Received: 8 May 2021 | Revised: 21 October 2021 | Accepted: 28 November 2021

DOI: 10.1002/rob.22053

RESEARCH ARTICLE

WILEY

# Minimum-time row transition control of a vision-guided agricultural robot

**Qiang Li** | **Yunjun Xu**

Department of Mechanical and Aerospace Engineering, University of Central Florida, Orlando, Florida, USA

**Correspondence**
Yunjun Xu, Department of Mechanical and Aerospace Engineering, University of Central Florida, Orlando, FL, USA.
Email: yunjun.xu@ucf.edu

**Funding information**
National Science Foundation, Grant/Award Number: 1924622

## Abstract

This paper presents a vision-based, subspace optimal controller aiming to improve the row transition performance of an agricultural robot in a strawberry field. The contribution of this paper is twofold. First, only RGB cameras, instead of complicated sensor suites, are used for cross-bed navigation and row alignment. Second, a real-time adaptive dynamic programming-based algorithm is designed for an optimal row transition. The conditions for row alignment are derived in an augmented pixel coordinate frame. Based on these conditions, a simple motion rule is utilized to reduce the search space dimension so that the proposed algorithm can be implemented in real-time. Additionally, the inverse-dynamics policy of the algorithm is updated using vision feedback at each control step to adapt to uncertainties. The proposed controller is tested in both simulations and field experiments. In a simulation comparison, the minimum-time solution achieved using the proposed algorithm is 44.7 s, which is very close to that of a benchmark algorithm (44.4 s). However, the CPU time required by the proposed algorithm is only 4.3% of time needed by the benchmark algorithm. Twenty field experiments using the presented design were all successful in row transition, with a mean final alignment error of 0.5 cm.

KEYWORDS

agricultural robotics, autonomous navigation, field robotics, motion control, vision processing

## 1 | INTRODUCTION

Recent advances in mechatronics, robotics, and machine learning techniques, as well as the prevalence of low-cost sensors, have prompted the development of agricultural robots to mitigate the pressure from an increase in global population (United Nations, 2019; Valin et al., 2014) and the ever-increasing labor costs (Shangguan et al., 2021). Agricultural robots with custom-designed hardware and software are expected to fulfill different requirements in agricultural field operations with minimal human interference. To date, many different robots have been developed and/or deployed in both greenhouse environments (Arad et al., 2020; Mahmud et al., 2019; Schor et al., 2016; van Henten et al., 2002; Xiong et al., 2020) and field conditions (Åstrand & Baerveldt, 2002; Birrell et al., 2020; Nagasaka et al., 2009; Underwood et al., 2017). Growers are enthusiastic about using robotic technologies to reduce labor costs and increase profit margins in harvesting (Arad et al., 2020; Birrell et al., 2020; van Henten et al., 2002; Xiong et al., 2020), phenotyping (Narvaez et al., 2017; Underwood et al., 2017), transplanting (Nagasaka et al., 2009), weed control (Åstrand & Baerveldt, 2002), disease detection (Dusadeerungsikul & Nof, 2019; Schor et al., 2016), and so on.

A reliable motion control system is crucial to the success of a mobile agricultural robot, whether it is used in greenhouses or field environments. Navigation in a greenhouse is relatively easy, as the environment is usually known and well-structured (Le et al., 2020). In many cases, robots are limited to move on constrained track systems (Arad et al., 2020; van Henten et al., 2002). In comparison, controlling agricultural robots in an unstructured or semistructured field is challenging. For a large portion of farmlands that have row crops, robot motion control normally consists of row-wise translations (or over-bed) and row transitions at headland (or cross-bed) (Freese & Xu, 2019; Underwood et al., 2017). There is a rich literature on row-wise translation in different types of farmlands (Ball et al., 2016; de Paula Veronese et al., 2016; Dong et al., 2011; Freese & Xu, 2019; Higuti et al., 2019; Le et al., 2020; Nagasaka et al., 2009; Schwendner et al., 2014; Underwood et al., 2015, 2017; van Henten et al., 2002; Winterhalter et al., 2021; Xiong et al., 2020); nevertheless, studies on row transition controls are limited (Ball et al., 2016; Freese & Xu, 2019; Le et al., 2020; Li et al., 2020; Nagasaka et al., 2009; Underwood et al., 2017; Xiong et al., 2020). In this paper, we present a real-time subspace optimal control method that significantly improves the row transition performance of a field robot in a strawberry field.

## 1.1 | Related work

A survey of state-of-the-art agricultural robots reveals that the majority of existing motion control methods rely on global positioning system (GPS; Ball et al., 2016; de Paula Veronese et al., 2016; Nagasaka et al., 2009; Tu et al., 2019; Underwood et al., 2017; Winterhalter et al., 2021) or time-of-flight (TOF) sensors like laser scanners (Le et al., 2020; Schwendner et al., 2014) and LIDAR (Higuti et al., 2019; Hiremath et al., 2014; Underwood et al., 2015; Xiong et al., 2020) for localization in a field. With the emergence of high accuracy GPS, centimeter-level navigation can be achieved at an accessible cost for long-term and offline planning (Ball et al., 2016; Thuilot et al., 2002). On the other hand, TOF-based methods can give robots situational awareness, which is ideal for navigating in a dynamic environment (Kragh & Underwood, 2020).

