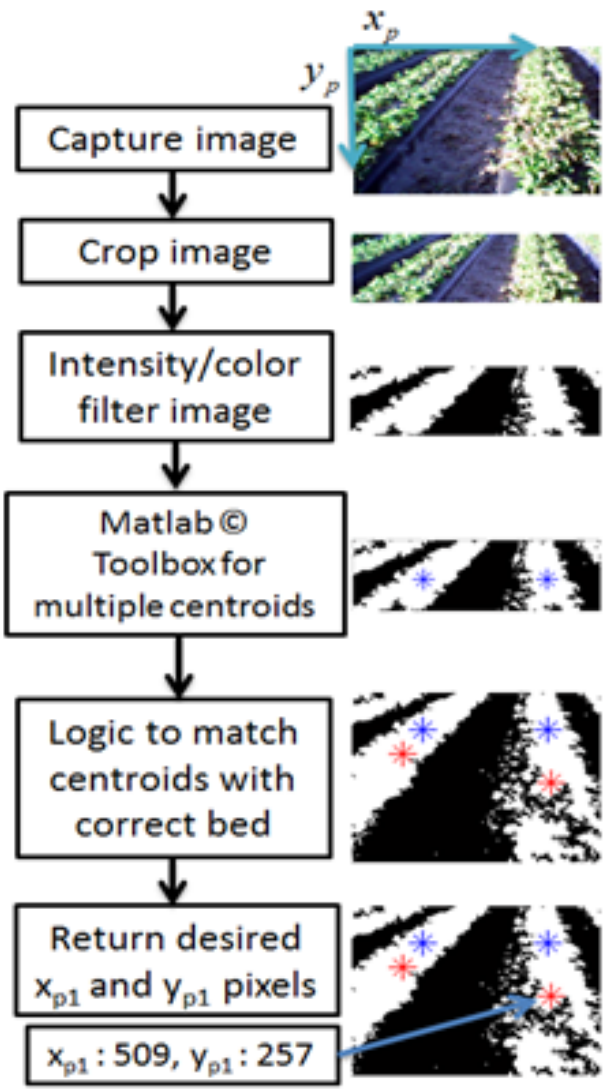


Figure 4.1: Steps involved in the centerline points detection of strawberry rows [54]

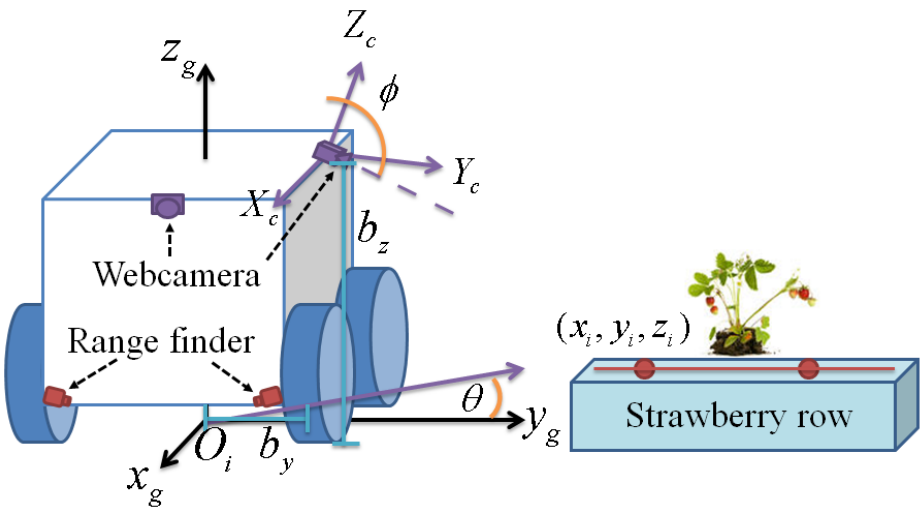


Figure 4.2: Coordinates and camera pose as vehicle prepares for control phase 3.1. In control phase 3.2, the camera mounted to the front (along x global axis) will face the strawberry row

intrinsic matrix.

$$K = \begin{bmatrix} f_x + \Delta_{fx} & 0 & c_x \\ 0 & f_y + \Delta_{fy} & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2.1}$$

in which Δ_{fx} and Δ_{fy} are the bounded uncertainties associated with the focal lengths f_x and f_y , respectively from Assumption 2. After the calibration [66], the following parameters are obtained for the selected camera, $f_x \approx f_y \approx 1040$ pixels $c_x = 320$ and $c_y = 240$ pixels

respectively. Based on [67], the projected points to camera coordinates are

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = K R_\phi^X R_\theta^Z \begin{bmatrix} b_x - x_g + x_i \\ b_y - y_g + y_i \\ b_z - z_g + z_i \end{bmatrix} \tag{4.2.2}$$

in which

$$R_\phi^X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \tag{4.2.3}$$

and

$$R_\theta^Z = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2.4}$$

where $[\ x \quad y \quad z\ ]^T$ is the position state of the vehicle, and z is assumed to be constant based on Assumption 3 (A3) from **Section** 1.2. The coordinate frame rotation angles ϕ and θ are depicted in **Fig**. 4.2 and comprise rotations of the camera in the world frame as the second portion of the extrinsic matrix. The scalar form of Eq. (4.2.2) to convert three-dimensional

world coordinates to three-dimensional camera coordinates is

$$\begin{aligned} X_c &= (f_x c\theta + c_x s\varphi s\theta)(b_x - x_g + x_i) \\ &\quad +(f_x s\theta - c_x s\varphi c\theta)(b_y - y_g + y_i) \\ &\quad +c_x c\varphi(b_z - z_g + z_i) \\ Y_c &= (c_y c\varphi + f_y s\varphi)(b_z - z_g + z_i) \\ &\quad +s\theta(c_y s\varphi - f_y c\varphi)(b_x - x_g + x_i) \\ &\quad +c\theta(c_y s\varphi - f_y c\varphi)(b_y - y_g + y_i) \\ Z_c &= c\varphi(b_z - z_g + z_i) + s\varphi s\theta(b_x - x_g + x_i) \\ &\quad -s\varphi s\theta(b_y - y_g + y_i) \end{aligned} \tag{4.2.5}$$

in which $s(\cdot)$ and $c(\cdot)$ indicate the sine and cosine functions. A final step is to apply the perspective projection to achieve the Cartesian image coordinates $[x_{i,c}, y_{i,c}]$ by

$$\begin{aligned} x_{i,c} &= X_c/Z_c \\ y_{i,c} &= Y_c/Z_c \end{aligned} \tag{4.2.6}$$

Due to the compression of data from the imaging process only 2 data points (x_p, y_p) are available from an image. However, 3 variables, (x_i, y_i, z_i), of the centerline or interest point exist in the world coordinate system and are needed for navigation. Solutions to this problem exist in the literature [15], [16] binocular vision will capture two images simultaneously and compare the pixel difference to solve the third unknown (Z_c) of equation 6. A simpler approach is used in this study following the assumptions required in [21]. Presume the height of the centerline interest point on the bed is constant (20 cm) and the variance of plant heights negligible due to the selected image filtering technique. In conjunction with

this assumption, the pixel locations will be scaled using the following,

$$x_{i,p} = a_1 {x_{i,c}}^2 + b_1 x_{i,c} + c_1$$
$$y_{i,p} = a_2 {y_{i,c}}^2 + b_2 y_{i,c} + c_2 \tag{4.2.7}$$

instead of the traditional constant scalar. All the parameters in Eq. 4.2.7 are calibrated via experiments. The final scaled pixel results were compared to experimental data for verification and to establish a numerical uncertainty bound. For an extreme set of three dimensional coordinates (x_i, y_i, z_i) the accuracy was within 5 pixels for each axis, (x_p, y_p) and subsequently lower for more conservative trials. This value is reasonable given the sources of error possible during the comparison: camera pitch angle, feature points of images estimated by human and finally a fixed focal length of the webcamera. Overall, the pixel accuracy is acceptable for 640x480 resolution images and after rounding will be modeled as 99% (from the 5 pixel error compared to the average of the resolution). The percentage difference as compared with the nominal model will be implemented as numerical uncertainty bounds for the cross-bed controllers sufficing as the Δ terms of Assumption 2 (A2).

4.3 Nonlinear Controllers Derivation and Simulation

4.3.1 Nonlinear Controller for the X-Axis Motion

Concluding an alignment check step after phase 2 (**Fig.** 2.1) and before phase 3.1, it can be assumed the robot will be traveling nearly perpendicular towards the desired row. Applica-

tion of the perspective projection from Eq. 4.2.6 including the scaling Eq. 4.2.7 will produce the x-axis pixel output model

$$y_1 = h = [X_c/Z_c] = [h_1/h_3] = x_{p,1} \tag{4.3.1}$$

The derivative of the output with respect to the position state is then

$$\partial h/\partial x_1 = [(\partial h_1/\partial x_1)h_3 - (\partial h_3/\partial x_1)h_1]/h_3^2 \tag{4.3.2}$$

in which x_1 is the first state in $[\ x \quad y \quad z\ ]^T$. This can be simplified as $[(\partial h/\partial x_1)B][(\partial \hat{h}/\partial x_1)B]^{-1} = \Delta_{fx}$ to model the uncertainty. Here the caret symbol represents a nominal model. Lets assume that $0 \leq \|\Delta_{fx}\| \leq D < 1$ The uncertainty will then simplify to

$$\Delta = \partial h/\partial x_1\Big(\partial \hat{h}/\partial x_1\Big)^{-1} = \frac{(\partial h_1/\partial x)h_3 - (\partial h_3/\partial x)h_1}{(\partial \hat{h}_1/\partial x)h_3 - (\partial h_3/\partial x)\hat{h}_1} \tag{4.3.3}$$

h_1 and h_3 are taken from **Eq.** 4.2.6 and after a long derivation and use of Assumption 7 (A7), we obtain $\partial h_1/\partial x_1 = -f_x$ $\partial h_3/\partial x_1 = \partial \hat{h}_3/\partial x_1 = 0$ and $\partial \hat{h}_1/\partial x_1 = -\widehat{f}_x$ while

$$\begin{aligned} \hat{h}_1 = &[\hat{f}_x c\theta + c_x s\phi s\theta](b_x - x + x_i) \\ &+[\hat{f}_x s\theta - c_x s\phi c\theta](b_y - y + y_i) \\ &+c_x c\phi(b_z + z_i) \end{aligned} \tag{4.3.4}$$

$\hat{h}_3 = h_3$ as the equation does not contain the focal length. Overall the uncertainty simplifies to

$$\begin{aligned} \partial h/\partial x_1\Big(\partial \hat{h}/\partial x_1\Big)^{-1} &= \tfrac{(\partial h_1/\partial x_1)h_3 - (\partial h_3/\partial x_1)h_1}{(\partial \hat{h}_1/\partial x_1)h_3 - (\partial h_3/\partial x_1)\hat{h}_1} \\ &= -f_x h_3/ - \hat{f}_x h_3 = -f_x/ - \hat{f} = f_x/\hat{f}_x \end{aligned} \tag{4.3.5}$$

An equivalent expression for Eq. 4.3.5 is

$$f_x = (1 + \Delta_{fx})\hat{f}_x \tag{4.3.6}$$

In order to create a tracking based control scheme, a reference error is formed as

$$e = e_1 = x_{p,1} - x_{p,1}^d \tag{4.3.7}$$

The desired pixel value for this controller will be the center of the image on the x axis, and can be modified based on the image resolution. The camera is fixed to the center of the robot frame (**Fig.** 4.2). Therefore, once the controller has converged the center of the vehicle will be very close to that of the strawberry row.

The following sliding surface is used

$$s = e_1 + \lambda \int_{t_0}^{t} e_1 dt \tag{4.3.8}$$

Following [68], the SISO sliding mode control for this step is

$$u = (L_B \hat{h})^{-1} [\dot{y}_1^d - \lambda y_1 + \lambda y_1^d - k sign(s)] \tag{4.3.9}$$

in which $sign$ is the signum function, and $\dot{y}_1^d = 0$. To guarantee the asymptotic stability, based on [68] and inclusion of the assumed bound D for uncertainty, the control gain k should be calculated by

$$k = [D \left| s(x_{p,1}^d - \lambda x_{p,1} + \lambda x_{p,1}^d) \right| + \eta s]/(1 - D) \tag{4.3.10}$$

The stability proof for the robust controller will be shown for the multi input, multi output vector form in the following **Section** 4.3.2.

Remark 2. It is necessary for this control section to converge to within several cm of the center of the strawberry bed to bound the state uncertainty for the subsequent multi

input multi output nonlinear controller in which the x initial offset states will assumed to be minimized. This is essential due to the coupling effect that occurs between the vehicle states and camera output through the camera matrix and projection, e.g. x offset and heading error drastically influence x pixel value, therefore it is assumed the x offset to be small at the beginning of the next control phase 3.2 of **Fig.** 2.1.

4.3.2 Nonlinear Controller for the Y-Axis Motion and Heading

For the next step of phase 3.2, controlling the translational distance in y and the heading angle θ are simultaneously considered by using two pixels from the image, x-axis and y-axis. The output model in this step is

$$\mathbf{h}^T = \begin{bmatrix} X_c/Z_c & Y_c/Z_c \end{bmatrix}^T = \begin{bmatrix} h_1/h_3 & h_2/h_3 \end{bmatrix}^T = \begin{bmatrix} y_1 & y_2 \end{bmatrix}^T \tag{4.3.11}$$

The derivatives of the outputs with respect to the states are found as

$$\begin{aligned} &\partial h_i/\partial x_j = [(\partial h_i/\partial x_j)h_3 - (\partial h_3/\partial x_j)h_i]/h_3^2 \\ &i = 1, 2; \;\; j = 1, 2, 3 \end{aligned} \tag{4.3.12}$$

Here $i = 1, 2$ is the number of output pixels, $j = 1, 2, 3$ is the number of state vectors $\mathbf{x}^T = [x, y, \theta]$. The resultant combined derivative matrix becomes

$$\partial \mathbf{h}/\partial \mathbf{x} = \begin{bmatrix} \partial y_1/\partial x_1 & \partial y_1/\partial x_2 & \partial y_1/\partial x_3 \\ \partial y_2/\partial x_1 & \partial y_2/\partial x_2 & \partial y_2/\partial x_3 \end{bmatrix} \tag{4.3.13}$$

$$(L_B h) = \left(\frac{\partial h}{\partial x} B\right) = \begin{bmatrix} \partial y_1/\partial x_1 & \partial y_1/\partial x_2 & \partial y_1/\partial x_3 \\ \partial h_2/\partial x_1 & \partial h_2/\partial x_2 & \partial h_2/\partial x_3 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \tag{4.3.14}$$

$$= \begin{bmatrix} \cos\theta \cdot \partial y_1/\partial x_1 + \sin\theta \cdot \partial y_1/\partial x_2 & \partial y_1/\partial x_3 \\ \cos\theta \cdot \partial y_2/\partial x_1 + \sin\theta \cdot \partial y_2/\partial x_2 & \partial y_2/\partial x_3 \end{bmatrix}$$

Via a similar derivation as in Eq. 4.3.5, the uncertainty bound is

$$\Delta = (\partial \mathbf{h}/\partial \mathbf{x}) \left(\partial \hat{\mathbf{h}}/\partial \mathbf{x}\right)^{-1} = \begin{bmatrix} f_x/\hat{f}_x & 0 \\ 0 & f_y/\hat{f}_y \end{bmatrix} \tag{4.3.15}$$

Once again the uncertainty is assumed as bounded and the following expression becomes relevant

$$0 \leq |\Delta_{ii}| \leq D_{ii} < 1, i = 1, 2 \tag{4.3.16}$$

For generation of the control term, it is first necessary to define an error, that will drive the projected pixels of the target bed centerline point $x_{p,1}$ and $y_{p,1}$ to the desired pixel locations $x^d_{p,1}$ and $y^d_{p,1}$. Thus the error term is

$$\boldsymbol{e} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} x_{p,1} - x^d_{p,1} \\ y_{p,1} - y^d_{p,1} \end{bmatrix} = \boldsymbol{y} - \boldsymbol{y}_d \tag{4.3.17}$$

Here $\boldsymbol{y} = [y_1, y_2]^T$ and $\boldsymbol{y}_d = \boldsymbol{h}_d = [y_{1,d}, y_{2,d}]^T$. An example of multi-input multi-output control using integral control for non-holonomic robots is described in [69]. We define the sliding surface as

$$\boldsymbol{s} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} e_1 + \lambda_1 \int_{t_0}^{t} e_1 dt \\ e_2 + \lambda_2 \int_{t_0}^{t} e_2 dt \end{bmatrix} \tag{4.3.18}$$

, in which $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1, & \lambda_2 \end{bmatrix}^T$, and the nonlinear robust controller is designed to be

$$\boldsymbol{u} = (L_B\hat{\boldsymbol{h}})^{-1}[\dot{\boldsymbol{y}}_d - \boldsymbol{\lambda}\cdot\boldsymbol{e} - \boldsymbol{k}\cdot sign(\boldsymbol{s})] \tag{4.3.19}$$

Here $\boldsymbol{k} = [k_1, k_2]^T$ and $\dot{\boldsymbol{h}}_d = 0$, where *sign* is the signum function and produces the characteristic switching inherent to sliding mode control. The "$\cdot$" is the point-wise multiplier.

Proof: The Lyapunov candidate function is selected to be

$$V = \frac{1}{2}\boldsymbol{s}^T\boldsymbol{s} \geq 0 \tag{4.3.20}$$

Ensuring the closed-loop system will have asymptotic stability is accomplished by proving the derivative of the Lyapunov function is negative definite. Consider the index term $i = 1, 2$ for brevity

$$\dot{V}_i = \dot{s}_i s_i \leq -\eta_i \left|s_i\right|, \eta_i \geq 0, i = 1, 2 \tag{4.3.21}$$

Following the method in [68], the objective is stabilization of the model around the sliding surface as $\dot{s} = 0$

$$\dot{\boldsymbol{s}} = (\dot{\boldsymbol{y}} - \dot{\boldsymbol{y}}_d) + \boldsymbol{\lambda}\cdot(\boldsymbol{y} - \boldsymbol{y}_d) = 0 \tag{4.3.22}$$

The control term appears from only a single differentiation of the sliding equation as

$$\dot{\boldsymbol{y}} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_g}\dot{\boldsymbol{x}}_g = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_g}B\boldsymbol{u} = (L_B\boldsymbol{h})\boldsymbol{u} \tag{4.3.23}$$

By using Eq. 4.3.15 and Eq. 4.3.16 to formulate $(L_B\boldsymbol{h})(L_B\hat{\boldsymbol{h}})^{-1} = (I + \Delta)$ an expression for the sliding surface is simplified as

$$\begin{aligned}\dot{\boldsymbol{s}} &= (L_B\boldsymbol{h})(L_B\hat{\mathbf{h}})^{-1}[\dot{\boldsymbol{y}}_d - \boldsymbol{\lambda}\cdot\boldsymbol{y} + \boldsymbol{\lambda}\cdot\boldsymbol{y}_d - \boldsymbol{k}\cdot sign(\boldsymbol{s})] \\ &- \dot{\boldsymbol{y}}_d + \boldsymbol{\lambda}\cdot(\boldsymbol{y} - \boldsymbol{y}_d) \\ &= \Delta[\dot{\boldsymbol{y}}_d - \boldsymbol{\lambda}\cdot\boldsymbol{y} + \boldsymbol{\lambda}\cdot\boldsymbol{y}_d] - \Delta\boldsymbol{k}\cdot sign(\boldsymbol{s}) - \boldsymbol{k}\cdot sign(\boldsymbol{s})\end{aligned} \tag{4.3.24}$$

Let's define $\Lambda_i \triangleq \dot{y}_{d,i} - \lambda_i y_i + \lambda_i y_{d,i}$, therefore, the scalar form of Eq. (4.3.24) is

$$\dot{s}_i = \Delta_{ii}\Lambda_i - \Delta_{ii}k_i sign(s_i) - k_i sign(s_i) \tag{4.3.25}$$

Remembering the modeled uncertainty bound Eq. 4.3.16 and substituting into the differentiated candidate function with the bounding parameter η produces

$$\begin{aligned}\dot{s}_i s_i &= \left[\Delta_{ii}\Lambda_i - \Delta_{ii}k_i sign(s_i) - k_i sign(s_i)\right] s_i \\ &\leq D_{ii}\left|s_i\Lambda_i\right| + D_{ii}k_i - k_i = -\eta_i s_i\end{aligned} \tag{4.3.26}$$

It is now trivial to rearrange items and solve for the variable gain k_i, $i = 1, 2$

$$k_i = \frac{D_{ii}\left|s_i\Lambda_i\right| + \eta_i s_i}{1 - D_{ii}} \tag{4.3.27}$$

which will be applied into the control law. Also as shown in [68], since $D_{ii} \leq 1$, there is always a solution to **Eq.** 4.3.27. Considering the assortment of plausible initial conditions, it is important to note $\lambda_i > 0$ and $\eta_i > 0$ are tunable parameters that will influence the exponential convergence rate as well as amount of time for the errors to approach zero. For the case when $s_i = 0$, once again, we can follow [68] to show it is a trivial case, thus the closed-loop system is asymptotically stable.

Remark 3. The sign function in the controller Eq. 4.3.25 may cause high frequency chattering. This can be easily addressed by replacing the sign function with the boundary layer function [68]. However the driving system in this study only works at a low frequency (about 2 Hz), and the chattering phenomenon is not obvious as shown in the later experiments.

Remark 4. The desired convergence is to within several centimeters y off set and a couple degrees for heading for this multi input multi output controller. These parameters are more relaxed compared to *Remark 2*, because of the coupling effect and an additional simple feedback loop will be utilized to drive the vehicle closer until it is over the row concluding the cross-bed control.

This additional step mentioned in *Remark 4* after the multi pixel input and multi control output is necessary. This is because the camera pitch orientation is fixed and the detection algorithm will continually find a recurring y pixel location for the centroid once the row runs from the top of the image to the bottom, ie. the robot is very close or over the row. From the design of this controller, as the robot approaches the row the translational gain decreases exponentially with the error to zero as desired. Therefore a simplistic proportional vision based feedback loop will be utilized to drive the robot onto the strawberry row and subsequently the robot will switch to the Over-bed navigation.

4.4 Simulation

Before the controllers can be verified in experimental settings it is imperative to model the vehicle, camera and control equations to perform numerical simulations. The control equations are programmed directly from the derivations and grouped into the control block. A camera model block incorporates all the calibration parameters as well as the ability to reconstruct pixel coordinates from three-dimensional Cartesian coordinates as explained in **Section** 4.2. The third block of vehicle kinematics will quantify the vehicle motion in the world coordinate frame as a function of the control commands and current heading angle. **Fig.** 4.3 is a diagram highlighting the connectivity and important parameters for the simulation that are taken from **Chapter** 4 as well as the control derivations of **Section** 4.3.1 and 4.3.2. The first simulation was performed using the single input single output controller

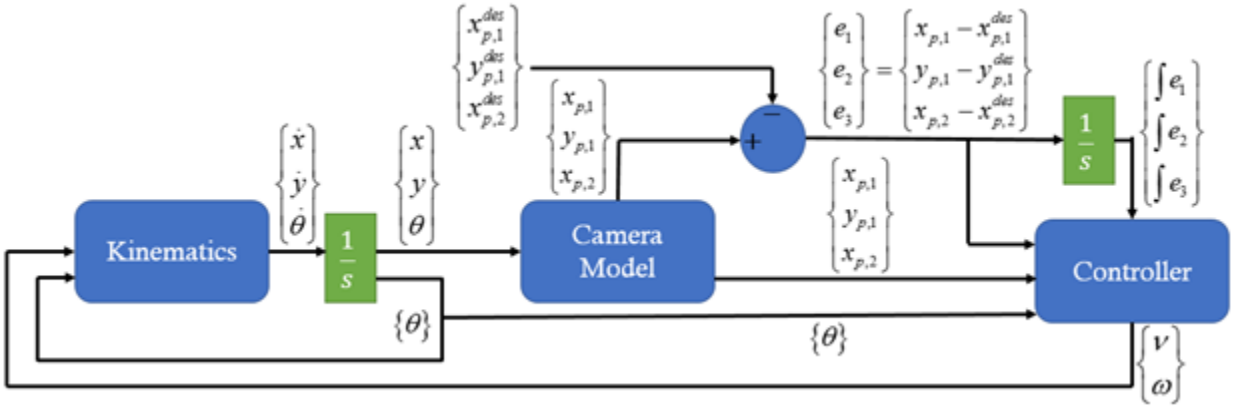
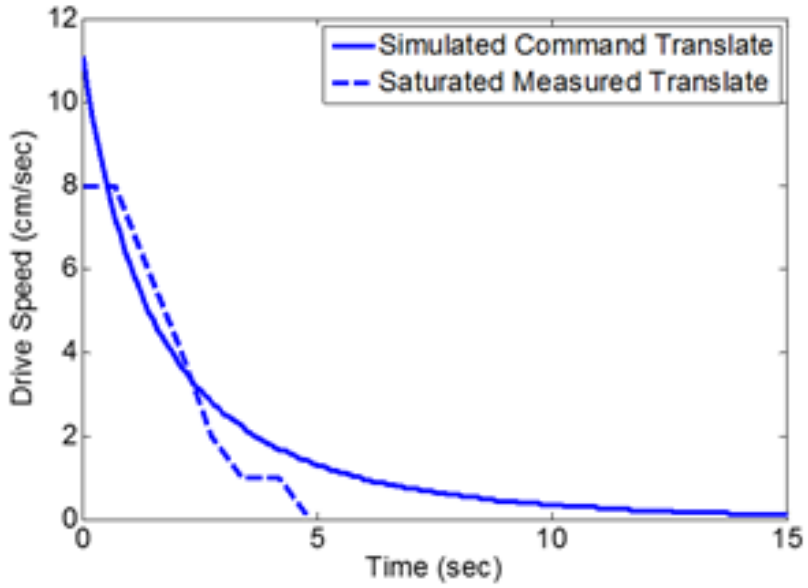


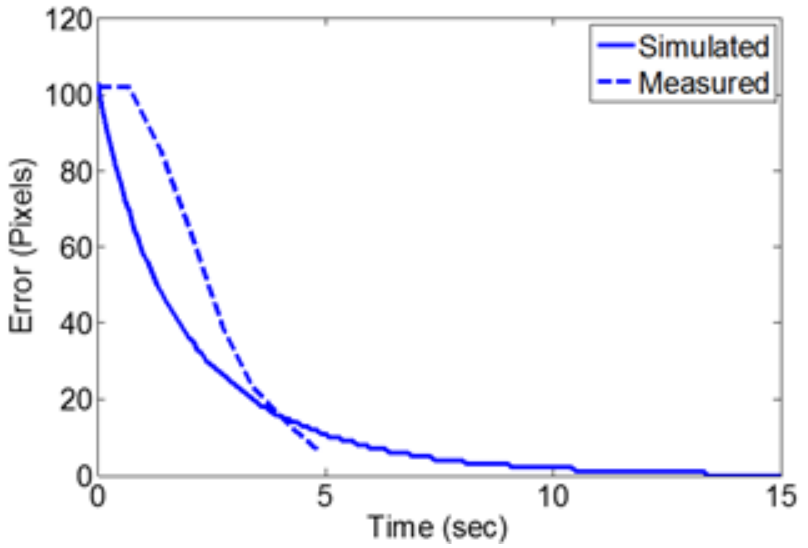
Figure 4.3: Nonlinear controller with vision feedback block diagram

in which a single pixel is taken from the image of a sample strawberry row and centerline, then

generation of the control command is done to reduce the offset error. Figure 4.4(a) shows both the numerically simulated response using Simulink® software and the conventional ODE solver as well the measured response for the laboratory simulated experiment. The initial condition was set to 25 cm x offset from the desired location in the program and an estimated 25 cm for the lab setup. Figure 4.4(a) depicts the exponential decrease of the control gain and simultaneously asymptotic reduction of the error **Fig.** 4.4(b) to a final value of approximately 8 pixels at about 5 seconds. This result was desirable as the simulation duration was limited to 15 seconds with an exit command for the controller once the error reached the tolerance set to 9 pixels. Simulation of the multi input multi output nonlinear sliding mode controller was accomplished in a similar matter. Two pixels from the emulated centerline image were fed into the program as initial conditions corresponding to approximately 8 degrees heading offset and 40 cm translational y offset. Features of interest for subplot of **Fig.** 4.5(a). include, exponential convergence of the control commands for both translate and rotate, similar to that of **Fig.** 4.4(a) only differing in magnitudes. An artificial saturation was placed upon the rotation channel so that the robot did not move too quickly while turning. It is also important to note that the vehicle motion was simulated by moving the cameras manually, this provides explanation as to the discrepancy between the simulated rotate curve and measured saturated rotate curve. The important features of **Fig.** 4.5(b) are the reduction of each error curve to within the preliminary defined tolerance of 10 pixels magnitude for both x and y directions. Final error for the measured x pixel was 10, while the y pixel was -5 at the end of the control duration at approximately 7



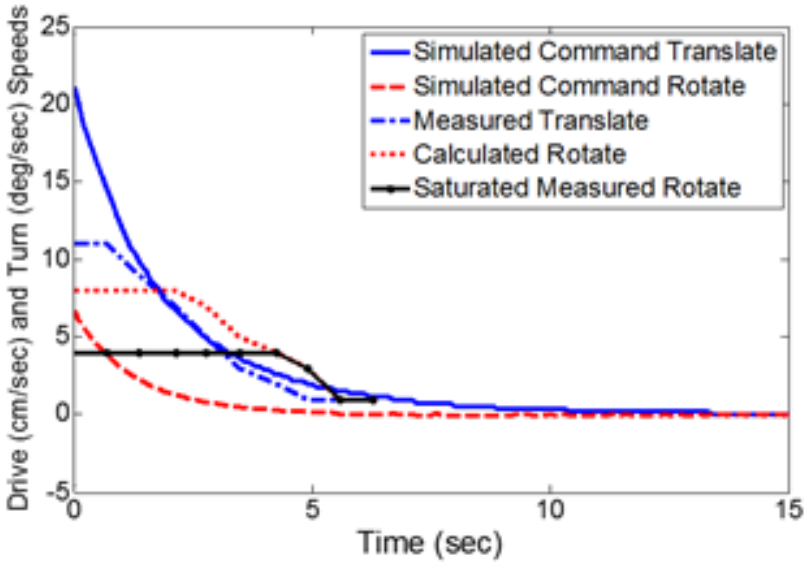
(a)



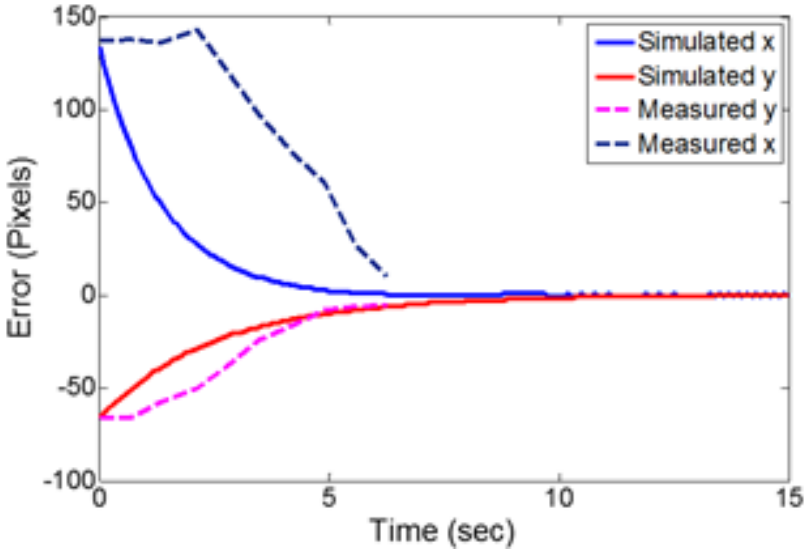
(b)

Figure 4.4: Single input single output nonlinear sliding mode control simulation (**a**) control (**b**) error

seconds. The numerical simulation continues for the full duration of 15 seconds, in which the chattering phenomena due to the inherent characteristics of the sliding controller can be seen (Simulated x curve, between 10-15 seconds of **Fig.** 4.5(b)).



(a)



(b)

Figure 4.5: Multi input multi output nonlinear sliding mode control simulation (**a**) control (**b**) error

CHAPTER 5

EXPERIMENTAL VALIDATION

5.1 Over-Bed PID Algorithm

The proposed over-bed path control method for DDAGR is first tested in Pappys Patch, a small commercial farm with wide travel paths as shown in **Fig.** 5.1. The control algorithm parameters and DDAGR width settings are listed in **Table** 5.1. Two experiments are conducted: 30-meter forward motion with a small initial condition (IC) misalignment and 70-meter reverse motion with a large IC misalignment. As shown in **Table** 5.1, in the first test, initially DDAGR has a 0.22 cm bias in the y direction and a 0.52° bias in heading. Since two tests are conducted in different rows, the bed width is slightly different.

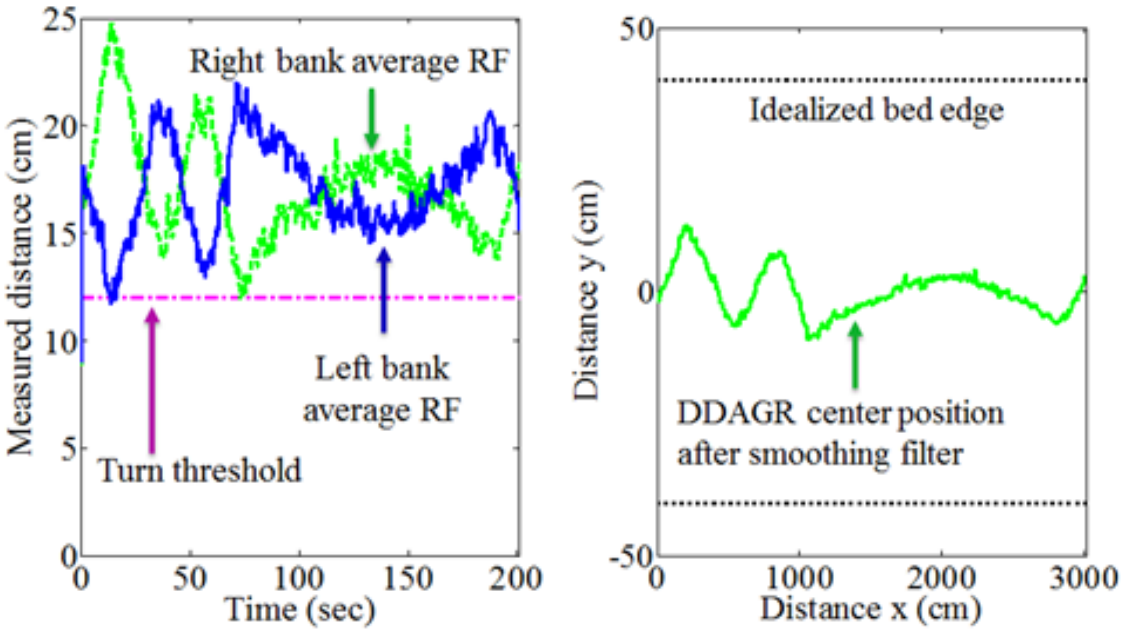
Fig. 5.2(a) shows the average measured RF distance for both sides of DDAGR. It is reasonable that the average readings for the left and right bank are out of phase which is apparent in the plot. The turn threshold indicates where the control for the turn channel was fixed to the predefined limit to ensure the vehicle did not approach much closer to the bed wall. The DDAGR center position along the path compared to idealized bed edges is depicted in **Fig.** 5.2(b), note the scale difference between x and y axes to enhance the path characteristics. In the real field the bed edges are not uniform nor perfectly straight. This

Table 5.1: Small Commercial Farm Experiment Settings

Parameters/settings	First test: 30 meters forward	Second test: 70 meters reverse
Speed limit ω (drive)	15 cm/sec	-15 cm/sec
Speed limit ω (turn)	2.25 °/sec	2.25 °/sec
I.C. e_y	0.22 cm	-1.11 cm
I.C. e_θ	0.52°	5.4°
ϵ_x	2 cm	2 cm
δ_y	4 cm	4 cm
ϵ_y	12 cm	12 cm
σ_y,σ_θ	5 cm, 1.5°	5 cm, 1.5°
k_p	0.003	0.003
k_i	0	0
k_d	0.03	0.03
w_θ	0.6	0.6
w_y	0.6	0.6
Vehicle internal width	107.0 cm	105.0 cm
Bed width	72.4	70.6
Path width	59.2	58.0