GPS- or TOF-based methods have some limitations. GPS signals are weak in tall-growing fields (Higuti et al., 2019) and can be easily perturbed by electromagnetic interference (Ball et al., 2016). Furthermore, building accurate GPS maps of an agriculture field is usually done by manually driving the robot throughout the area, which is time-consuming and labor-intensive. This approach is nonideal for cases where the GPS map needs to be built multiple times in one growing season (Defterli et al., 2016). Additionally, GPS needs to work with other sensors such as inertial measurement unit (IMU), camera, and/or LIDAR to provide orientation information (Bak & Jakobsen, 2004; Ball et al., 2016; de Paula Veronese et al., 2016; Nagasaka et al., 2009; Winterhalter et al., 2021). Meanwhile, TOF sensor-based methods need to prebuild a point cloud map of the environment for navigation (Schwendner et al., 2014; Underwood et al., 2015; Xiong et al., 2020). Because depth is a relative measurement, unless the point cloud map is thoroughly labeled (Underwood et al., 2015), such methods rely on additional sensors like encoders and/or cameras to help locate a robot on the map (Xiong et al., 2020). As outdoor LIDAR and laser scanners are already expensive, TOF-based solutions are costly if additional sensors are required.

Cameras are often integrated with GPS or TOF to provide visual information (Ball et al., 2016; Gai et al., 2021; Schwendner et al., 2014; Winterhalter et al., 2021). In Winterhalter et al. (2018) and Kneip et al. (2020), Hough pattern and stereovision-based row detection methods are proposed, respectively. Although camera-based navigation methods are popular in indoor environments (de la Puente & Rodríguez-Losada, 2014; Ji et al., 2015; Liang et al., 2013; Nguyen et al., 2020), there is not much research done in terms of using camera-only methods for outdoor agricultural fields, whether in row-wise translation or row transition. Hague and Tillett (1996) proposes to use vision-based sensing techniques to localize the robot with respect to crop rows. Our previous work (Freese & Xu, 2019) uses the color pattern of a typical row-based field to transit between rows, which is not robust in varying lighting conditions and often results in alignment errors when the rows are not perfectly straight. In Li et al. (2020), the visual marker-based cross-bed motion is studied, which can mitigate the influence of changing lighting conditions. These two existing control schemes (Freese & Xu, 2019; Li et al., 2020) are not optimal. It is worth noting that visual fiducial systems that provide measurements by visual markers have been widely used in robotics (Fiala, 2005; Kalaitzakis et al., 2021; Olson, 2011). For example, visual fiducial systems use a two-dimensional (2D) barcode to provide relative position and orientation information between cameras and tags (Fiala, 2005; Olson, 2011).

Proportional-integral-derivative (PID) controllers, and their variations, are the most widely used methods in row transition and/or headland maneuvering that follow paths or waypoints that are preplanned using different methods (Bakker, 2009; Ball et al., 2016; Le et al., 2020; Nagasaka et al., 2009; Shojaei, 2021; Underwood et al., 2017; Xiong et al., 2020). A pole detection algorithm is used in Le et al. (2020) to identify row ends and then square trajectories are followed in row transition using a pure pursuit controller. In Underwood et al. (2017), the shortest path is selected in a manually built map for the Ladybird robot to conduct row-crop phenotyping tasks. A metric map is prebuilt via SLAM in Xiong et al. (2020), and the strawberry picking robot follows straight-line trajectories connecting user-selected waypoints. The GPS/IMU-guided rice transplanter in Nagasaka et al. (2009) follows a fixed steering command during headland turns. A search-based lattice planner is used in Ball et al. (2016) to generate obstacle-free paths every 10 s.

To date, advanced control methods have yet to be used a lot in row transition motions. A fuzzy logic controller is designed for grain carts to mimic human behaviors in harvesting (Shangguan et al., 2021). A robust adaptive tracking controller is developed for a farm vehicle, where the time-invariant sliding is estimated via projection mapping and the time-varying sliding is compensated by variable structure control (Fang et al., 2006). Neural network and genetic algorithms are applied for path planning of an agricultural robot in simulation (Noguchi & Terao, 1997).

Field conditions such as terrain type and friction can have a big impact on the motion control performance of agricultural robots.

In some scenarios, ground condition is evaluated a priori, and unfavorable areas or paths are excluded (Angelova et al., 2007; Auat Cheein et al., 2017; Rankin & Matthies, 2010). For example, terrain appearance and geometry information are used in an experience-based method to predict slippage conditions, so that the robot can reroute to avoid such areas (Angelova et al., 2007). In other scenarios, adverse influence from poor slippage conditions will be compensated according to the actual deviation from its desired trajectories (Cariou et al., 2009; Kayacan et al., 2018). As an example, Cariou et al. (2009) estimated the skidding effect of a four-wheel-steering robot in low grip conditions based on the measured deviations in its lateral and angular movements.

### 1.2 | Our work and contributions

The presented work is a part of research designing a multitask field robot scouting throughout a typical strawberry field with raised beds. The robot transits between rows at headlands, only relying on RGB cameras. As discussed earlier, our previous row transition (or cross-bed) controllers follow a nonoptimal square trajectory (Freese & Xu, 2019; Li et al., 2020). Additionally, sands in a typical strawberry field are loose and tend to build up under the wheels when the robot is turning, causing both slippage and inconsistent turns.