Figure 5.1: Strawberry orchard tests with large travel path widths, DDAGR in its expanded form

center position is calculated by first smooth filtering of the RF readings and then averaging the pair on the left and right banks. It is evident from this figure that the final offset in the y direction, e_y , is -3.15 cm, while the final heading bias, e_θ , is 1.05°. **Fig.** 5.3(a) displays the encoder measured drive speed with a zoomed-in portion shown in **Fig.** 5.3(b). Similarly, **Fig.** 5.3(c) shows the turn speed of the vehicle for this test with a comparable zoomed-in segment presented in **Fig.** 5.3(d). As expected the drive speed remains relatively constant at the speed limit (15 cm/sec). Intuitively, the peaks for the turn channel occur at the same time as the RF readings reaching the turn threshold which is expected (17 and 75 seconds). Although the encoder data trends appear to have a chattering phenomenon, it is important to keep in mind that the overall timescale for the experiment is several minutes. Furthermore, all of the following attribute to the variability of the measurement of the vehicle speeds:



(a) (b)

Figure 5.2: Range Finder navigation data in the 30-meter forward motion: (**a**) Left and right wheel bank average RF reading; (**b**) Calculated vehicle center

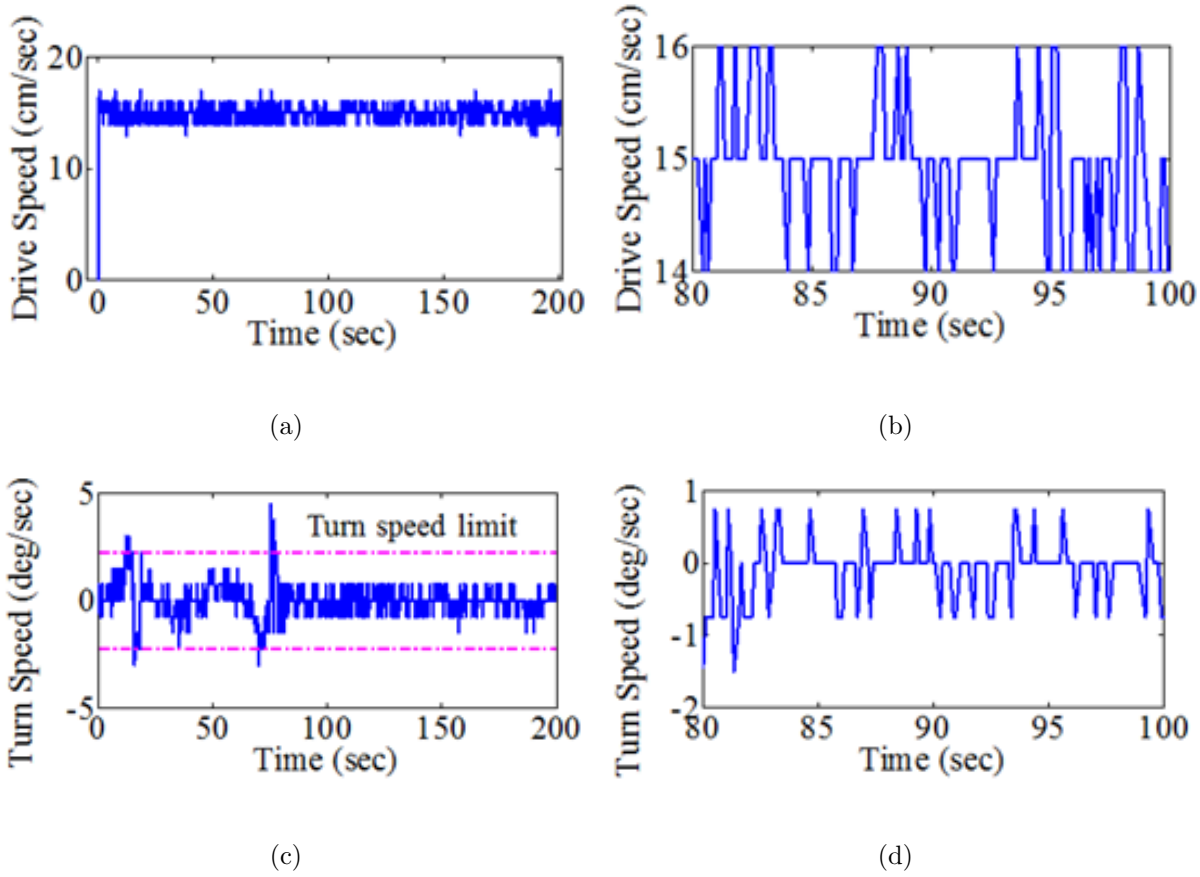


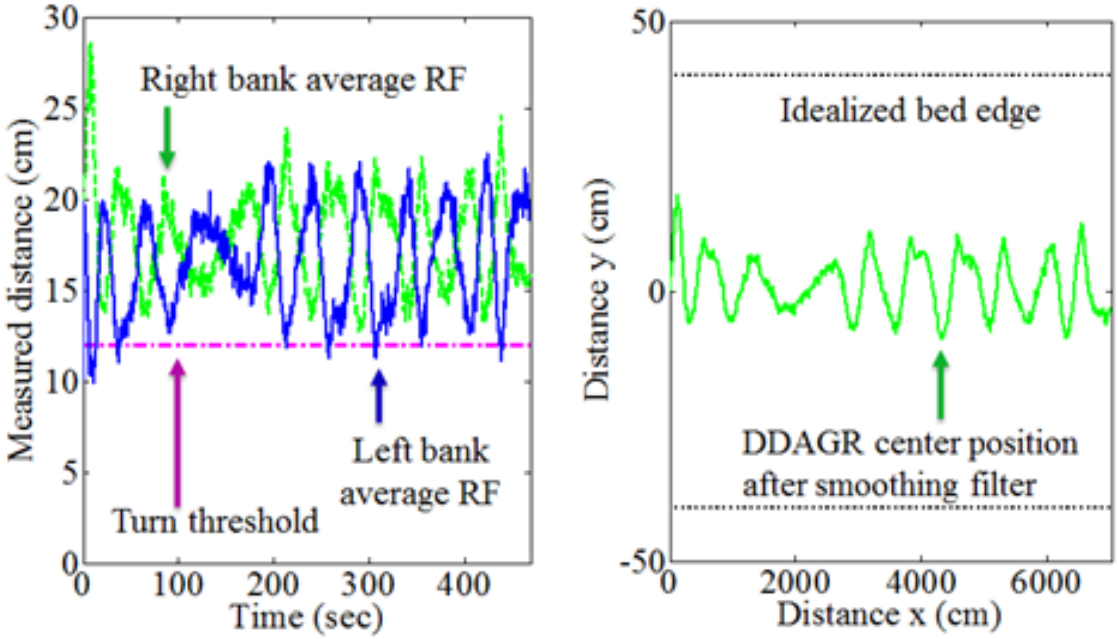
Figure 5.3: Encoder navigation data in the 30-meter forward motion (**a**) Drive speed; (**b**) Zoomed-in drive speed; (**c**) Turn speed; and (**d**) Zoomed-in turn speed

encoder resolution, range finder accuracy, drivetrain imperfections, and soft sand inducing wheel slip, as well as uneven terrain as is expected in the strawberry field. The average loop time including calculation and commands was 0.15 seconds for this test. An average e_y was determined as -0.03 cm, while the calculated average e_θ was 0.11° using Eqs. 3.1.1 and 3.1.3 and the time interval for each control loop.

In the second experiment, a relatively larger initial condition for e_θ was used, and the robot was driven backwards for 70 meters. A significant feature in **Fig.** 5.4(a) is the

ability of DDAGR to compensate for the large biased initial condition, as the very first crossing of the threshold (around 5 seconds) occurs at the beginning of the experiment as is expected and each subsequent approach to the threshold the overshoot is minimized. One other item of interest is the fact that the right bank average does not meet the threshold on this trial, which we attribute to the filtering effect, sensor error, difference in response capability of each drive side, and finally the initial condition offset bias towards the left bank. Nevertheless, the vehicle center path is exhibited in **Fig.** 5.4(b) as stable and non-diverging for the entire 70 meters of travel with the final e_y and e_θ determined to be 2.99 cm and 0.39°. Similarly, as shown in **Fig.** 5.5(a), the drive speed is stable around the average speed of -15 cm/sec, while the turn speed summits of **Fig.** 5.5(c) once again match time wise with the RF readings crossing the threshold as expected. Based on experiment observation, the actual motor speed follows the commanded value, however due to measurement noise there are spikes above the preset turn speed limit as evident in the plot. **Fig.** 5.5(b) and **Fig.** 5.5(d) are zoomed-in selections of the drive speed and turn speed plots respectively. Once again, the average loop time was 0.15 seconds and an average e_y was determined as -1.2 cm, while e_θ was 0.27°. The second field test is used to validate the DDAGR design and its over-bed control method in a commercial field with less room for steering. The field conditions and robot are shown in **Fig.** 5.6 and **Table** 5.2 lists the experiment settings.

A typical travel path for over 21 meters is shown in **Fig.** 5.7. Only Control Region (*ii*) for over-bed navigation, consisting of the PID equations was required in these tests due to the smaller uniform path widths and consistent structured bed walls, therefore the



(a) (b)

Figure 5.4: Range Finder navigation data in the 70-meter reverse motion: (**a**) Left and right wheel bank average RF reading; (**b**) Calculated vehicle center

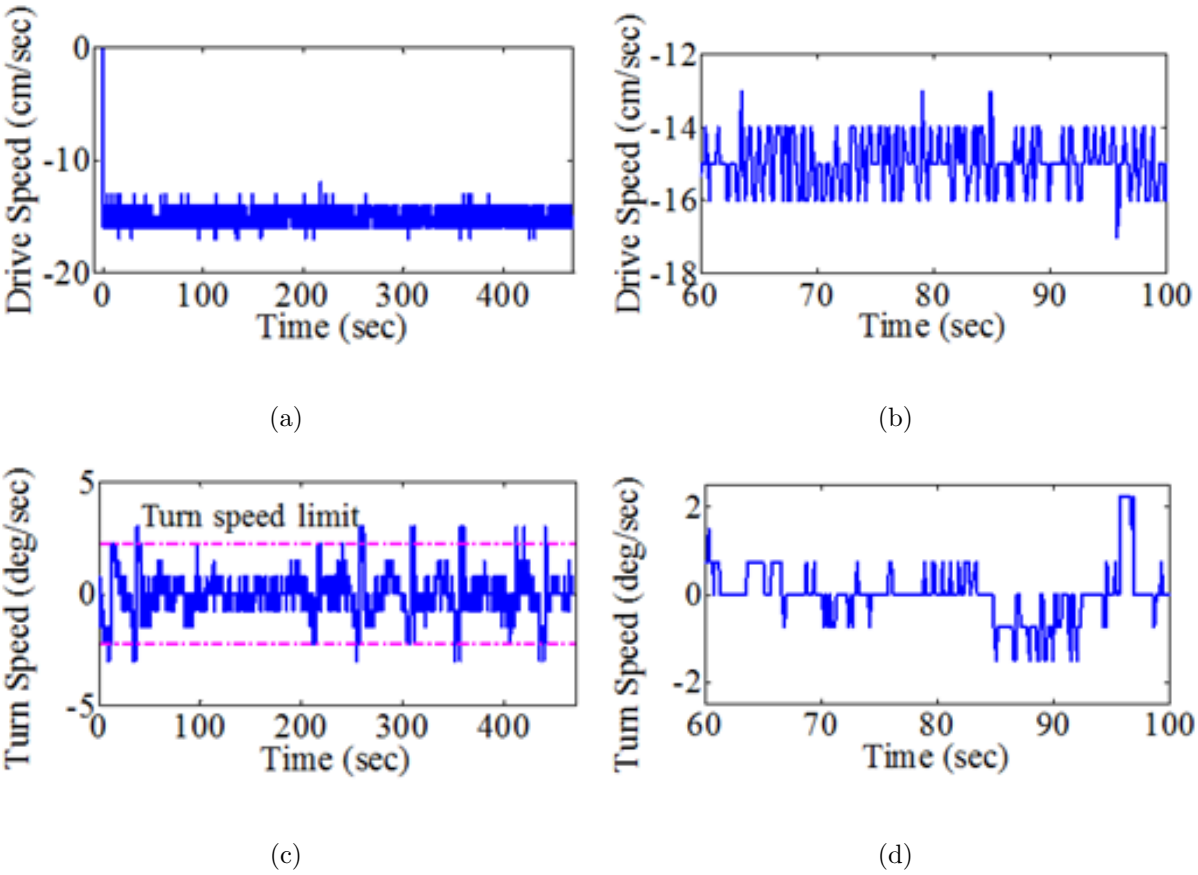


Figure 5.5: Encoder navigation data in the 70-meter reverse motion (**a**) Drive speed; (**b**) Zoomed-in drive speed; (**c**) Turn speed; and (**d**) Zoomed-in turn speed

Table 5.2: Large Commercial Farm Experiment Settings

Parameters/settings	First test: 21 meters traveling fast	Second test: 10 meters with stops
Speed limit ω (drive)	20 cm/sec	10 cm/sec
Speed limit ν (turn)	4.5 °/sec	5.4 °/sec
I.C. e_y	-0.27 cm	0.16 cm
I.C. e_θ	0.64°	0.26°
ϵ_x	1 cm	1 cm
δ_y	1 cm	1 cm
ϵ_y	2 cm	2 cm
σ_y,σ_θ	0 cm, 0°	0 cm, 0°
k_p	1	1
k_i	0.1	0.1
k_d	0	0
w_θ	0	0.5
w_y	1	0.5
Vehicle internal width	80.0 cm	80.0 cm
Bed width	70.7	70.7
Path width	48.7	48.7

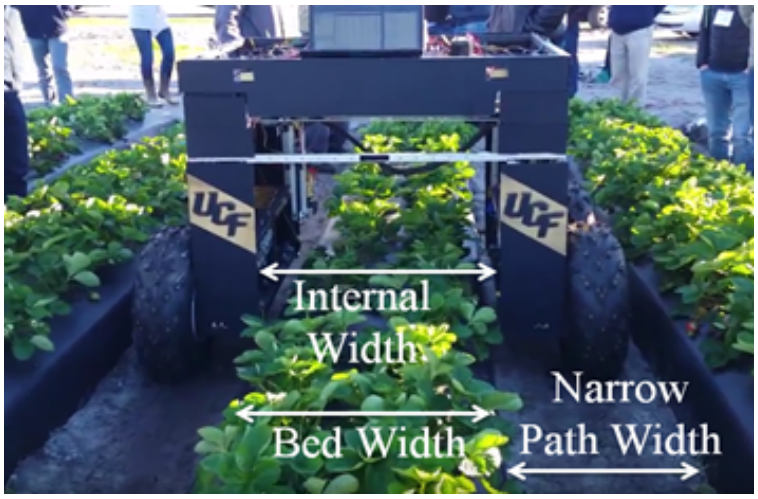


Figure 5.6: Commercial Strawberry orchard tests with small travel path widths, DDAGR in compact form

controller tolerances were set relatively low. The initial conditions for the simulation were established based on an average estimate of the vehicle orientation and offsets with respect to the row center (denoted as the Simulated Path in **Fig.** 5.7). For the experiment, the vehicle was driven to the end of the row at an average translational speed of 20 cm/sec. Using the controller, the largest translational offset of the vehicle was 6.5 cm and the final y error was 0.86 cm. The scaling of the X to Y axis is 6.5:1 to enhance the path trajectory as before. Overall, the integrated IMU data with the drivetrain encoder feedback (Kangaroo/IMU Path) provides a very accurate projection of the stable vehicle path while travelling to the end of the row. Furthermore, in order to accomplish auxiliary tasks such as imaging and sample collection, DDAGR must be able to stop at a desired location while travelling down the strawberry row. The steering algorithm package must be able to continue once

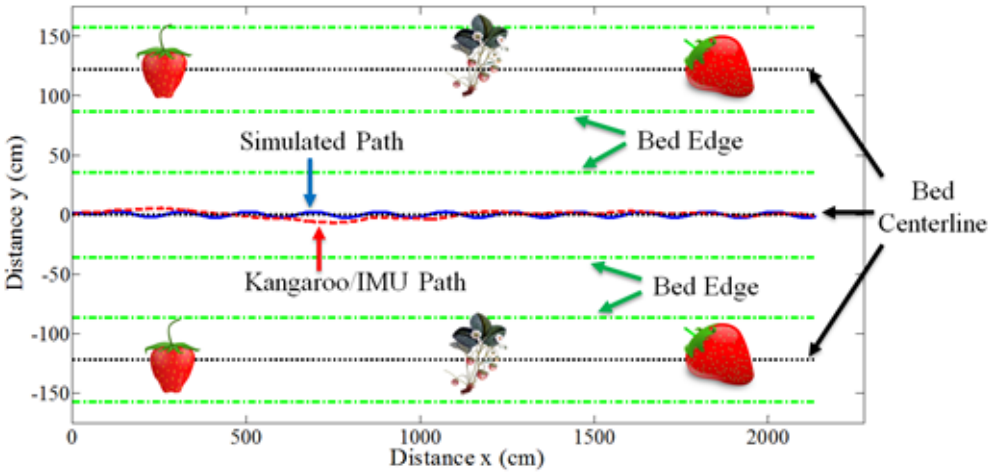


Figure 5.7: PID controller and simulation for 21 meters in a commercial farm, the average drive speed is 20 cm/sec

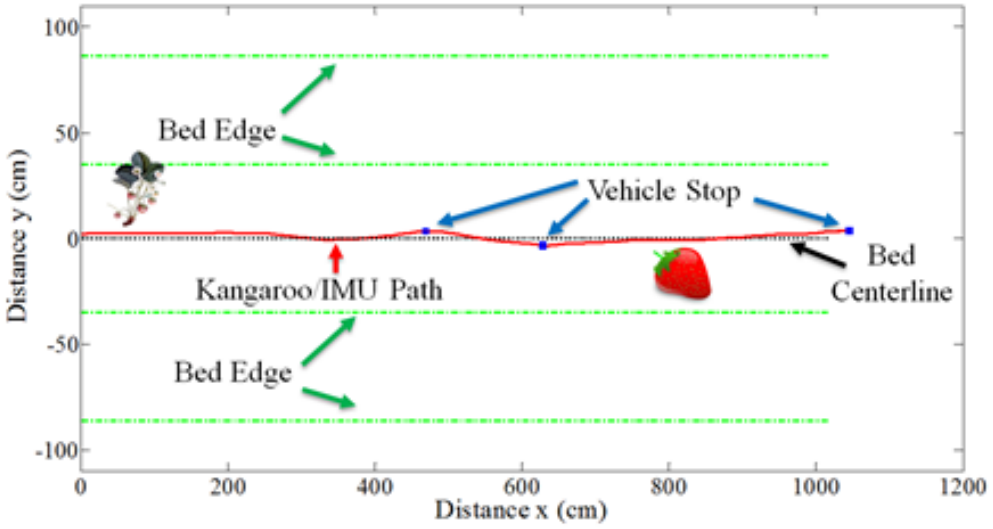


Figure 5.8: PID controller with scheduled stops in a commercial farm, the average speed is 10 cm/sec

the intermediate steps have been performed without sacrificing navigational accuracy. To demonstrate this task, the robot was allotted a 10 meter row portion in which it had to stop and actuate an attached manipulator and XYZ system at 440 cm as well as 600 cm and finally drive to the end of the row as shown in **Fig.** 5.8. The blue squares indicate where DDAGR stopped for actuation of the manipulator system and stopped to reverse and determine the robot heading with respect to the bed at the end of the row at 442, 596, and 1005 cm respectively. This data set proves the robustness of the controller, with respect to the uncertainty that can be expected in a typical commercial strawberry farm. It also proves that the high level logic allows the controller to be interrupted to perform supplementary tasks and return to accurate navigation after the process is complete. Once again the X to Y scaling is 5.5:1 to accentuate the robot offsets; nevertheless the maximum translational y offset for the vehicle was 3.4 cm while over the strawberry bed.

5.2 SISO and MIMO Non-linear Controllers

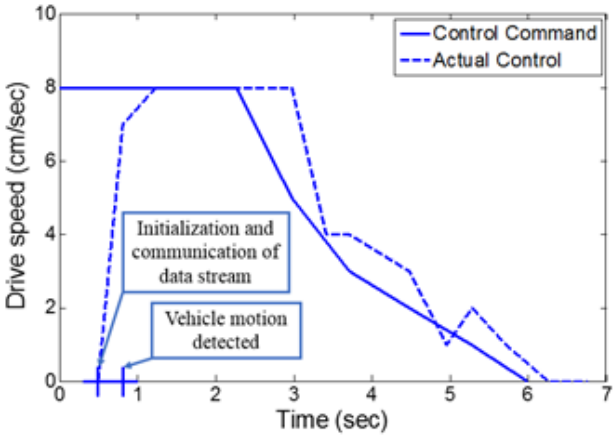
The proposed nonlinear controllers for the cross-bed motion are implemented on DDAGR and tested in a typical commercial strawberry field in Florida. **Table** 5.3 summarizes the parameters of interest that are either defined, measured or calculated during the experiments. It is important to note that the SISO controller only contain parameters for the x-axis. The control was implemented in Matlab for ease of integrating the complementary sub-systems that will be accessed during DDAGRs operation, however conversion to C would prove more

Table 5.3: SMC Parameters and Field Experiment Settings

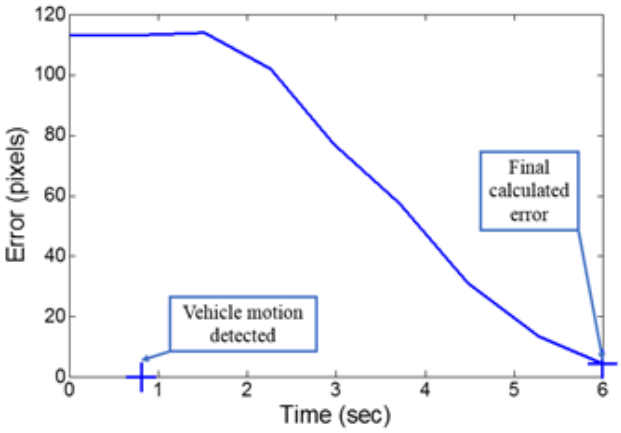
Parameter	SISO	MIMO
λ_y,λ_x	6×10^{-6}	6×10^{-6},6×10^{-6}
η_x,η_y	7×10^{-4}	5×10^{-3},5.5×10^{-4}
${x_{p,1}}^d$,${y_{p,1}}^d$	320 pixels	[320,360] pixels
I.C. (x_0, y_0, θ_0)	$x_0 = 25$ cm	$y_0 = 40$ cm, $\theta_0 = 13^\circ$
Control tol. x_p, y_p	9 pixels	[13,15] pixels
Speed limit ω (drive)	8 cm/sec	N/A
Speed limit ν (turn)	N/A	4 deg/sec
Average update time	0.74 sec	0.78 sec
Final calc. error x_p,y_p	4 pixels	[2,-6] pixels

efficient for just the calculations. A saturation plateau is used to prevent the robot from driving too fast, allowing adequate time for image capture and processing. **Fig.** 5.9(a) depicts the calculated as well as the measured control efforts for the SISO controller in phase 3.1 of the over-bed control. Evident in the plot is the delay between the commanded control and the actual measured control indicated by the first cross mark on the time axis. This is attributed to the time required for initializing the stream of the navigation data. The vehicle motion begins immediately after this step and is identified in the subsequent data update, designated by the second cross mark as vehicle motion detected in the plot. As shown in **Fig.** 5.9(b), the error of the pixel is driven towards zero at the end of this control section, which means DDAGR drives to minimize the distance error in the x direction. Initial and final images captured by the side camera onboard DDAGR are shown in Figure 5.10. In **Fig.** 5.10(a), the initial x-axis pixel of 433 corresponds to about a 25cm displacement between the robot center and the row center (x_0). In the final image (**Fig.** 5.10(b)), the x-axis pixel is 319, which is very close to the desired value of 320, $x^d_{p,1}$ from **Table** 5.3.

In accordance of *Remark* 3, and the anticipated chatter phenomena of the sliding surface, several parameters were noted that would contribute to mitigate the unwanted rapid switching of the control. The first parameter is directly influenced as the control tolerance values set in Table 5.3. In effect, this allows switching off the nonlinear controller once the pixel error was within the set range, these were elected based on the desired accuracy mentioned by *Remarks* 2 and 4 in **Sections** 4.3.1 and 4.3.2. The second parameter that will reduce the frequency of chattering is the overall control loop time as recorded and indicated



(a)

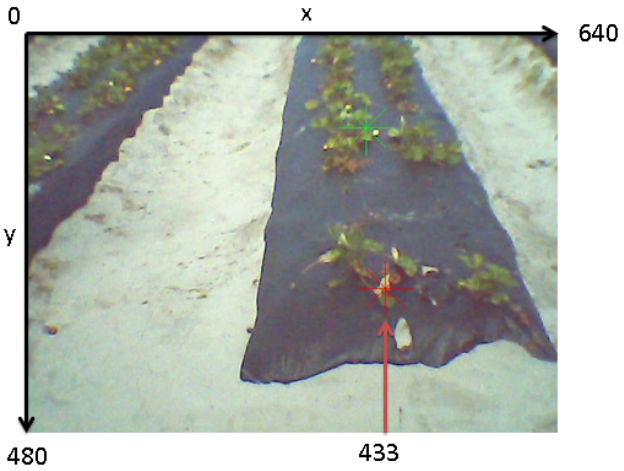


(b)

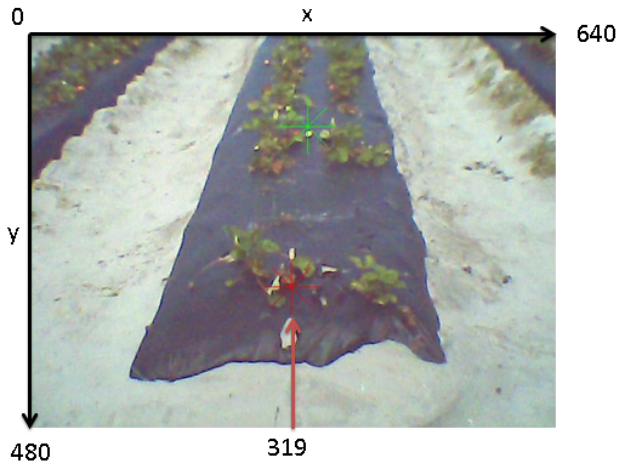
Figure 5.9: SISO controller in Phase 3.1, (a) Drive speed, and (b) Pixel error [54]

by average update time in **Table** 5.3. This value is dependent upon the image acquisition time, filtering, control calculation and actuation and therefore restricts how frequently the commanded control value is changed which is approximately 1.33 Hz for these tests. One final influence of the chatter magnitude was noted in the simulations as the control gain parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\eta}$ as seen in the table these were set conservatively.

In between the SISO and MIMO controllers is a rotation, step 2 of **Fig.** 2.1, in which the vehicle utilizes encoder feedback to rotate to a specific angle. From the preliminary alignment correction the desired rotation angle is 90 degrees. Inherently, operations in real fields produce uncertainties for rotation from measurements, drivetrain symmetry, wheel slip, ect. Therefore the MIMO controller developed in **Section** 4.3.2 is used to correct for either under or overshoot of the robotic system. Simultaneously the robot will drive towards the strawberry row to minimize the required effort of the next control phase, placing the vehicle over the row. Figure 5.11(a) displays the commanded and actual control effort. Once again a saturation is implemented, this time on the rotation channel so that the control loop has adequate time to respond and the vehicle does not jolt. The initialization time and motion detection are once again specified by cross marks. Based on **Fig.** 5.11(b), the final error for the $x$axis pixel is 2, while the final error for the y-axis pixel was -6. Both values are within the tolerances defined in **Table** 5.3. The initial and final images captured by the front camera aboard DDAGR are shown in Figure 5.12. In **Fig.** 5.12(a), the initial x-axis pixel of 476 corresponds to about 13 degrees heading offset e_θ. The initial y-axis pixel of 292 correlates to approximately 40 cm displacement between the robot center and the desired

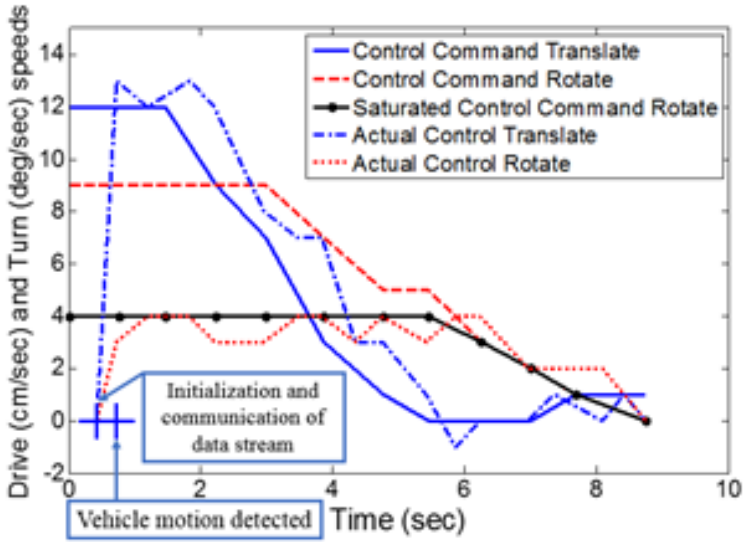


(a)

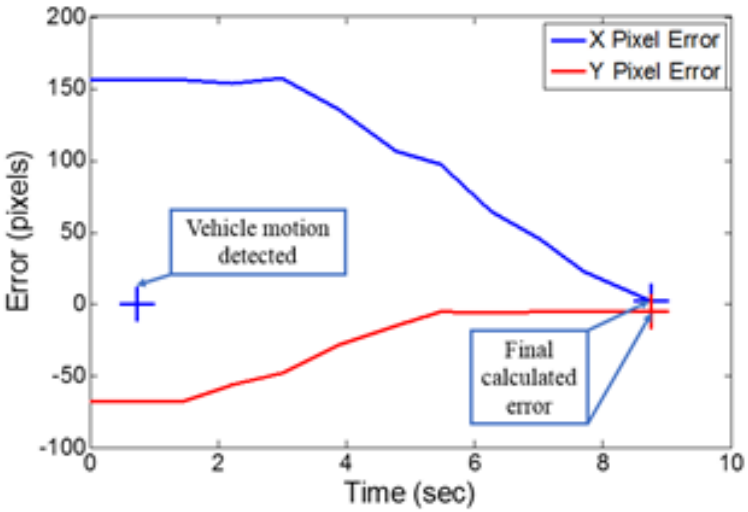


(b)

Figure 5.10: SISO controller in Phase 3.1, (a) Initial image, and (b) Final image



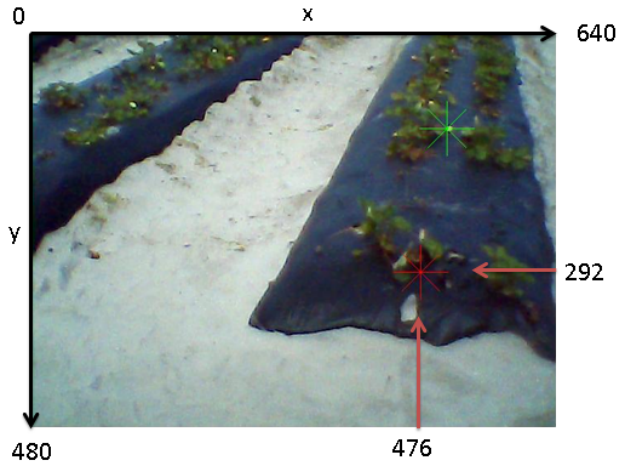
(a)



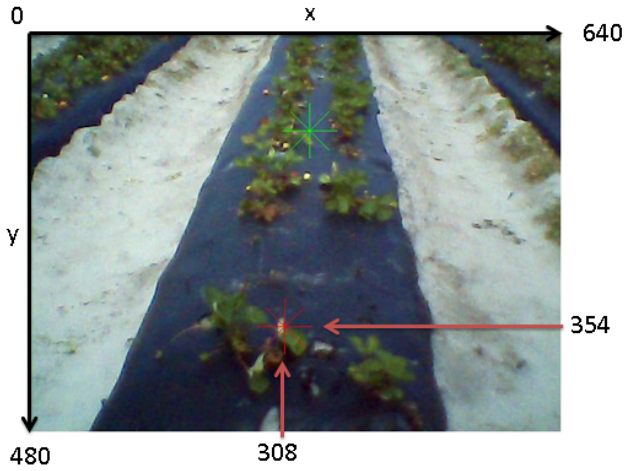
(b)

Figure 5.11: MIMO controller in Phase 3.2: (**a**) Drive and turn speeds, and (**b**) Pixel errors [54]

distance away from the row (y_0). In the final image (**Fig.** 5.12(b)), the x-axis pixel is 308, close to the desired value of 320 while the y-axis pixel is 354 close to the desired value of 360. While these values are within the defined tolerance, it is important to note they were taken after the control had ended and were processed offline. Nevertheless, the final orientation of the vehicle is shown in Fig. 5.13, substantially aligned with the desired row.



(a)



(b)

Figure 5.12: MIMO controller in Phase 3.2, (a) Initial image, and (b) Final image

Figure 5.13: DDAGR orientation after a cross-bed motion, desired row is visible appearing at the center of the vehicle in the picture [54]

CHAPTER 6

REACHABILITY AND COMBINED NAVIGATION

6.1 Reachability Analysis

The content discussed here is adopted from the paper currently under review titled "Vision Based Nonlinear Robust Scouting Control in Agricultural Fields with Reachability Analyses." It is important to verify the stability of the system considering the sources of bounded uncertainties identified earlier by assumptions 1-7, which can be done formally using a method known as reachability analysis. A reachable set is a bounded region of states that are attainable while respecting the restricted group as the system propagates through time [70]. This method of analysis is commonly performed to search for unsafe states in which if the reachable set intersects, the system cannot be deemed safe. For the purpose of our work, proof of the final region of convergence is desired as stated by *Remarks* 1,2 and 4. By considering the reachable set of the system in conjunction with the control as the robot reduces the error, the final state bounds can be approximated and therefore verified to be within the requirements of each respective remark. Development of the formulations for reachability analysis of a differential drive vehicle model will be similar to [44], [71] in which the uncertainty parameters will be modeled and incorporated into the governing equations.

First consider the model for the state evolution as a function of all the indicated uncertainties which will vary slightly from the nominal dynamics model.

$$\dot{x} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(x(t), rf(t), u(t), p(t)) \tag{6.1.1}$$

$x(0) = \begin{bmatrix} x(0) & y(0) & \theta(0) \end{bmatrix}^T \in X(0) \subset \Re^n$ are the initial vehicle states in the world coordinate system as $n = 3$. Error in the sensor based measurements for states will exist in the over-bed section due to noise in range finder accuracy, resolution, as well as disturbed strawberry row walls, $rf(t) \in \Delta RF \subset \Re^n$. The state estimation in the headland of the field is only possible from encoder based measurements or reconstruction of three dimensional coordinates from image data, either of these will have significantly higher accuracy compared to a GPS approach. Since the uncertainty inherent of the camera is modeled and compensated for in the nonlinear controllers, the state reachability analysis will be conducted on the integrated encoder based velocity data. This will avoid inaccuracies from reconstruction of coordinates using pixels but will require incorporation of the the acknowledged bounded wheel slip in the agrarian environment. The control phases to be assessed in the reachability study are depicted in **Fig.** 6.1. It is important to note that the reachability analysis as well as the control capability will be correctly determined for either direction of travel to the next desired strawberry row as will be shown later in the full field tests. The uncertainty parameter for slip will be modeled by $p(t) \in P \subset \Im^m$ where $\Im = [a, b]$ and $[a, b] \in \Re$ are intervals of the uncertainty that will influence the control, modelled around slip estimations

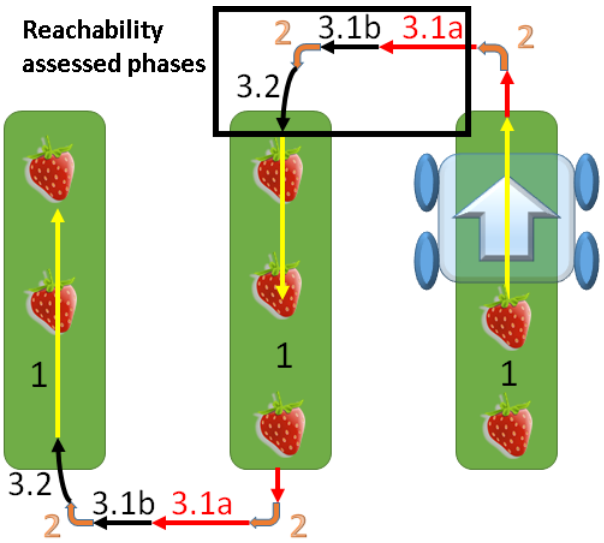


Figure 6.1: Successive control phases used in reachability study (3.1a,3.1b,2,3.2)

from experiments and will vary during each control phase (3.1,2,3.2). Finally the control is $u(t) \in U \subset \Re^m$ where m is restricted to two as the vehicle is only able to translate and rotate. Intuitively, the control and its respective uncertainty must maintain a uniform continuity over each control region also referred to as Lipschitz.