In this paper, adaptive dynamic programming (ADP)-based subspace optimal control is designed to improve the field robot row transition performance while sticking to RGB cameras only. The main contributions are summarized as follows:

(1) The center point of the visual marker at a row headland is projected in an augmented pixel coordinate frame. Via this projection, the row alignment problem during row transition is converted into a pixel trajectory tracking problem.
(2) The necessary and sufficient conditions for row alignment are developed in the augmented pixel coordinate frame to guarantee a perfect row alignment in the local north-east-down (NED) coordinates.
(3) The real-time ADP subspace optimal control method (Li & Xu, 2020) is significantly revised and enhanced to be used in the augmented pixel coordinate frame. The search dimension is significantly reduced from 3D to 1D, so the computational cost and the memory usage are dramatically reduced. State and control constraints are considered, and the row transition time is greatly reduced.
(4) A vision-based adaptation rule is derived to update the uncertainty parameters in longitudinal and rotational mobilities of the robot at each time step.
(5) Installing visual markers in an agriculture field requires far less skills than building a GPS waypoint map or LIDAR point cloud map, thus the proposed row transition solution can be easily deployed in any row-based farmland. Furthermore, the solution only uses RGB cameras and is, therefore, much cheaper than GPS- or LIDAR-based solutions.

### 1.3 | Structure of the paper

Section 2 describes the field conditions of a typical strawberry field and introduces the robot platform. The row transition control problem is formulated in Section 3. Section 4 develops the row alignment conditions in the augmented pixel coordinate frame. The subspace, minimum-time ADP controller is presented for row transition in Section 5. Simulation validations and field trials are shown in Sections 6 and 7, respectively. The study is further discussed in Section 8 and concluded at the end.

## 2 | RESEARCH BACKGROUND

### 2.1 | Strawberry field navigation

Figure 1 shows the satellite photo of a nearby commercial strawberry field where our robot is tested. The dark vertical lines are the strawberry beds that are covered by black plastic films. The average distance between adjacent strawberry beds is 1.5 m. Our robot is


**FIGURE 1** A satellite photo of a local strawberry field (Google maps) [Color figure can be viewed at wileyonlinelibrary.com]

custom-designed to navigate through the strawberry field and perform tasks such as detecting disease and collecting strawberries.

As discussed in Freese and Xu (2019) and Li et al., (2020), navigation in a typical strawberry field consist of two phases, over-bed and cross-bed. Controllers corresponding to these two phases serve distinctive purposes using feedback information from different sets of sensors.

Previously, a compound cross-bed transition procedure of multiple closed-loop and open-loop steps was developed (Li et al., 2020). Yet, the transition speed is mediocre and the performance can be easily affected by uncertainties in tire/terrain interaction such as tire sinkage, friction, compaction resistance, and bulldozing force (Liang et al., 2004). In this paper, we present the development and experiment verification of a new minimum-time cross-bed controller that is capable of adapting to the uncertainty in ground surface conditions based solely on vision feedback.

## 2.2 | Robot platform

The robot platform designed to scout strawberry fields is shown in Figure 2. A detailed discussion about this robot platform can be found in Menendez-Aponte et al. (2016). The robot is an all-wheel-drive, skid-steering vehicle, equipped with two electric motors to drive the wheels, a laptop computer running the navigation and control software, a front-facing camera for cross-bed navigation, and eight bottom-mounted ultrasonic sensors for over-bed navigation. Telescoping tubes are used in the robot frame so that the distance between the robot wheels can be adjusted to fit different strawberry field dimensions. Communications among the laptop computer, the ultrasonic sensors, and the powertrain system are hosted on an Arduino microcontroller board. The robot includes a leaf sampling subsystem formed by an $X$–$Y$–$Z$ table, a 3DOF manipulator, an end-effector, and halogen lights, which will not be discussed here. Arduino C codes are programmed for sensors and actuators aboard the robot. The navigation and control algorithms are programmed in MATLAB. The information is transmitted via serial communication between C and MATLAB software. It is worth mentioning that designs with differential drive system and castors wheels are popular in existing agricultural robots (Marchant et al., 1997). However, the drive system design of the robot is not the focus of this study, and the current robot in Figure 2 is used here to experimentally validate the effectiveness of the new vision-based minimum-time row transition controller.

Here we only list the important specifications or modifications that are related to the constraints considered in the new controller design. The detailed architecture of the navigation and control system is illustrated in Figure 3. The pose of the robot is estimated using the camera feedback in the cross-bed phase and the ultrasonic sensor readings in the over-bed phase. When the camera feedback is used at the cross-bed phase, the pose of the robot is estimated relative to the AprilTag (Olson, 2011) installed at the strawberry bed end. The identity number of each AprilTag is used to distinguish the otherwise indistinguishable strawberry beds in the field. At the over-bed phase, ultrasonic sensor readings are used to estimate the orientation of the robot with respect to the direction of the strawberry bed. The position of the robot along the strawberry bed is estimated based on the encoder feedback from the motor motion controller. The scheduler keeps track of the identified strawberry beds within the field of view (FOV), and determines which phase the robot is currently in based on the estimated robot pose, before activating the corresponding controller.

Some important robot specifications for the design and verification of the new navigation and control algorithm are summarized in Table 1. These parameters are either important factors of the robot performance or constraints affecting the controller design, as will be discussed in the next section. Note that the tire radius is measured different from the value listed in Menendez-Aponte et al. (2016). A wide FOV is necessary to avoid the scenario that the marker is outside of the camera view when the robot is moving to the marker while the robot is not turning towards it.

# 3 | MINIMUM-TIME ROW TRANSITION CONTROL PROBLEM

## 3.1 | Robot dynamics

As shown in Figures 4 and 5, the following five reference frames are used in the robot dynamic modeling and visual information processing. The first one is the local NED frame ($O^N$; $X^N$, $Y^N$, $Z^N$), of which the origin $O^N$ is located at the end of a strawberry bed. As the strawberry beds are in the north-south direction in the experimental field, the $X^N$ axis overlaps the centerline of a strawberry bed. The second frame is the robot frame ($O^R$; $X^R$, $Y^R$, $Z^R$), of which the origin $O^R$ is located at the center point of the robot top plate, and $X^R$, $Y^R$, and $Z^R$ point forward, rightward, and downward, respectively. The third frame is the camera frame ($O^C$; $X^C$, $Y^C$, $Z^C$). The center of

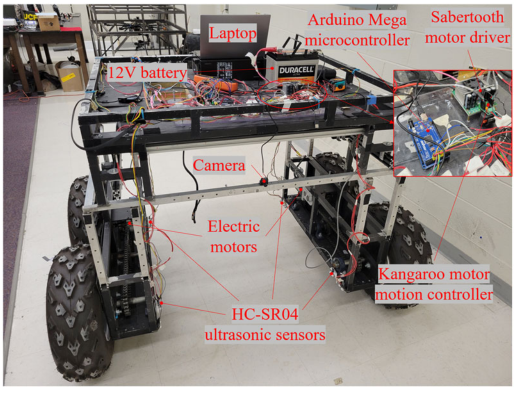


**FIGURE 2** The scouting robot platform [Color figure can be viewed at wileyonlinelibrary.com]

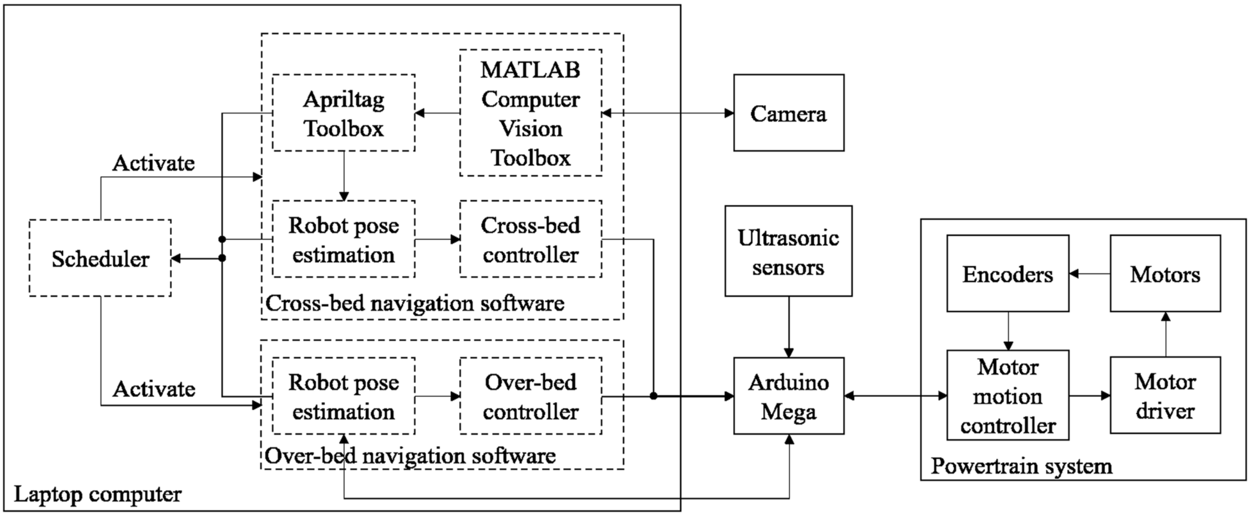


**FIGURE 3** Architecture of the navigation and control system. Solid boxes represent the hardware. Dashed boxes represent the software. Line arrows represent the directions of information flow

**TABLE 1** Important specifications of the robot

| Names | Values |
|---|---|
| Wheelbase | 0.65 m |
| Track width | 1.46 m |
| Maximum wheel angular velocity | 0.78 rad/s |
| Constrained wheel angular velocity | 0.3 rad/s |
| Tire radius | 0.27 m |
| Height of the robot | 1.17 m |
| Mounting height of the camera | 0.86 m |
| Tilt angle of the camera | 40° |
| Camera FOV | 91° × 51° |
| Camera resolution | 1280 × 720 |
| Laptop CPU frequency | 2.3 GHz |