From Assumption 5, the translate speed will be modeled with uncertainty as: $\Delta\nu_{\min} \leq \Delta\nu \leq \Delta\nu_{\max}$ and so the actual translation control becomes

$$\nu = \nu_c + \Delta\nu \tag{6.1.2}$$

Here ν_c indicates the commanded speed which is constant for Phase 1 and calculated from Eq. 4.3.9 for Phase 3.1b. Similarly, the input uncertainty for rotation control (Phase 2) is $\Delta\omega_{\min} \leq \omega_c \leq \Delta\omega_{\max}$, where the actual rotation control is

$$\omega = \omega_c + \Delta\omega \tag{6.1.3}$$

Here ω_c indicates the commanded rotation rate which is constant for Phase 2. The multi-input multi-output control section Phase 3.1b will calculate both control terms ν_c and ω_c from Eq. 4.3.19.

The reachability analysis is to determine the final states based on the closed-loop vehicle dynamics in the form of $\dot{\boldsymbol{x}} = B\boldsymbol{u}$, where $\boldsymbol{u} = [\nu, \omega]^T = u(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\eta})$, and $\boldsymbol{\lambda}$ and $\boldsymbol{\eta}$ are control parameters as shown in Eqs. 4.3.9 and 4.3.19. In addition to the aforementioned control command uncertainties, the feedback $\boldsymbol{x}$ measured by sensors have uncertainties. The uncertainties of the initial condition in the beginning of the cross-bed motion are determined from the conclusion of the over-bed motion. Here we assume

$$\begin{aligned} \Delta x_{min} \leq \Delta x_g(t_0) \leq \Delta x_{max} \\ \Delta y_{min} \leq \Delta y_g(t_0) \leq \Delta y_{max} \end{aligned} \tag{6.1.4}$$

As mentioned in [46] it is important to consider all sources of additive disturbance sequences to evaluate the final controller stability and convergence to within an admissible set. Therefore, the two sources for uncertainty include range finder sensor noise and strawberry bed variance. Use of Minkowski addition in a single dimension for each of the indicated uncertainties results in the expansion of the uncertainty set [72].

The uncertainty in the heading angle measurement is

$$\Delta\theta_{min} \leq \Delta\theta(t_0) \leq \Delta\theta_{max} \tag{6.1.5}$$

at the end of the over-bed phase. The final range of states at the end of each controlled phase are successively applied as the state uncertainty region for the next control phase. As

the range finders can no longer be referenced once in the cross-bed section, the initial states for each subsequent control phase are propagated from the drivetrain motor encoders. It is also important to note that θ will change approximately 90° during Phases 2 and 3.2.

similarly, the Over-bed control of phase 1 is modeled from the summation of range finder uncertainty where $\Delta RF_{row,min} \leq \Delta RF_{row} \leq \Delta\ RF_{row,max}$ and $\Delta RF_{noise,min} \leq \Delta RF_{noise} \leq \Delta\ RF_{noise,max}$ as discussed in Assumption 6.

$$\Delta RF_{row} + \Delta RF_{noise} \leq \Delta RF \tag{6.1.6}$$

Which will make

$$\tilde{e}_\theta = 0.5 \begin{bmatrix} \tan^{-1}\left(\frac{|(RF_{FR}+\Delta RF)-(RF_{RR}+\Delta RF)|}{L}\right) \\ +\tan^{-1}\left(\frac{|(RF_{FL}+\Delta RF-RF_{RL}+\Delta RF|}{L}\right) \end{bmatrix} \tag{6.1.7}$$

similarly,

$$\tilde{d} = \frac{(RF_{FR}+\Delta RF)-(RF_{FL}+\Delta RF)}{2} + \frac{(RF_{RR}+\Delta RF)-(RF_{RL}+\Delta RF)}{2} \tag{6.1.8}$$

and

$$\tilde{e}_y = \tilde{d}\cos\tilde{e}_\theta \tag{6.1.9}$$

The control law then becomes modified to

$$\tilde{e} = w_\theta \tilde{e}_\theta + w_y \tilde{e}_y \tag{6.1.10}$$

and

$$\tilde{\omega} = k_p\tilde{e} + k_i\int_{t_0}^{t}\tilde{e}dt + k_{\tilde{d}}\tilde{e} \tag{6.1.11}$$

The differential equations governing the vehicle motion are then modeled including the identified uncertainties for the cross-bed portion.

$$\begin{cases} x_g = \int_{t_0}^{t_f} (\nu_c + \Delta\nu)\cos(\theta + \Delta\theta)dt + \Delta x_g(t_0) + x_g(t_0) \\ y_g = \int_{t_0}^{t_f} (\nu_c + \Delta\nu)\sin(\theta + \Delta\theta)dt + \Delta y_g(t_0) + y_g(t_0) \\ \theta = \int_{t_0}^{t_f} (\omega_c + \Delta\omega)dt + \Delta\theta(t_0) + \theta(t_0) \end{cases} \tag{6.1.12}$$

Here $\Delta\theta(t_0) + \theta(t_0)$ can be modeled as $\tilde{e}_\theta$ from the last iteration of phase 1 while $\Delta x_g(t_0) + x_g(t_0)$ and $\Delta y_g(t_0) + y_g(t_0)$ can both be ascertained from $\tilde{e}_y$ before and after the 90 degree rotation of phase 2 respectively.

The next step to analyze the reachable configuration of the vehicle comprises a conservative linearization [44]. This linearization step is necessary as propagation of a linear system can be assumed as straight lines for trajectories. Since the dynamic model for the differential drive robot is nonlinear as it is an Affine system, motion in the global frame will be nonlinear. Therefore, the error induced from assuming linear motion will have to be compensated for. This is accomplished by a first order Taylor series expansion of the form:

$$\dot{x}_i \in f_i(z^*) + \frac{\partial f_i(z)}{\partial z}|_{z=z^*}(z - z^*) + L_i(ts) \tag{6.1.13}$$

where the Lagrange Remainder is

$$L_i(ts) = \left\{ \frac{1}{2}(z - z^*)^T + \frac{\partial^2 f_i(z)}{\partial z^2}|_{z=\xi}(z - z^*) | \xi \in X(ts) \times U, z \in X(ts) \times U \right\} \tag{6.1.14}$$

Here $z = \begin{bmatrix} x^T & u^T \end{bmatrix}^T$ and $z^* = \begin{bmatrix} x^* & u^* \end{bmatrix}^T$ are the linearization points of the states and control including uncertainties and ts signifies a timestep for the loop. f_i are the dynamic

equations for the vehicle. After the linearization, the error is to be utilized in formulation of the reachable sets or final states.

Representation of the reachable set is done through formation of zonotopes [73]. This method has the benefit of linear transformations and are formulated by vector addition of Minkowski sums of line segments. The subsequent steps include computation of a convex hull, that will bound the current state as well as the uncertainty, an expansion of this hull formerly referred to as enlargement, from the linearization error, and iteration through time to find the final system states. These computations will be used to verify the autonomous robot will be able to achieve the desired final conditions for each control phase, which will then formulate the initial conditions for the next control phase. Numerical values for the uncertainties were established from numerous field experiment data. The final state expansion for each respective phase can be viewed in Fig. 6.2. Here t_0 indicates the beginning of each control phase while t_f is the final time of that phase. For simplicity each of the initial states for each control phase were reset to 0. However for the final phase, θ_0 was set to a constant, so that the multi-input multi-output actuation would not be a trivial translation and both control terms would be necessary and executed as $\theta = 0$ is what is desired.

It is mentioned in [44] that a constant time step is useful and so the uncertainties were modeled statistically around the elected time step (0.5 s). As the first task, the undisturbed desired vehicle path was simulated to obtain the control gains in Matlab and Simulink. Then, numerical values for the control input uncertainty were established as averages from experimental field data and are listed in **Table** 6.1. For Phase 3.1a, the desired travel distance

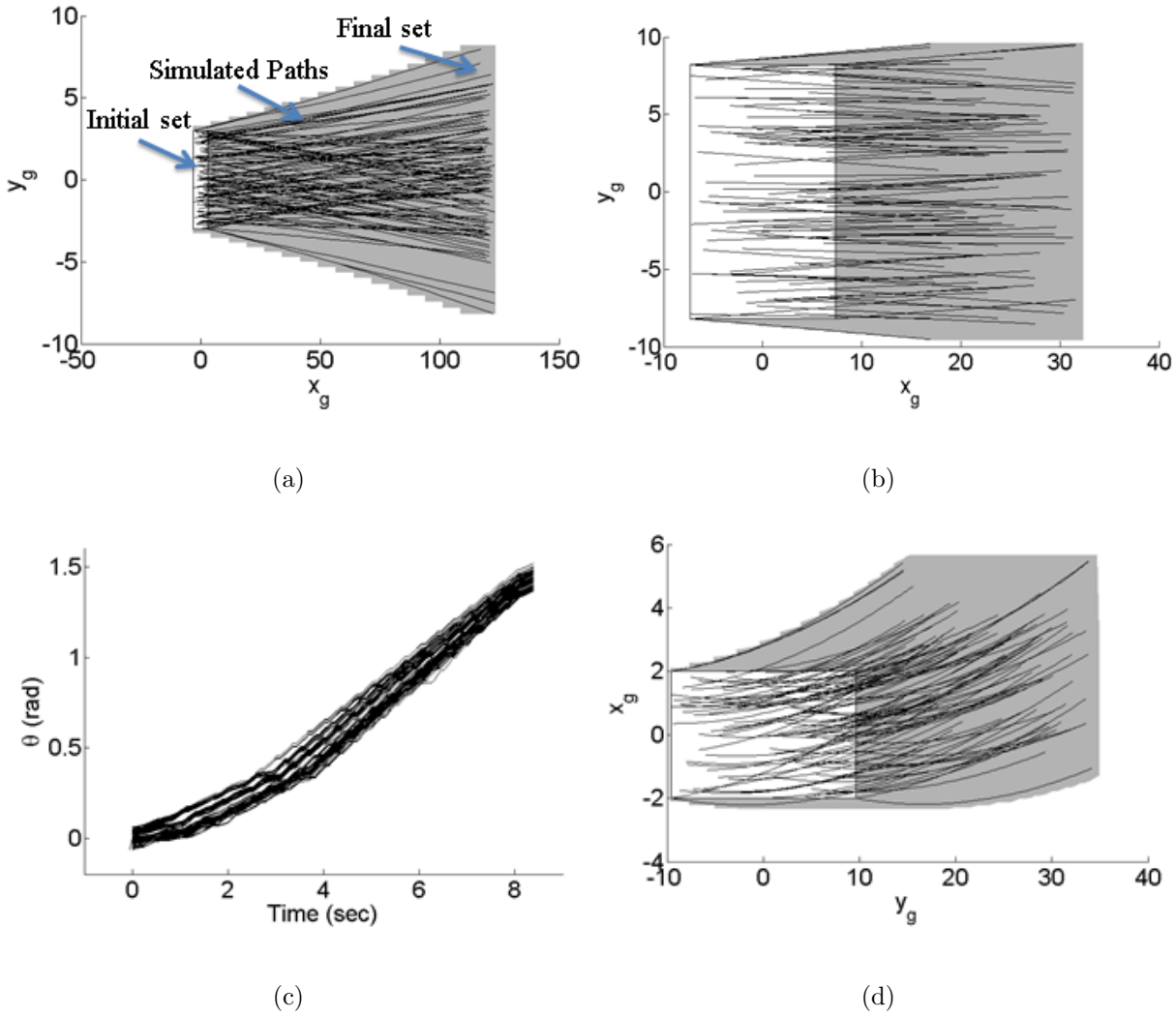


Figure 6.2: Results of Reachability Analysis (**a**) Phase 3.1a; (**b**) Phase 3.1b; (**c**) Phase 2; and (**d**) Phase 3.2

Table 6.1: Uncertainty Bounds for the Reachability Simulation

Bounds	$\Delta\nu$	$\Delta\omega$	$\Delta\theta$	Δx_g	Δy_g
Phase	[cm/s]	[°/s]	[°]	[cm]	[cm]
3.1a	0.12	0.17	2.0	3.0	3.0
3.1b	0.12	0.17	2.9	7.3	8.2
2	0	0.29	3.5	2	9.6
3.2	0.14	0.23	2.9	2	9.6

was $120\,cm$ in x, while the single input controller distance was $25\,cm$ for Phase 3.1b. An 82° (1.4 rad) rotation is required for Phase 2, while Phase 3.2 consists of $25\,cm$ translation in y and 8° rotation for θ from the multi-input control. It is a valid assumption for the simulation that the translational speed of the robot is 0 during Phase 2, likewise the feedback nature of Phase 3.2b will reset the uncertainty bound of x_g to a smaller value, eg. $\Delta x_g = 2\,cm$, as the controller is designed for this purpose. By propagation of the dynamics with the uncertainty the final states and vehicle path are modeled by the simulations using the CORA package in Matlab [72]. As the specific phases selected for analysis are successive (Phase 3.1a - 3.2), the final state set for each simulated region is used as the initial conditions for the subsequent trial, as can be seen in Fig. 6.2. Fig. 6.2(a) depicts the dispersion of the position states in the $120\,cm$ of travel in Phase 3.1a. Figure 6.2(b) shows the position dispersion of the vehicle in Phase 3.1b under the proposed controller Eq. 4.3.9. In Phase 2, an 82° rotation, the

position information with respect to time is shown in Fig. 6.2(c); while the last subplot, Fig. 6.2(d) shows the vehicle position under the controller Eq. 4.3.19., denote the change of axes due to the previous rotation step. The results of the simulation indicate a bounded region of the states for each phase, ideally, the distance states x_g and y_g will be within 2 and 3 cm respectively, once the tolerances and external webcamera feedback have been implemented. These values are well within the capability range of the last translation step, that will place the vehicle onto the desired row. θ will also be minimized from the control and within 1° based on the elected control tolerance.

6.2 Combined Field Test

6.2.1 Experimental Expectations and Circumstances

For the mobile robot to be considered fully autonomous in terms of navigation, it must be capable of transferring from a position in the farm environment either over the strawberry rows or in the headland to other locations. The first tasks conducted were simulation of each of the controllers and tuning of any parameters that will help enhance performance. This step is vital for making sure the robot is operational before taking it out to the field experiments. Slip estimation will be conducted to provide a reference for the initial conditions to be used for the nonlinear controllers. The preliminary results indicated the translational accuracy was up to 98% while the rotational accuracy was around 97% for several samples. These parameters

are important because there are several intermediate steps required between the nonlinear controllers in which the only feedback mechanism available is the encoder measurements. As these values will change depending upon field variances and soil density or composition, it will be important to collect a distribution of the accuracies over several experimentations. It is also important to consider the largest initial conditions that can be handled by the nonlinear controllers. This is from the physical restrictions of the camera such as sensor size and placement including orientation. Therefore, the nonlinear controllers will be tested with significant initial conditions within their respective capable ranges. In conjunction with this, assuming the wheel slip and other disturbances are adequately compensated for there will also be tests with very small initial conditions in which the robot is very well aligned at the beginning of each nonlinear control phase. Intuitively, the objective is to minimize the control efforts, motor usage and time for all initial conditions or states within the control space.

It is anticipated that the nature of the feedback for the non-linear controller will be able to compensate in the existences of overshoot due to disturbance and other characteristics. Overshoot is a common occurrence for systems that contain higher order terms such as velocity, acceleration, inertia and damping. When describing the vehicle model as control affine, meaning the actuation is linear however the system responds nonlinearly with respect to the states, there exists a term formally referred to as drift in which the new state derivative such as velocity is a function of a residual as well as the control and kinematics interaction. This term can be extremely difficult to compensate for and is usually superfluous when de-

veloping the control. As our vehicle will in fact have a significant amount of inertia once in motion this can pose a substantial problem if overlooked. Once the system has been tested and is capable of converging with the prescribed median initial conditions, the additive disturbance will be enacted. The proposal is to induce reasonable bounded uncertainty into the vision feedback loop during control iterations to ascertain the two following concepts: (1) Switching of the control will happen slow in the event of overshoot with low magnitudes, reducing inertial influences. (2) The nonlinear controller will be robust during operation and will continually attempt to minimize error. Logical justification for bounded disturbances include environmental factors such as wind and weed growth that can potentially influence the shape and form of the strawberry bed during the image filtering and detection.

6.2.2 Data Filtering and Algorithm Enhancement

Although the Over-bed control algorithm of **Section** 3.2 worked adequately for the large commercial farm in which the strawberry row structures were well defined and uniform, it was slightly less successful at other test locations. Therefore, enhancements were made to filter the range finder data, before it was utilized in the control calculations. In the events where the PID gains were not sufficient at minimizing the offset errors (e_y and e_θ), or the vehicle approached the row too closely due to other disturbances, the following cases were implemented to prevent collision with the strawberry row. The parameters utilized here are in reference to the the navigation diagram **Fig.** 3.2(b).

Table 6.2: Algorithm for Over-bed Travel with Data Filtering

Step 1	Initialize function: $X = OB(x_d)$
Step 2	While $x_d > x_{tol}$
Step 3	Get range finder data: RF_i, $i = 1:4$
Step 4	Zero Order Hold filter of range finder data: for $i = 1:4$
Step 5	If $abs((RF_i + 1) - RF(i) > \Delta_{tol})$ then $(RF_i + 1 = RF_i)$,$counter++$
Step 6	Else $RF(i) + 1$
Step 7	Average filter range finder data: $RF_(i) + 1 = (RF_(i) + 1 + RF_(i))/2$
Step 8	Calculate e_θ and e_y: Eq. 3.1.1 and Eq. 3.1.3
Step 9	Check for stop conditions: eg. $RF_i = 0$
Step 10	If true, $\nu = 0$ and $\omega = 0$
Step 11	Else check filter counter: $counter_i > val$
Step 12	If true, pause and reset counter: for 0.75 seconds, $\nu = 0$ and $\omega = 0$. $counter_i = 0$
Step 13	Else check $e_y < \sigma_{y_g}$ and $e_\theta < \sigma_\theta$
Step 14	If true, calculate PID gains: Eq. 3.2.1
Step 15	Elseif $abs(e_y) < \sigma_{y_g}$ and $abs(e_\theta) > \sigma_\theta$: Case 1
Step 16	Else $abs(e_y) > \sigma_{y_g}$: Case 2 (stop if pass x_d)
Step 17	Return X

Case 1: If $|\theta_e| > \sigma_\theta$ and $|y_e| \leq \sigma_y$, DDAGR will first stop its translational motion as $\nu = 0$. Here σ_θ and σ_y are tolerance values in θ_e and θ_y. Then it will rotate with a turning rate of $\omega = -sgn(\theta_e)c$ until $\theta_e = 0$. Here c is a constant turning rate, e.g. $c = \omega_{max}/2$.