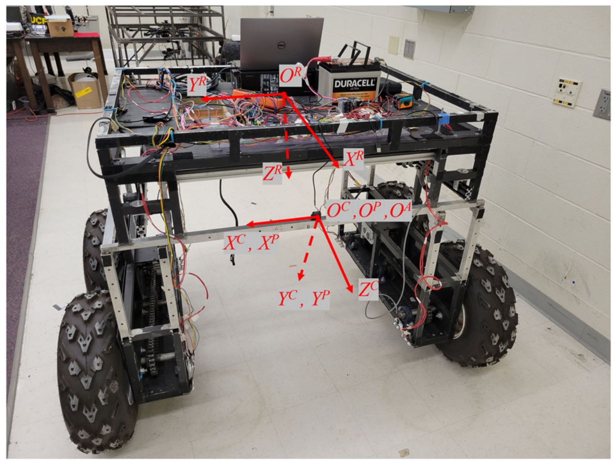


**FIGURE 5** Illustration of the robot, camera, pixel, and augmented pixel frames. The camera, pixel, and augmented pixel frames share the same origin and orientation but use different measurement units [Color figure can be viewed at wileyonlinelibrary.com]

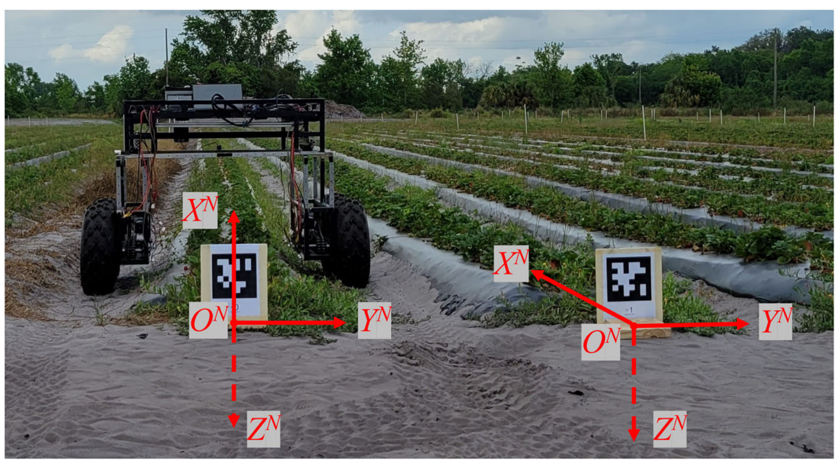


**FIGURE 4** Illustration of the local NED frames affixed to the strawberry beds. When the robot navigates with respect to a strawberry bed, its location is given in the corresponding local NED frame. NED, north-east-down [Color figure can be viewed at wileyonlinelibrary.com]

the projection is designated as the origin $O^C$. The $X^C$ and $Y^C$ axes point rightward and downward in the image plane, respectively, and the $Z^C$ axis overlaps the principal axis of the camera and points outward perpendicular to the image plane. The fourth coordinate frame is the pixel coordinate frame ($O^P$; $X^P$, $Y^P$) that has an origin at the principal point. The directions of $X^P$ and $Y^P$ are the same as those of $X^C$ and $Y^C$. The last one is the augmented pixel frame ($O^A$; $X^P$, $Y^P$, $Z^C$) that combines the pixel coordinates and the $Z^C$ axis of the camera frame. Let us use $x$, $y$, and $z$ to denote the coordinates along the $X$, $Y$, and $Z$ axes of a reference frame, respectively. The capitalized superscript refers to the coordinate frame. For example, $x^R$ denotes the $X^R$ coordinate of the robot frame.

The robot motion in the local NED frame is described by the following discrete-time model that is modified based on the kinematic model in Laumond et al. (1998)

$$\begin{bmatrix} \mathbf{x}^N_{r,t_{i+1}} \\ \psi_{r,t_{i+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^N_{r,t_i} \\ \psi_{r,t_i} \end{bmatrix} + Tr_w\eta \begin{bmatrix} \cos\psi_{r,t_i} & \cos\psi_{r,t_i} \\ \sin\psi_{r,t_i} & \sin\psi_{r,t_i} \\ 2\rho/W_r & -2\rho/W_r \end{bmatrix} \mathbf{u}_{t_i}, \tag{1}$$

where subscript $t_i$, $i = 0, 1, 2, \dots, f$, denotes the time steps. $\mathbf{x}^N_r = [x^N_r \;\; y^N_r]^T$ is the location of the robot in the NED coordinate frame. $\psi_r$ is the heading angle of the robot around the $Z^N$ axis, which is 0 when the robot is heading north. $T$ is the sampling time. $r_w$ is the radius of the robot wheels. Different from Laumond et al. (1998), two uncertainty parameters $0 < \eta < 1$ and $0 < \rho < 1$ are introduced here to account for the effect of tire/terrain interaction on the longitudinal and rotational motions of the robot, respectively. Additionally, these two uncertainties will count for the wear of the drivetrain subsystem. It is worth mentioning that in Kayacan et al. (2018) a similar set of traction parameters is included in a simple kinematic model to represent slip conditions. Here $\rho$ (called "ground condition factor") and $\eta$ (called "decomposition factor") are the uncertain parameters that will be adaptively estimated. $W_r$ is the track width of the robot, that is, the distance between the centerlines of its left and right wheels. The angular velocity of the wheels $\mathbf{u}_{t_i} = [\Omega_{L,t_i} \;\; \Omega_{R,t_i}]^T$ is the control input, with subscripts $L$ and $R$ denoting the left and right sides of the robot, respectively. For convenience, we define $\bar{\mathbf{x}}^N_{r,t_i} = [(\mathbf{x}^N_{r,t_i})^T \;\; \psi_{r,t_i}]^T$ and $\breve{\mathbf{x}}^N_{r,t_i} = [(\mathbf{x}^N_{r,t_i})^T \;\; z^N_r]^T$, where $z^N_r$ is the height of the robot that is known and assumed to be constant.

Denote the desired robot location-to-go at $t_i$ as $\mathbf{x}^N_{d,t_i} = [x^N_{d,t_i} \;\; y^N_{d,t_i}]^T$. Then based on Equation (1), the desired heading angle for the robot to reach $\psi_{d,t_i}$ is

$$\psi_{d,t_i} = tan^{-1}\left[\left(y^N_{d,t_i} - y^N_{r,t_i}\right)/\left(x^N_{d,t_i} - x^N_{r,t_i}\right)\right] \triangleq \Psi\left(\mathbf{x}^N_{d,t_i}, \mathbf{x}^N_{r,t_i}\right). \tag{2}$$

Define $\mathbf{x}_{e,t_i} = \mathbf{x}^N_{d,t_i} - \mathbf{x}^N_{r,t_i}$ and $\psi_{e,t_i} = \psi_{d,t_i} - \psi_{r,t_i}$. From the equalities in Equation (1), we have

$$\Omega_{L,t_i} + \Omega_{R,t_i} = \|\mathbf{x}_{e,t_i}\|_2/Tr_w\eta, \tag{3}$$

and

$$\Omega_{L,t_i} - \Omega_{R,t_i} = \psi_{e,t_i}W_r/(2Tr_w\eta\rho). \tag{4}$$

Following the inverse-dynamics (ID) policy idea (Forrest-Barlach & Babcock, 1987) and comparing Equations (3) and (4), the control needed to reach state $\mathbf{x}^N_{d,t_i}$ is

$$\mathbf{u}_{t_i} = \frac{1}{2Tr_w\eta}\begin{bmatrix} \|\mathbf{x}_{e,t_i}\|_2 + \frac{1}{2\rho}\psi_{e,t_i}W_r \\ \|\mathbf{x}_{e,t_i}\|_2 - \frac{1}{2\rho}\psi_{e,t_i}W_r \end{bmatrix} \triangleq \pi(\bar{\mathbf{x}}_{r,t_i}, \bar{\mathbf{x}}_{d,t_i}). \tag{5}$$

*Remark* 1. The control commands generated via Equation (5) are direct motor commands. When saturated individually, the difference between the left and right motor commands may be truncated, causing the robot to understeer and rendering it unable to reach $\psi_{d,t_i}$. Consequently, $\mathbf{x}^N_{d,t_i}$ will not be reached. Thus, it is important to saturate the control commands coordinately to reserve the difference in motor speeds

## 3.2 | Minimum-time row transition control problem

The minimum-time cross-bed alignment can be formulated as an optimal control problem minimizing the following performance index

$$J = t_f - t_0, \tag{6}$$

subject to the robot dynamics Equation (1) and the initial conditions of

$$\mathbf{x}^N_{r,t_0} = \mathbf{x}^N_{r,0}, \psi_{r,t_0} = \psi_{r,0}. \tag{7}$$

The robot is aligned with the target strawberry bed if and only if

$$y^N_{r,t_i} = 0, \;\; \psi_{r,t_i} = 0, \tag{8}$$

which defines the terminal constraints.

Limited by the vertical FOV of the onboard camera, the visual marker will inevitably enter the blind zone of the front camera as the robot drives towards the target bed. Meanwhile, the ultrasonic sensors will be inactive before the robot enters that bed. Thus, there will be a session connecting the cross-bed motion and the over-bed motion driven by an open-loop controller. To ensure a successful transfer for this session, the terminal constraints in Equation (8) is adjusted so that the robot stops at a safety distance $d_s$ away from the strawberry bed as follows:

$$\mathbf{x}^N_{r,t_f} = \begin{bmatrix} -d_s & 0 \end{bmatrix}^T, \psi_{r,t_f} = 0. \tag{9}$$

Additionally, the area that is available for the cross-bed trajectory planning needs to be constrained so that the robot will not bump into neighboring strawberry beds. Therefore, the following constraints

$$x^N_{r,t_i} \le -d_s, \tag{10}$$

and

$$-d_b \le y^N_{r,t_i} \le d_b \tag{11}$$

need to be considered, where $d_b$ is the distance between the neighboring strawberry beds.

Besides the above constraints relating to the cross-bed motion task, the robot design also imposes some constraints on the row transition problem. To start with, the camera used for navigation has a limited horizontal FOV. To localize the robot based on visual feedback, the visual marker needs to remain inside the camera FOV. This limitation imposes an inequality constraint on the heading angle of the robot:

$$\|\psi_{b,t_i} - \psi_{r,t_i}\| < \zeta/2, \tag{12}$$

where $\psi_{b,t_i} = \arctan(y^N_{r,t_i}/x^N_{r,t_i})$ is the bearing angle of the robot in the local NED frame. $\zeta$ is the horizontal FOV angle of the camera. Since $\psi_{b,t_i}$ is a function of $\mathbf{x}^N_{r,t_i}$, this constraint is time-varying.