Case 2: If $|y_e| > \sigma_y$, DDAGR will first stop. Then it will rotate with a turning rate of $\omega = -sgn(y_e)c$ and a constant translational velocity ν until $y_e < 0.5\sigma_y$. After that DDAGR will rotate with a constant turning rate of $\omega = -sgn(\theta_e)c$ and no translational motion of $\nu = 0$, until $\theta_e = 0$.

Table 6.2 shows psuedocode of the updated Over-bed algorithm. The first step is implementation of the Over-bed function in which the desired travel distance x_d is supplied. The next step is a while loop, that will iterated the control calculations and process until the robot has driven to within a small tolerance of x_d down the strawberry row. Inside of the loop, the range finder data is sampled and updated each cycle. A zero order hold filter is first applied to the data to remove erroneous readings. If any current RF value exceeds the previous by a specific amount, the current value is replaced with the previous and a counter is incremented by 1 for that specific range finder. Otherwise, the current value is passed along into an averaging filter. The averaging filter will take the previous measurement and the current measurement and divide by two. It was determined that this window size of two, would provide a reasonable smoothing effect to the data, yet would not cause significant delay of the data accuracy.

Next, e_θ and e_y can be calculated from Eqs. 3.1.1 and 3.1.3. It is then necessary to check for any of the stop conditions that may have occurred, for example, if the range finders

readings are above a certain threshold, for several samples, the robot has passed the end of the bed and needs to stop. If the vehicle has met a stop condition it will cease its motion and wait for instruction, if it has passed the end of the strawberry bed, it will prepare to execute a Cross-bed phase. Otherwise, here it will check the filter counter for each range finder to verify several samples have not been replaced in a row from the zero order hold. If any of the range finder counters (1:4) have exceeded a preset value, the vehicle will pause it's motion for 0.75 seconds before continuing. It was determined during experimentation that this would provide a fluid transition as well as long enough duration for better range finder readings, intuitively, the range finders are more accurate while motionless. Also all of the range finder counters are reset after the pause.

Otherwise, the calculated values for e_θ and e_y will now be used to establish the control gains. First, if $e_\theta < \sigma_\theta$ and $e_y < \sigma_y$ the PID gains will be calculated from Eq. 3.2.1. However, if only the absolute value of y_e is within tolerance, σ_y, and the heading angle is too high, the vehicle will stop and rotate to reduce θ_e. Else, both errors exceed the defined tolerances, the vehicle will stop, then perform the log spiral motion with both translation and rotation to reduce e_θ and e_y simultaneously. Also, during this step if the travel distance exceeds x_d the vehicle will stop. An additional rotation only is subsequently executed here to ensure e_θ is close to zero. Finally, the total travel distance is returned by this function as it will be necessary for other operations.

6.2.3 Two Bed Test Left Turn

The parameters for the full experiment with first an over-bed portion followed by a cross-bed section and finally an over-bed phase are highlighted in **Table** 6.3. The weights in **Eq.** 3.2.2, w_y and w_θ were both set to be 0.00025. It is evident that the single-input cross-bed controller only contains parameters for one degree of freedom in the x-axis. **Fig.** 6.3 depicts the vehicle path from the combined test with all of the individual motions labeled.

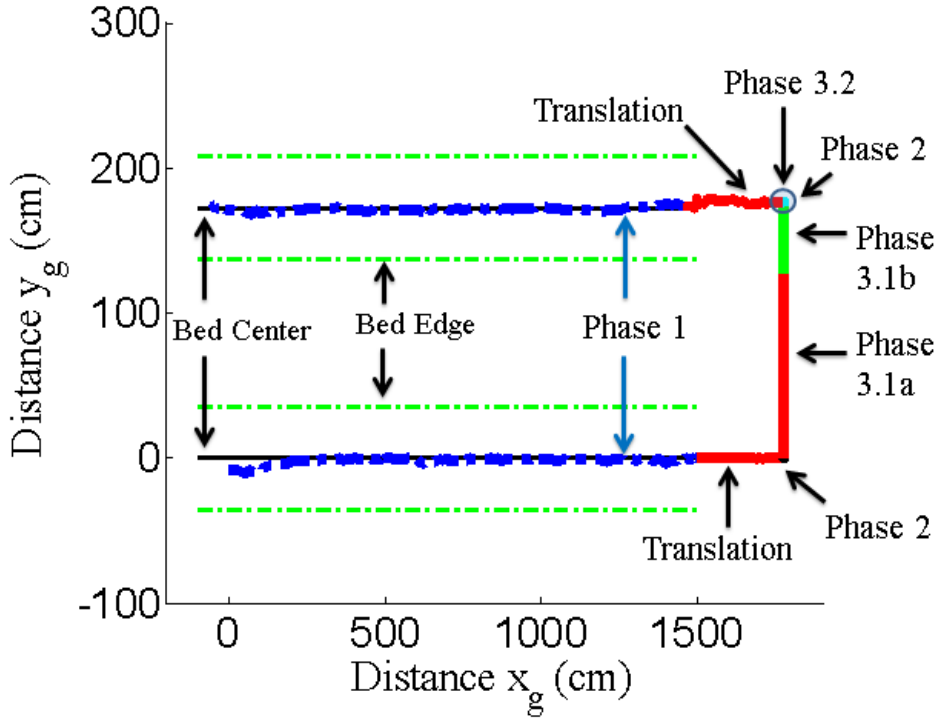


Figure 6.3: DDAGR path for bed transfer, start location (0,0)

It is evident that the vehicle travels safely and closely to the desired path of the center of the strawberry beds, as well as successfully performs the cross-bed motion. The overall

Table 6.3: Experiment Parameters Two Bed Test

Phase/Ctrl. Eq.	Tuned Gains	Initial States
1/Eq. (3.2.1)	$k_p = 466$	$x_0 = 0\,cm$
	$k_i = 20$	$y_0 = -7.1\,cm$
	$k_d = 2$	$\theta_0 = 0.3^\circ$
3.1a/Eq. (4.3.9)	$\lambda_x = 9.5 \times 10^{-4}$	$x_0 = 25\,cm$
	$\eta_x = 1.02 \times 10^{-5}$	$y_0 = 0\,cm$
		$\theta = 0^\circ$
3.2/Eq. (4.3.19)	$\lambda_x = 7.13 \times 10^{-6}$	$x_0 = 0\,cm$
	$\eta_x = 5.0 \times 10^{-3}$	$y_0 = 3\,cm$
	$\lambda_y = 7.13 \times 10^{-6}$	$\theta_0 = 12^\circ$
	$\eta_y = 9.0 \times 10^{-4}$	

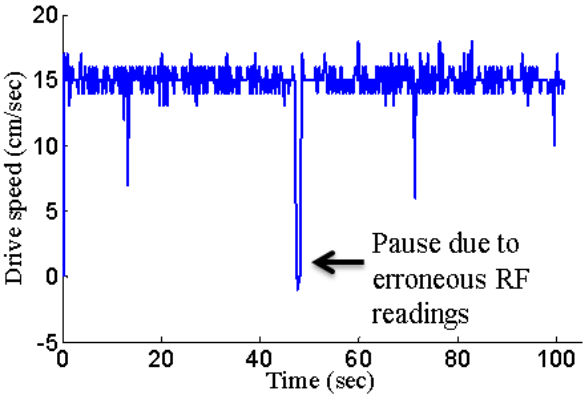
experimental results are shown in **Table** 6.4. **Fig.** 6.4(a) shows the measured drive speed for Phase 1, as well as the robustness of the algorithm in the instance where multiple degraded RF readings are observed and the vehicle briefly pauses at approximately 47 seconds. The vehicle turn speed for Phase 1 can be visualized in **Fig.** 6.4(a). Predominantly, the calculated gain from **Eq.** 3.2.1 and **Eq.** 3.2.2 determine the speed however, disturbances such as measurement noise, wheel slip and motor actuation are also present. The control gains as well as pixel errors are shown for each of the nonlinear controllers (Phase 3.1b and Phase

Table 6.4: Experiment Results Two Bed Test

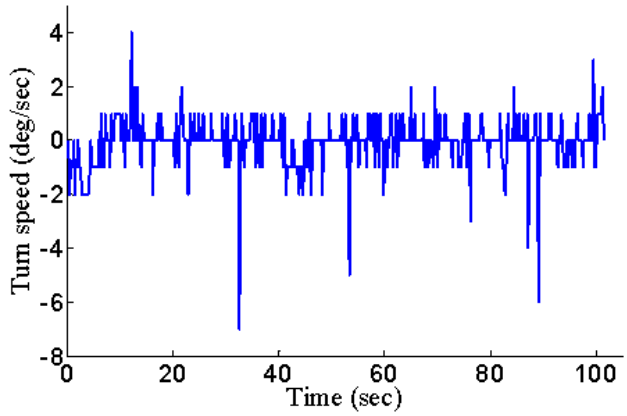
Phase/Ctrl. Eq.	Control Param.	Final Error
1/Eq. (3.2.1)	$\sigma_\theta = 8^\circ$	θ_e =1.8°
	$\sigma_y = 20\,cm$	$y_e = 1.2\,cm$
3.1a/Eq. (4.3.9)	$x^d_{p,1} = 8$ pixels	$x_{p,1} = 7$ pixels
3.2/Eq. (4.3.19)	$x^d_{p,1} = 7$ pixels	$x_{p,1} = 2$ pixels
	$y^d_{p,1} = 8$ pixels	$y_{p,1} = 7$ pixels

3.2), indicating convergence of the controllers in **Fig.** 6.5. Evident in the plots is the exponential decay of the control gain to zero as is expected for the nonlinear controllers **Figs.** 6.5(a),6.5(b). Similarly the pixel error decays to within the desired tolerance within the allotted control period of 20 seconds. It is also apparent that the control gains switch at a relatively low frequency as desired and described in **Section** 6.2.1. A saturation was placed on each of the control speeds to allow adequate time for image acquisition and prevent the vehicle from moving too quickly. The control parameters listed in the table for Phase 1 determine when the robot would enter either of the cases mentioned in **Section** 6.2.2. For the vision based controllers Phase 3.1b and Phase 3.2, the control parameters allow the control loop to continue until the vehicle had converged to within an acceptable distance of the desired location, based on the pixel difference between the detected centroid pixel and the desired pixel.

For Phase 1, an average offset error was determined to be -1.3 cm for y_g, and 0.2° for θ, with the maximum deviation of $10\,cm$, and 3.3°. For Phase 3.1b an initial pixel error was calculated as 87 pixels, which was reduced to within the set tolerance as 7 pixels (**Fig.** 6.5(c)). For Phase 3.2, the initial pixel errors for the second nonlinear controller were -153 and -13 for x and y axes respectively. Definitively, the controller was able to reduce these errors to 2 pixels for the x axis and -7 pixels for the y axis (**Fig.** 6.5(d)). It is apparent that the translation control of the this controller was not very large, due to the small initial error. This was accomplished by adequate performance of all other control phases contingent of the prescribed initial conditions. Similarly, the translation control changes sign during the trial due to the small initial error as well as the coupling between the gain calculations from the derivative matrix as well as the small dependence of y pixel location with respect to the heading angle. A sample of the before and after robot perspective can be seen for the single input controller in **Fig.** 6.6 also included is the filtered image. Figure 6.7 shows the before and after view from the robot for the multi input multi output controller.



(a)



(b)

Figure 6.4: Phase 1, Two Bed Test, (a) Drive speed, and (b) Turn speed

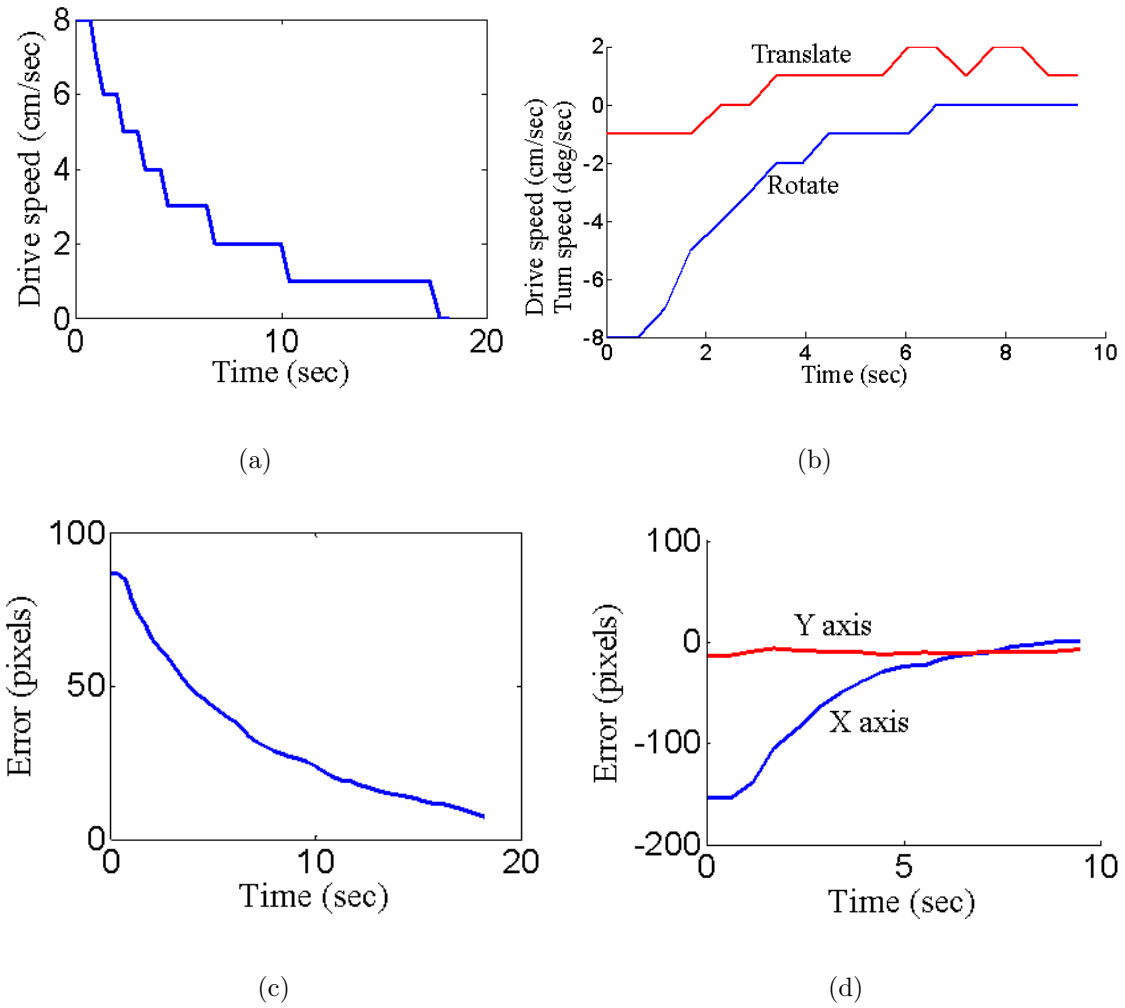
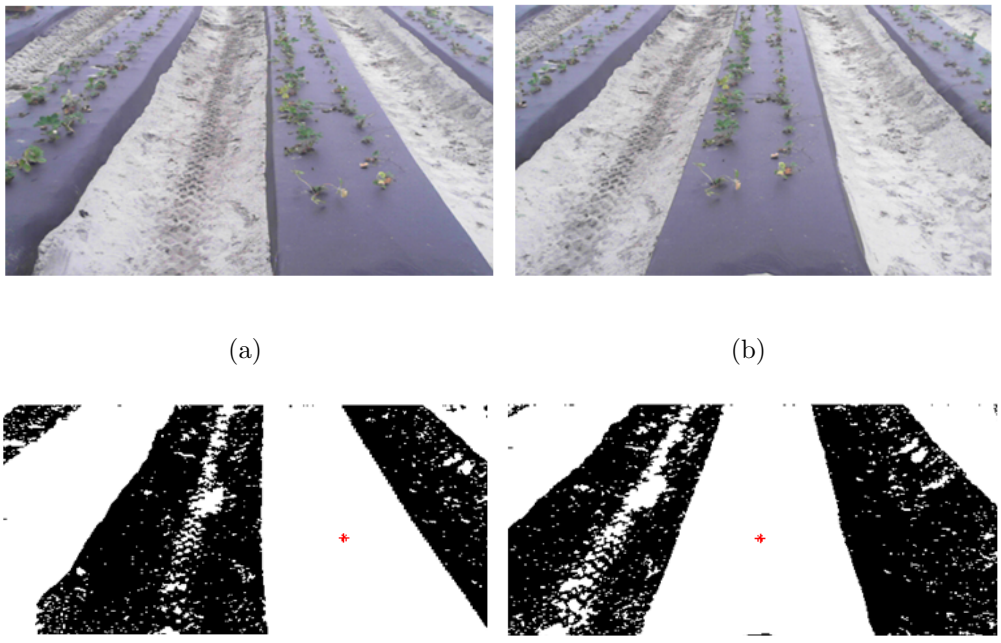


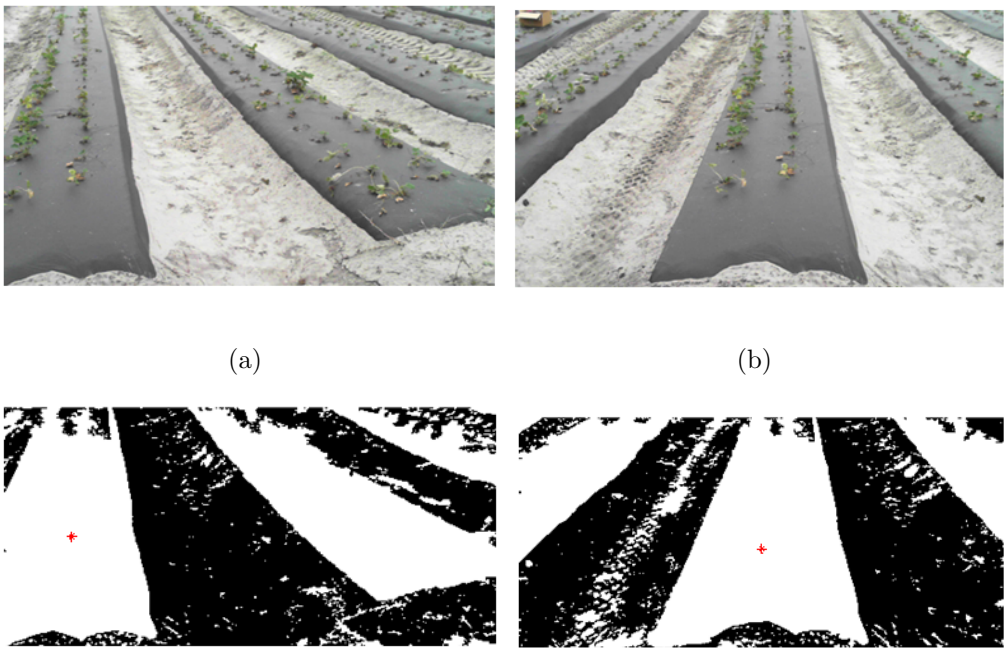
Figure 6.5: Control and Pixel Error, (a) Phase 3.1b, (b) Phase 3.2, (c) Phase 3.1b, (d) Phase 3.2



(a) (b)

(c) (d)

Figure 6.6: Phase 3.1b, (a) Initial frame, (b) Final frame, (c) Initial frame filtered (d) Final frame filtered



(a) (b)

(c) (d)

Figure 6.7: Phase 3.2, (a) Initial frame, (b) Final frame, (c) Initial frame filtered (d) Final frame filtered

6.2.4 Three Bed Test Right Turn

To prove the capability of DDAGR, the final trial involved driving completely to the end of two rows and onto the third row. Similarly, a right turn for the first cross-bed motion with vision feedback was first implemented with an undershoot for the initial conditions for x translation of the first control and θ for the multi-input controller. The second left turn cross-bed motion operated purely from encoder feedback to test how well the robot could capably align with a dead-reckoning attempt. This experiment was conducted on the north side of the strawberry field and had multiple differences than compared with the southern side which justifies the encoder only feedback for the cross-bed portion. The most significant of these were, due to the elevation, water would collect making the soil fairly dense compared to other portions, however this also cause a variety of colors which are less discernible from the actual plastic covered bed. Likewise, reflection from the sun onto the plastic would cause saturation of the webcamera sensor during preliminary samples as the camera was now facing the sun.