Furthermore, because the maximum torque output from an electric motor decreases as it reaches the top of its revolution curve, it is necessary to impose a control constraint on the wheel speed to secure an ample torque output for the smooth operation of the robot. Therefore, the following control constraint needs to be satisfied

$$\|\Omega_{L,t_i}\| \le \Omega_{\max}, \|\Omega_{R,t_i}\| \le \Omega_{\max}. \tag{13}$$

In summary, the studied algorithm will find the optimal commands $\mathbf{u}^*_{t_i}, t_i = t_0, \ldots, t_f$ to minimize the performance index Equation (6), subject to the equality constraints of Equations (7)–(9) and the inequality constraints of Equations (10)–(13).

## 4 | VISION-BASED ROW ALIGNMENT CONDITIONS

### 4.1 | Coordinate transformations

To solve the minimum time cross-bed motion control problem defined in Section 3.2 using visual feedback information, we need to establish the coordinate transformations among those related coordinate frames.

Let $Q_m$ be the center point of the visual marker, and the projected location of $Q_m$ in the camera frame is (Zhang, 2000)

$$\mathbf{x}^C_m = z^C_m \mathbf{M}^{-1}\mathbf{p}, \tag{14}$$

where $\mathbf{x}^C_m = [x^C_m \;\; y^C_m \;\; z^C_m]^T$ is the coordinate vector of $Q_m$ in the camera frame. $\mathbf{p} = [x^P_m \;\; y^P_m \;\; 1]^T$ is the coordinate vector of $Q_m$ in the pixel frame. $\mathbf{M}$ is the intrinsic matrix of the camera as $\mathbf{M} = \mathrm{diag}\{f_x, f_y, 1\}$, where $f_x$ and $f_y$ are the pixel focal lengths of the camera.

The location of $Q_m$ in the robot frame via the coordinate transformation is given by

$$\mathbf{x}^R_m = \mathbf{x}^R_c + \mathbf{R}^{RC}\mathbf{x}^C_m, \tag{15}$$

where $\mathbf{x}^R_c = [x^R_c \;\; y^R_c \;\; z^R_c]^T$ is the position vector of the camera in the robot frame. $\mathbf{R}^{RC}$ is the coordinate transformation matrix from the camera frame to the robot frame as

$$\mathbf{R}^{RC} = \begin{bmatrix} 0 & \sin\theta_c & \cos\theta_c \\ 1 & 0 & 0 \\ 0 & \cos\theta_c & -\sin\theta_c \end{bmatrix}, \tag{16}$$

where $\theta_c < 0$ is the pitching angle of the camera around the $Y^R$ axis.

The robot location in the local NED frame then can be expressed as follows:

$$\breve{\mathbf{x}}^N_r = \mathbf{x}^N_m - \mathbf{R}^{NR}\mathbf{x}^R_m, \tag{17}$$

where $\mathbf{x}^N_m = [0 \;\; 0 \;\; z^N_m]^T$ is the coordinate vector of the marker in the local NED frame. $\mathbf{R}^{NR}$ is the coordinate transformation matrix from the robot frame to the local NED frame as

$$\mathbf{R}^{NR} = \begin{bmatrix} \cos\psi_r & -\sin\psi_r & 0 \\ \sin\psi_r & \cos\psi_r & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{18}$$

Substituting Equations (14) and (15) into Equation (17) yields the following expression of the robot location

$$\breve{\mathbf{x}}^N_r = \mathbf{x}^N_m - \mathbf{R}^{NR}\mathbf{x}^R_c - z^C_m\mathbf{R}^{NC}\mathbf{M}^{-1}\mathbf{p}, \tag{19}$$

where

$$\mathbf{R}^{NC} = \mathbf{R}^{NR}\mathbf{R}^{RC} = \begin{bmatrix} -\sin\psi_r & \cos\psi_r\sin\theta_c & \cos\psi_r\cos\theta_c \\ \cos\psi_r & \sin\psi_r\sin\theta_c & \sin\psi_r\cos\theta_c \\ 0 & \cos\theta_c & -\sin\theta_c \end{bmatrix}. \tag{20}$$

Note that in Equation (19), $\mathbf{p}$ can be directly obtained from the vision feedback, while $z^C_m$ and $\psi_r$ can be measured and then calculated by using the AprilTag imaging processing toolbox (Olson, 2011). Therefore, the full state feedback can be provided for Equation (1).

The measurement model is obtained by inversing Equation (19) as

$$\mathbf{p} = \mathbf{M}(\mathbf{R}^{NC})^{-1}\left(\mathbf{x}_m^N - \breve{\mathbf{x}}_r^N - \mathbf{R}^{NR}\mathbf{x}_c^R\right)\Big/ z_m^C. \tag{21}$$