It is important to note the Over-bed drive speed of DDAGR was increased from 15 cm/sec to 23 cm/sec for this experiment for two reasons. 1) Prove the robustness of the Over-bed algorithm in it's final form and 2) Demonstrate the vehicle travel efficiency has been improved. Nevertheless, the overall vehicle path is shown in **Fig.** 6.8 Evident in the figure is the capability of the vehicle to travel the full extent of the strawberry row (128

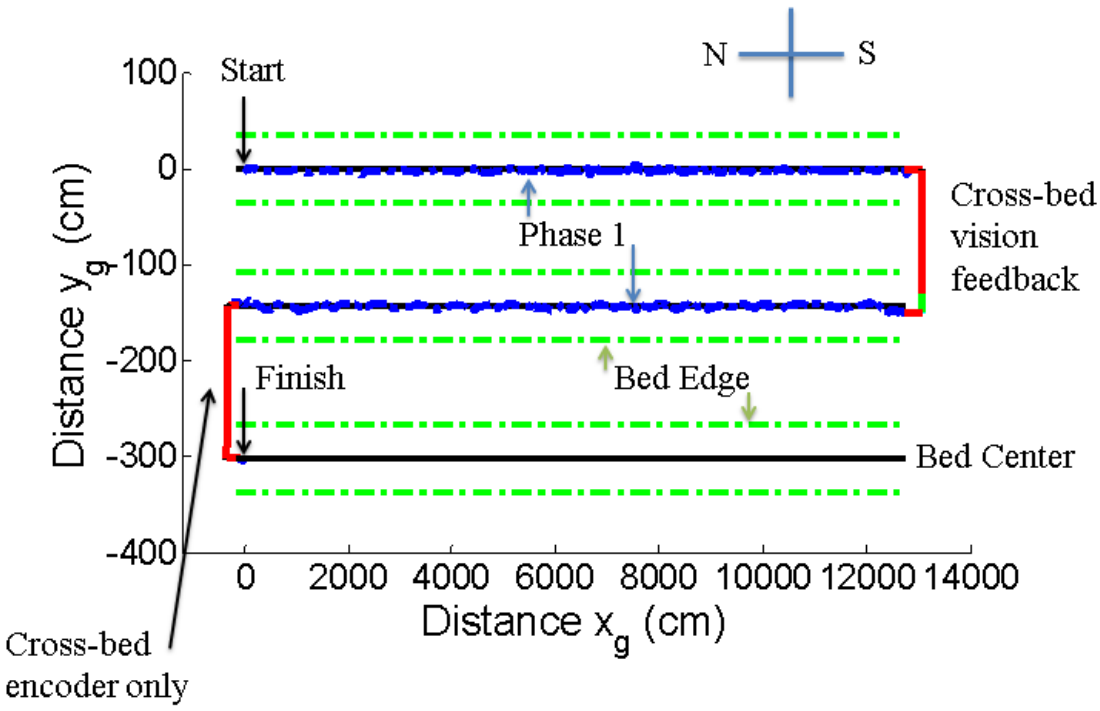
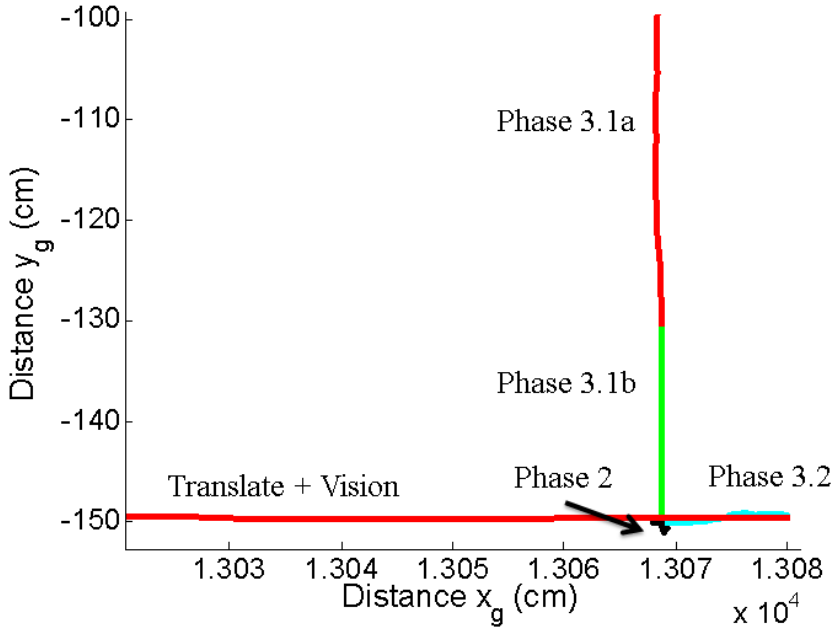
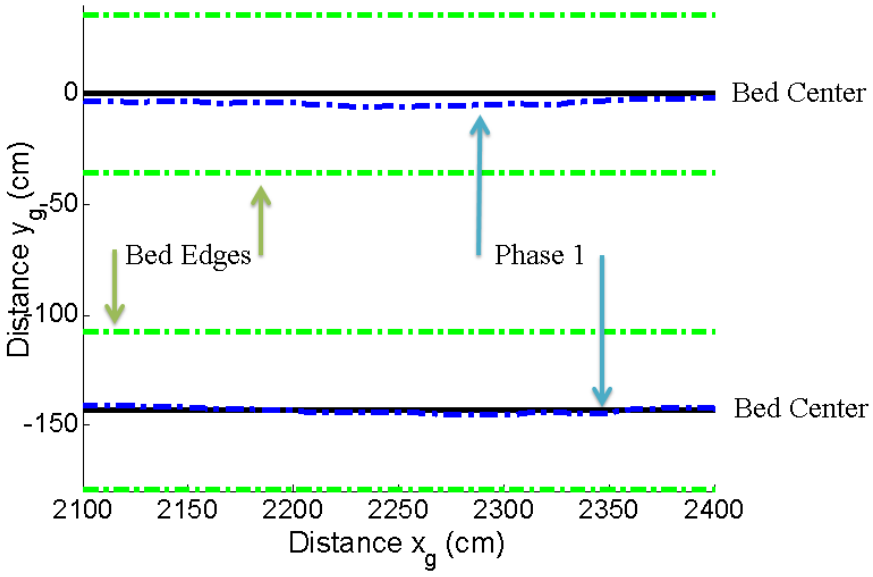


Figure 6.8: Vehicle path three bed field experiment



(a)



(b)

Figure 6.9: Excerpt of vehicle path for three bed test (a) Cross-bed, and (b) Over-bed

meters), detect the end of the row, perform the cross-bed control phase and continue on to the next over-bed portion.

A zoomed in portion of the Cross-bed vehicle path is presented in **Fig.** 6.9(a), some of the variance from perfect motion can be perceived. Similarly, it is evident that the vehicle performs the desired motion to properly align with the next strawberry row despite the mentioned disturbances. In **Fig.** 6.9(b) a small portion of the Over-bed phase is shown for both rows, with the axes scaled to the same order. The highlight from this figure is the ability of the controller to follow the bed center considering the numerous sources of uncertainty, ie. bed width, wheel slip, as well as measurement error. Overall the largest offset error e_y for the top row was -6.2 cm while the average error was only -2.0 cm. For the bottom row the maximum offset error was determined to be 7.2 cm with an average of -0.4 cm. The average e_θ was found to be 0.17° and -0.16° for the top and bottom row respectively.

Figure 6.10 displays the vision based controllers with the corresponding pixel errors. As expected the control gains as well as pixel errors approach zero over the duration of the experiment. Also evident is the capability of the controller to adjust from a disturbance that caused overshoot. In review of the images from the test it was apparent that the image filter had significant noise due to cloud cover occurring during the trial at approximately 4-5 seconds. Despite this external disturbance and after changing sign, the control gain converges to zero and reduces the pixel errors to within the desired tolerance within the allotted amount of time. Figure 6.11 shows a time progression for the x-axis single input controller. The first frame shows the beginning position with respect to the row before the

controller actuates, while the final frame shows the convergence of the strawberry row to the center of the image. The approximate duration between each frame is about 1 second.

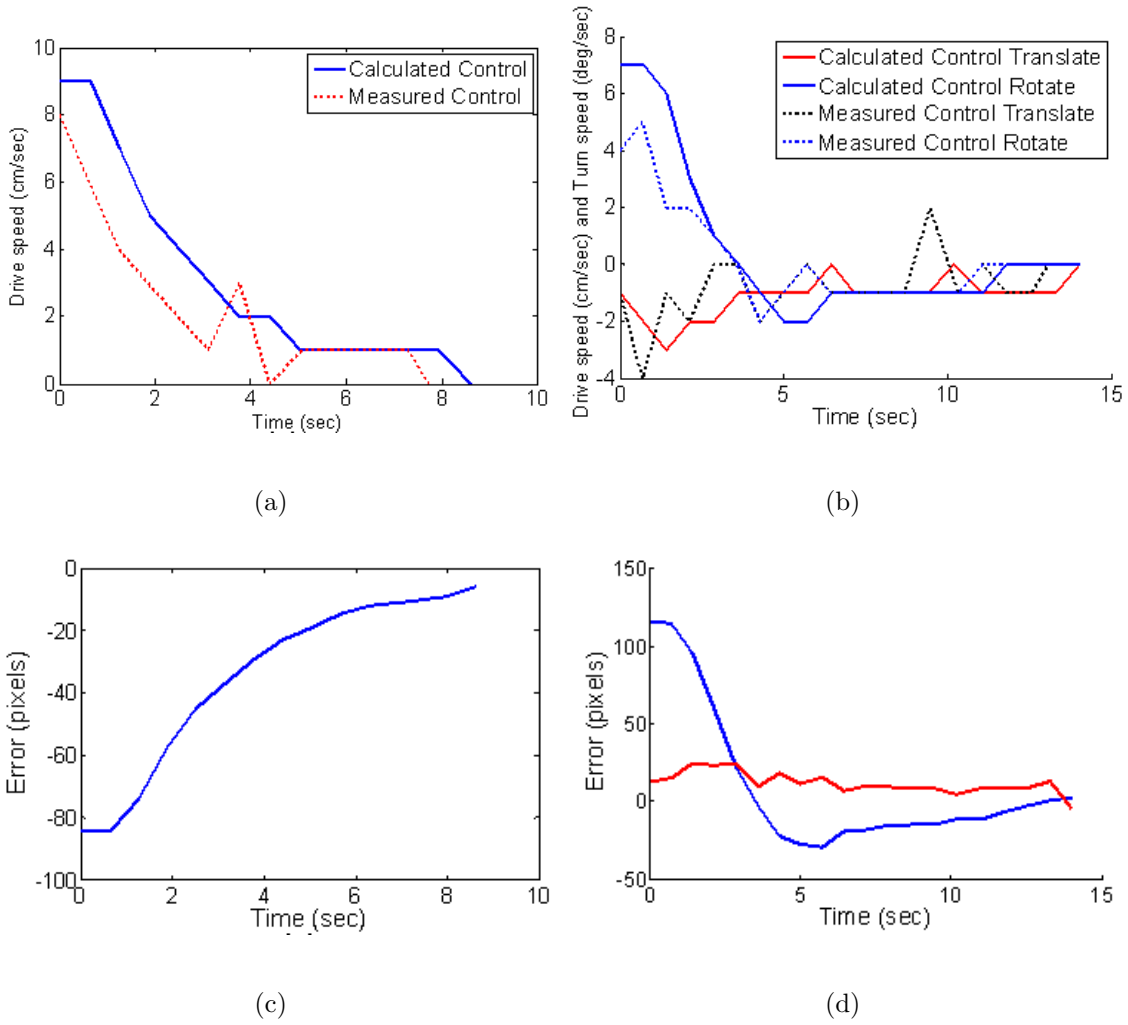


Figure 6.10: Control and Pixel Error, (a) Phase 3.1b, (b) Phase 3.2, (c) Phase 3.1b, (d) Phase 3.2

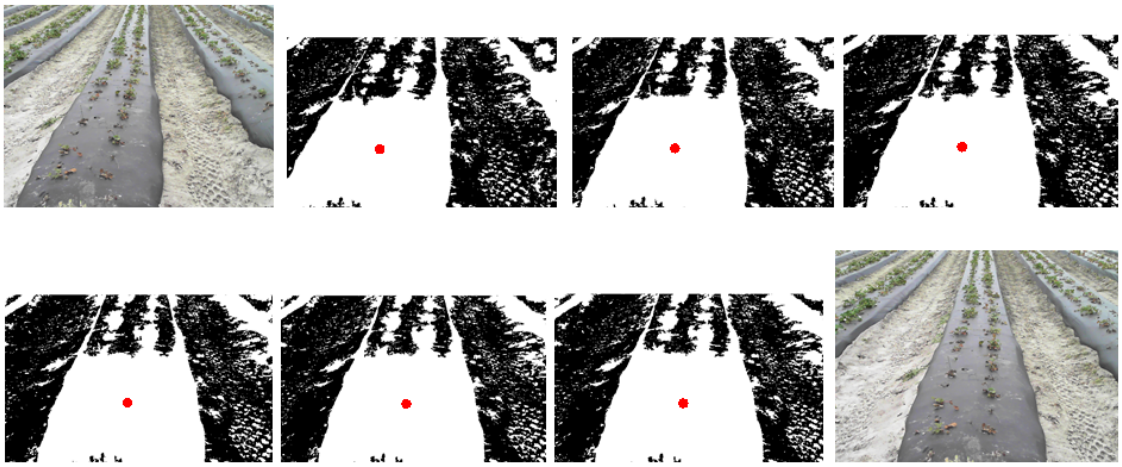


Figure 6.11: Time-lapse of single input controller, initial frame is top left

CHAPTER 7

FUTURE WORK AND CONCLUSIONS

7.1 System Integration with Leaf Picking Mechanism and Future Work

The final objective of the autonomous robotic system is to perform disease detection efficiently at many locations throughout the farm. The first intuitive step is to configure the disease detecting sensor to work with Matlab and attach it to the robot system. One of the goals of the autonomous vehicle is to store the dimension information for where the disease is located within the field. This will not be too difficult of a task for a single row as the encoder information will be available however for multiple rows it will be required to log GPS information. Once all the subsystems have been integrated the overall efficiency of this disease detection method will be evaluated. Ideally, it is desirable for the robot to visit as many suspected disease locations as possible in a given timeframe. Therefore, increasing of the travel speed while journeying down the rows, which constitutes the majority of travel time (estimated as $> 80\%$) will improve the evaluation and assessment. It will be necessary for the vehicle to operate at different stages of production which will also provide some variance in the aspect of field conditions. Therefore, full system tests will be desired at the beginning, middle as well as end of the production season.

In conjunction with the addition of the spectral sensor as one of the components of the autonomous system, classification of the strawberry plants is an essential step. The concept is to measure the spectral response of several healthy plants at multiple times during the season. This data will be used as training information for the classification algorithm and to which spectral responses of new samples will be compared. Identification as healthy, asymptomatic as well as symptomatic will then be possible for any of the data sets collected by the system.

While the vision system proved fairly robust to variable lighting conditions in the sense of shading from clouds occurring during some experiments, there is opportunity for improvement. In addition to the intensity change that can occur from cloud cover, when the webcamera faces the sun, reflectance from the plastic covering the strawberry row can cause saturation of the sensor. Implementation of lens polarization may provide heightened precision and increased image filtering accuracy. It was also determined that the morphological RGB and centroid method produced better preliminary results compared to the Hough Transform for the real strawberry environment. The primary advantage of the HT method over the centroid method is the time required, similarly it may also provide a better centerline approximation for the bed if the sensor has been saturated due to the sunlight from utilization of intensity gradients. Therefore, attempting the Hough Transform for the filtering technique is a plausible endeavor. Likewise, conversion to an alternative colorspace other than RGB which focuses more on hue, saturation or tone may prove to depict a greater variance between the elevated bed and the walking path along the side. It would be worthwhile

to statistically compare the variance between multiple methods and elect the best based on evaluation metrics of cost in the sense of amount of time, as well as overall centerline detection accuracy.

Intuitively, the navigation algorithm will be capable to operate in any plasticulture farming environment. Some minor adjustments such as travel dimensions and other tunable parameters can be easily modified to accommodate a variety of fields and settings. From this point, it would be appropriate to test the robotic system on other crops grown in a similar manner to strawberries, the varieties of which includes: squash, onion, cabbage as well as peppers.

7.2 Conclusions

Autonomous path control is required for an agricultural robot to safely and efficiently scout throughout commercial fields to conduct labor intensive operations such as harvesting and disease detection. In this research, three different controllers are designed for three distinct portions in the over-bed and cross-bed motions of a typical commercial strawberry field considering terrain variability, field patterns, and uncertainties in sensed information. With practical considerations, range finders are used in the over-bed PID controller, while RGB cameras are used in the cross-bed nonlinear robust controllers. Encoder feedback of the drivetrain is monitored throughout the entire operation. The over bed portion of the algorithm incorporated data filtering, to eliminate noises and errors, as well as detection of the end of

the row which is essential to begin the next control phase accurately. By inclusion of machine vision feedback along with several reasonable assumptions of the field such as operating on a plane, transferring from one strawberry row to another is accomplished. Both of the nonlinear controllers utilize pixels from collected images to formulate an error parameter to be tracked during the vehicle motion as it is controlled to align with the desired strawberry row. The closed-loop vision system under bounded uncertainties in the cross-bed portion is proven to be asymptotically stable, and the reachability analyses are conducted to make sure the state dispersion in the end of one phase is within the initial condition bounds of the next phase. Successive phases of: translation (3.1a), single input nonlinear controller (3.1b), pure rotation (2), and finally, arcing motion from the multi input multi output nonlinear controller (3.2) were depicted in the reachable sets trajectories, as well as the real motion of the vehicle in the field. The controllers were simulated prior to successful demonstration in multiple experiments at several different commercial strawberry farms.

In addition to the contributions of applying different sensing (e.g. a single fixed camera) and control technologies to drive an agricultural robot in and out of semi-structured strawberry rows, it is shown that with the small heading angle assumption in the over-bed motion, the uncertainty bound relating to the camera focal length is in a very simple form and usable in the nonlinear controllers development. Conclusively, the accuracy for each independent phase was evaluated from field experimentation and subsequently used for tuning. As a final demonstration, the unmanned ground vehicle executed the navigation steps consecutively for full path motion from one location in the strawberry field to another.

Considering the variability and uncertainty of a mechanical vehicle operating in a real farming environment, DDAGR's navigation capability is comparable to that of other agricultural vehicles based on an accuracy of several centimeters for distance as well as within a couple degrees for the heading angle. By utilization of multiple economical sensory devices and the developed control algorithms, in comparison to intricate and expensive navigation systems, the subsequent tasks of field scouting, and sampling are efficiently conducted.

In the context of disease detection for agricultural crops, the benefits for this technology and primary motivation for this research comprise: (1) capability of accurate and repeatable results considering the environment for imaging, as the workspace under the vehicle will be controlled. (2) The detection task can be performed continually throughout the production season, with information of the sample locations such as coordinates saved for future reference. (3) Disease detection for this specific crop is currently conducted by visual inspection which has the following drawbacks, overlooked locations, prevalence is dependent upon progression into the season and finally the spectral technology that will be attached to the leaf picking mechanism can perform at an earlier stage.

LIST OF REFERENCES

[1] C. Bravo, D. Moshou, J. West, A. McCartney, and H. Ramon, "Early disease detection in wheat fields using spectral reflectance," *Biosystems Engineering*, vol. 84, no. 2, pp. 137–145, 2003.

[2] D. Moshou, C. Bravo, R. Oberti, J. West, H. Ramon, V. S., and B. D., "Intelligent multi-sensor system for the detection and treatment of fungal diseases in arable crops," *Biosystems Engineering*, vol. 108, pp. 311–321, 2011.

[3] S. G. Defterli, Y. Xu, Y. Shi, and R. Ehsani, "Review of robotic technology for strawberry production," *American Society of Agricultural and Biological Engineers - Applied Engineering in Agriculture*, vol. 32, no. 3, pp. 301–318, 2016.

[4] S. M. Pedersen, S. Fountas, H. Have, and B. S. Blackmore, "Agricultural robots-system analysis and economic feasibility," *Precision Agriculture*, vol. 7, no. 4, pp. 295–308, 2006.

[5] M. Li, K. Imou, K. Wakabayashi, and S. Yokoyama, "Review of research on agricultural vehicle autonomous guidance," *International Journal of Agricultural and Biological Engineering*, vol. 2, no. 3, pp. 1–16, 2009.

[6] S. Hiremath, "Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter," *Computers and Electronics in Agriculture*, vol. 100, no. 3, pp. 41–50, 2014.

[7] F. Dong, W. Heinemann, and R. Kasper, "Development of a row guidance system for an autonomous robot for white asparagus harvesting," *Computers and Electronics in Agriculture*, vol. 79, no. 2, pp. 216–225, 2011.

[8] M. O'Connor, T. Bell, G. Elkaim, and B. Parkinson, "Automatic steering of farm vehicles using gps," *Precision Agriculture*, vol. 79, no. 2, pp. 767–777, 1996.

[9] T. Oksanen, M. Linja, and A. Visala, "Low-cost positioning system for agricultural vehicles," in *Computational Intelligence in Robotics and Automation*, pp. 297–302, IEEE, 2005.

[10] T. Hague, J. A. Marchant, and N. D. Tillett, "Ground based sensing systems for autonomous agricultural vehicles," *Computers and Electronics in Agriculture*, vol. 25, no. 1, pp. 11–28, 2000.

[11] Y. Nagasaka, H. Saito, K. Tamaki, M. Seki, K. Kobayashi, and K. Taniwaki, "An autonomous rice transplanter guided by global positioning system and inertial measurement unit," *J. of Field Robotics*, vol. 26, no. 6-7, pp. 537–548, 2009.

[12] M. Joerger and B. Pervan, "Measurement-level integration of carrier-phase gps and laser-scanner for outdoor ground vehicle navigation," *ASME J. Dyn. Syst. Meas. Control*, vol. 131, no. 2, pp. 021004–021004–11, 2009.

[13] J. Reid and S. Searcy, *Vision-based guidance of an agriculture tractor*. IEEE, control systems magazine ed., 1987.

[14] D. Khadraoui, C. Debain, R. Rouveure, P. Martinet, P. Bonton, and J. Gallice, "Vision-based control in driving assistance of agricultural vehicles," *The International Journal of Robotics Research*, vol. 17, no. 10, pp. 1040–1054, 1998.

[15] M. Kise, Q. Zhang, and F. Rovira Mas, "A stereovision-based crop row detection method for tractor-automated guidance," *Biosystems Engineering*, vol. 90, no. 4, pp. 357–367, 2005.

[16] Q. Wang, Q. Zhang, F. Rovira-Mas, and T. Tian, "Stereovision-based lateral offset measurement for vehicle navigation in cultivated stubble fields," *Biosystems Engineering*, vol. 109, no. 4, pp. 258–265, 2011.

[17] V. Subramanian, T. F. Burks, and A. A. Arroyo, "Development of machine vision and laser radar based autonomous vehicle guidance systems for citrus grove navigation," *Computers and Electronics in Agriculture*, vol. 53, no. 2, pp. 130–143, 2000.

[18] M. Sharifi and X. Chen, "A novel vision based row guidance approach for navigation of agricultural mobile robots in orchards," in *International Conference on Automation, Robotics and Applications*, (Queenstown, New Zealand), IEEE, Feb. 2015.

[19] T. Bakker, H. Wouters, K. Asselt, L. T. Bontsema, M. J., and S. G., "A vision based row detection system for sugar beet," *Computers and Electronics in Agriculture*, vol. 60, no. 1, pp. 87–95, 2008.

[20] V. Leemans and M.-F. Destain, "Application of the hough transform for seed row localisation using machine vision," *Biosystems Engineering*, vol. 94, no. 3, pp. 325–336, 2006.

[21] J. Marchant and R. Brivot, "Real-time tracking of plant rows using a hough transform," *Real-Time Imaging*, vol. 1, pp. 363–371, 1995.

[22] J. D. Will, J. F. Reid, Q. Zhang, and F. Rovira-Mas, "Hough-transform-based vision algorithm for crop row detection of an automated agricultural vehicle," in *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 219, pp. 999–1010, 2005.

[23] B. Astrand and A.-J. Baerveldt, "A vision based row-following system for agricultural field machinery," *Mechatronics*, vol. 15, no. 2, pp. 251–269, 2005.

[24] R. Ji and L. Qi, "Crop-row detection algorithm based on random hough transformation," *Mathematical and Computer Modelling*, vol. 54, no. 3-4, pp. 1016–1020, 2011.

[25] O. C. Barawid, A. Mizushima, K. Ishii, and N. Noguchi, "Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application," *Biosystems Engineering*, vol. 96, no. 2, pp. 139–149, 2007.

[26] G. Jiang, Z. Wang, and H. Liu, "Automatic detection of crop rows based on multi-rois," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2429–2441, 2015.

[27] J. Xue, L. Zhang, and T. E. Grift, "Variable field-of-view machine vision based row guidance of an agricultural robot," *Computers and Electronics in Agriculture*, vol. 84, pp. 85–91, 2012.

[28] Z. Tian, X. Junfang, W. Gang, and Z. Jianbo, "Variable field-of-view machine vision based row guidance of an agricultural robot," *International Journal of Agricultural and Biological Engineering*, vol. 7, no. 4, pp. 29–37, 2014.

[29] T. Zhang, J. Xia, G. Wu, and J. Zhai, "Automatic navigation path detection method for tillage machines working on high crop stubble fields based on machine vision," *International Journal of Agriculture and Biological Engineering*, vol. 7, no. 4, pp. 29–37, 2014.

[30] M. Patel, C. Kathuria, A. Singla, H. Sahoo, M. Nayak, and C. Ghanshyam, "Navigation control of a mobile robotic system in semi-structured and dynamic environment for controlled dose delivery of pesticides," in *International Conference on Trends in Automation, Communication and Computing Technologies*, (Bangalore, India), IEEE, Dec. 2015.

[31] J. Billingsley and M. Schoenfisch, "The successful development of a vision guidance system for agriculture," *Computers and Electronics in Agriculture*, vol. 16, pp. 147–163, 1997.

[32] A. Haytham, Y. Z. Elhalwagy, A. Wassal, and N. M. Darwish, "Modeling and simulation of four-wheel steering unmanned ground vehicles using a pid controller," in *International Conference on Engineering and Technology*, IEEE, 2014.

[33] A. English, P. Ross, D. Ball, and P. Corke, "Vision based guidance for robot navigation in agriculture," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2014.

[34] A. Velasquez, V. Higuti, D. Milori, D. Magalhaes, and M. Becker, "Helvis - a small-scale agricultural mobile robot prototype for precision agriculture," in *13th International Conference On Precision Agriculture*, (St. Louis, Missouri), IEEE, 2016.

[35] E. Kayacan, E. Kayacan, H. Ramon, O. Kaynak, and W. Saeys, "Towards agrobots: Trajectory control of an autonomous tractor using type-2 fuzzy logic controllers," *Transactions on Mechatronics*, vol. 20, no. 1, pp. 287–298, 2015.

[36] J. M. Bengochea-Guevara, J. Conesa-Munoz, D. Andujar, and A. Ribeiro, "Merge fuzzy visual servoing and gps-based planning to obtain a proper navigation behavior for a small crop-inspection robot," *Sensors*, vol. 16, no. 3, p. 276, 2014.

[37] A. Marcos, C. Soria, and R. Carelli, "Visual servo control of a mobile robot in agriculture environments," *Mechanics Based Design of Structures and Machines*, vol. 36, no. 4, pp. 392–410, 2008.

[38] T. Utstumo, T. W. Berge, and J. T. Gravdahl, "Non-linear model predictive control for constrained robot navigation in row crops," in *International Conference on Industrial Technology (ICIT)*, IEEE, 2015.

[39] T. Kraus, H. J. Ferreau, E. Kayacan, H. Ramon, J. DeBaerdemaeker, M. Diehl, and W. Saeys, "Moving horizon estimation and nonlinear model predictive control for autonomous agricultural vehicles," *Computers and Electronics in Agriculture*, vol. 98, pp. 25–33, 2013.

[40] A. S. Matveev, M. Hoy, J. Katupitiya, and A. V. Savkin, "Nonlinear sliding mode control of an unmanned agricultural tractor in the presence of sliding and control saturation," *Robotics and Autonomous Systems*, vol. 61, no. 9, pp. 973–987, 2013.

[41] T. Utstumo, T. W. Berge, and J. T. Gravdahl, "Robust sliding mode control of an agricultural tractor under the influence of slip," in *International Conference on Advanced Intelligent Mechatronics*, (Singapore), IEEE/ASME, July, 14-17 2009.

[42] D. Bevly, J. C. Gerdes, and W. Parkinson, B., "A new yaw dynamic model for improved high speed control of a farm tractor," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 124, no. 4, pp. 659–667, 2002.

[43] L. Garca-Prez, M. C. Garca-Alegre, and A. R. Guinea, "An agent of behaviour architecture for unmanned control of a farming vehicle," *Computers and Electronics in Agriculture*, vol. 60, no. 1, pp. 39–48, 2008.

[44] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[45] M. Althoff, O. Stursberg, and M. Buss, "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization," in *IEEE Conference on Decision and Control (CDC)*, IEEE, 2008.

[46] W. Langson, I. Chryssochoos, S. V. Rakovi, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.

[47] R. Gonzlez, M. Fiacchini, G. J. L., T. lamo, and F. Rodrguez, "Robust tube-based predictive control for mobile robots in off-road conditions," *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 711–726, 2011.

[48] E. Kayacan, E. Kayacan, H. Ramon, and W. Saeys, "Robust tube-based decentralized nonlinear model predictive control of an autonomous tractor-trailer system," *Transactions on Mechatronics*, vol. 20, no. 1, pp. 447–456, 2015.

[49] M. Hudson, "Florida agriculture by the numbers," 2014.

[50] J. Hancock, *Temperate Fruit Crops in Warm Climate*. Kluwer Academic Publisher, 2000.

[51] A. Savkin and C. Wang, "A framework for safe assisted navigation of semi-autonomous vehicles among moving and steady obstacles," *Robotica*, pp. 1–25, 2016. Available on CJO 2016 doi:10.1017/S0263574715000922.

[52] N. Li, C. Remeikas, Y. Xu, S. Jayasuriya, and R. Ehsani, "Task assignment and trajectory planning algorithm for a class of cooperative agricultural robots," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 5, 2015.

[53] P. Menendez-Aponte, C. Garcia, D. Freese, S. G. Defterli, and X. Y., "Software and hardware architectures in cooperative aerial and ground robots for agricultural disease detection," in *International Conference on Collaboration Technologies and Systems*, (Orlando), Oct. 2016.

[54] D. Freese and Y. Xu, "Nonlinear robust path control for a field robot scouting in strawberry orchards," in *Dynamic Systems and Control Conference*, (Tysons, Virginia), ASME, Oct. 2017.

[55] W. Pavan, C. Fraisse, and N. Peres, "Development of a web-based disease forecasting system for strawberries," *Computers and Electronics in Agriculture*, vol. 75, pp. 169–175, 2011.

[56] S. Sankaran, A. Mishra, R. Ehsani, and C. Davis, "A review of advanced techniques for detecting plant diseases," *Computers and Electronics in Agriculture*, vol. 72, pp. 1–13, 2010.

[57] H. Muhammed and A. Larsolle, "Feature vector based analysis of hyperspectral crop reflectance data for discrimination and quantification of fungal disease severity in wheat," *Biosystems Engineering*, vol. 86, no. 2, pp. 125–134, 2003.

[58] Y.-H. Yeh, W.-C. Chung, J.-Y. Liao, C.-L. Chung, Y.-F. Kuo, and T.-T. Lin, "Strawberry foliar anthracnose assessment by hyperspectral imaging," *Computers and Electronics in Agriculture*, vol. 122, pp. 1–9, 2016.

[59] R. Siegwart, R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. Cambridge, MA.: The MIT Press, 2004.

[60] Z. Qu, *Cooperative control of Dynamical Systems: Applications to Autonomous Vehicles*. Springer-Verlag London Limited Publisher, 2009.

[61] J. P. Laumond, S. Sekhavat, and F. Lamiraux, *Guidelines in nonholonomic motion planning for mobile robots. Robot Motion Planning and Control*. Springer-Verlag London Limited Publisher, 1998.

[62] P. Salaris, F. A. W. Belo, D. Fontanelle, L. Greco, and A. Bicchi, "Optimal paths in a constrained image plane for purely image-based parking," in *International Conference on Intelligent Robots and Systems*, (Nice, France), pp. 1673–1680, IEEE/RSJ, 2008.

[63] M. Serrano, S. A. Godoy, V. A. Mut, O. A. Ortiz, and J. E. Scaglia, "A nonlinear trajectory tracking controller for mobile robots with velocity limitation via parameters regulation," *Robotica*, pp. 1–20, 2016.

[64] S. Han, Q. Zhang, B. Ni, and J. Reid, "A guidance directrix approach to vision-based vehicle guidance systems," *Computers and Electronics in Agriculture*, vol. 43, pp. 179–195, 2004.

[65] G. Jiang, X. Wang, Z. Wang, and H. Liu, "Wheat rows detection at the early growth stage based on hough transform and vanishing point," *Computers and Electronics in Agriculture*, vol. 123, pp. 211–223, 2016.

[66] J. Y. Bouguet, *Camera Calibration Toolbox for Matlab*.

[67] Y. Xu and E. Ritz, "Vision based flexible beam tip point control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 17, no. 5, pp. 1220–1227, 2009.

[68] J. J. Slotine and W. Li, *Applied Nonliear control*. New Jersey: Prentice-Hall, 1998.

[69] M. Asif, J. M. Khan, and A. Memon, "Integral terminal sliding mode formation control of non-holonomic robots using leader follower approach," *Robotica*, pp. 1–15, 2016.

[70] E. Asarin, T. Dang, and A. Girard, "Reachability analysis of nonlinear systems using conservative approximation," in *International Conference on Hybrid Systems: Computation and Control*, vol. 2623, (Prague, Czech Republic), pp. 20–35, ACM, January 2003.

[71] M. Althoff and J. M. Dolan, "Reachability computation of low-order models for the safety verification of high-order road vehicle models," in *American Controls Conference*, vol. 2623, (Montreal, Canada), pp. 3559–3566, IEEE, June 2012.

[72] M. Altoff, *CORA 2016 Manual*. Technische Universitat Munchen, 85748 Garching, Germany, 2016.

[73] A. Girard, "Reachability of uncertain linear systems using zonotopes," in *International Conference on Hybrid Systems: Computation and Control*, vol. 5, (Zurich, Switzerland), pp. 291–305, ACM, March 2005.