Noticing that $(\mathbf{R}^{NC})^{-1} = \mathbf{R}^{CN}$ and $\mathbf{R}^{CN}\mathbf{R}^{NR} = \mathbf{R}^{CR}$, Equation (21) can be expanded as

$$\mathbf{p} = \mathbf{M}\left[\mathbf{R}^{CN}\left(\mathbf{x}_m^N - \breve{\mathbf{x}}_r^N\right) - \mathbf{R}^{CR}\mathbf{x}_c^R\right]\Big/ z_m^C$$
$$= \begin{bmatrix} -\dfrac{f_x\left(y_r^N \cos\psi_r - x_r^N \sin\psi_r\right)}{z_m^C} \\ -\dfrac{f_y\left[x_c^R \sin\theta_c - \left(z_m^N - z_r^N\right)\cos\theta_c + x_r^N \sin\theta_c \cos\psi_r + y_r^N \sin\theta_c \sin\psi_r\right]}{z_m^C} \\ -\dfrac{x_c^R \cos\theta_c + \left(z_m^N - z_r^N\right)\sin\theta_c + x_r^N \cos\theta_c \cos\psi_r + y_r^N \cos\theta_c \sin\psi_r}{z_m^C} \end{bmatrix}. \tag{22}$$

## 4.2 | Necessary and sufficient conditions of row alignment

Let $\mathbf{p}_a = [x_m^P \;\; y_m^P \;\; z_m^P]^T$ be the coordinates of the marker center $Q_m$ in the augmented pixel frame. For the robot to be aligned with the visual marker, the following necessary and sufficient conditions exist.

**Lemma 1** (Necessary condition). *If the robot is aligned with the target row, the augmented pixel coordinates of* $\mathrm{Q_m}$ *satisfies the following conditions:*

$$\mathbf{p}_a = \begin{bmatrix} 0 \\ f_y\left\{\tan\theta_c - \dfrac{\left(z_m^N - z_r^N\right)\sec^2\theta_c}{\left(x_c^R + x_r^N\right) + \left(z_m^N - z_r^N\right)\tan\theta_c}\right\} \\ -\left(x_c^R + x_r^N\right)\cos\theta_c - \left(z_m^N - z_r^N\right)\sin\theta_c \end{bmatrix}, x_r^N \le -d_s. \tag{23}$$

*Proof.* Substituting the alignment condition Equation (8) into Equation (22), we have

$$\mathbf{p} = \begin{bmatrix} 0 \\ -\frac{f_y}{z_c^m}\left[\left(x_c^R + x_r^N\right)\sin\theta_c - \left(z_m^N - z_r^N\right)\cos\theta_c\right] \\ -\frac{1}{z_m^C}\left[\left(x_c^R + x_r^N\right)\cos\theta_c + \left(z_m^N - z_r^N\right)\sin\theta_c\right] \end{bmatrix}, x_r^N \le -d_s. \tag{24}$$

Since $\mathbf{p} = [x_m^P \;\; y_m^P \;\; 1]^T$, it is straightforward to get the first and third equalities of Equation (23) from Equation (24). Substituting the third equality of Equation (23) into the second equality of Equation (24), we have the following necessary condition on $y_m^P$ for alignment

$$y_m^P = f_y \frac{\left(x_c^R + x_r^N\right)\sin\theta_c - \left(z_m^N - z_r^N\right)\cos\theta_c}{\left(x_c^R + x_r^N\right)\cos\theta_c + \left(z_m^N - z_r^N\right)\sin\theta_c}, x_r^N \le -d_s, \tag{25}$$

which can be further simplified into the second equality of Equation (23). Therefore, Equation (23) is the necessary condition of a perfect row alignment. □

**Lemma 2** (Sufficient condition). *If the augmented pixel coordinates of* $\mathrm{Q_m}$ *satisfies Equation (23), the robot perfectly aligns with the target row.*

*Proof.* From the proof of Lemma 1, we know that Equations (23) and (24) are equivalent. Because $f_x$, $f_y$, and $z_m^C$ are nonzero, from the first equalities of Equations (22) and (24), we have

$$y_r^N \cos\psi_r - x_r^N \sin\psi_r = 0, \tag{26}$$

or

$$x_r^N \sin\psi_r = y_r^N \cos\psi_r. \tag{27}$$

From the second and third equalities of Equations (22) and (24), we have

$$x_r^N \sin\theta_c \cos\psi_r + y_r^N \sin\theta_c \sin\psi_r = x_r^N \sin\theta_c, \tag{28}$$

and

$$x_r^N \cos\theta_c \cos\psi_r + y_r^N \cos\theta_c \sin\psi_r = x_r^N \cos\theta_c. \tag{29}$$

Because $\sin\theta_c$ and $\cos\theta_c$ cannot simultaneously be zero, from Equations (28) and (29), we get

$$x_r^N \cos\psi_r = x_r^N - y_r^N \sin\psi_r. \tag{30}$$

Summing up the squares of both sides of Equations (27) and (30) correspondingly, we have

$$\begin{aligned}\left(x_r^N\right)^2 = \left(y_{r,t_{i+1}}^N\right)^2 \cos^2\psi_r + \left(x_r^N\right)^2 + \left(y_r^N\right)^2 \sin^2\psi_r \\ - 2x_r^N y_r^N \sin\psi_r,\end{aligned} \tag{31}$$

or

$$y_r^N\left(y_r^N - 2x_r^N \sin\psi_r\right) = 0, \tag{32}$$

which indicates $y_r^N = 0$ or

$$y_r^N = 2x_r^N \sin\psi_r. \tag{33}$$

If Equation (33) holds true, by substituting Equation (27) into Equation (33), we obtain

$$y_r^N(1 - 2\cos\psi_r) = 0, \tag{34}$$

which again indicates $y_r^N = 0$ or

$$\cos\psi^r = 1/2. \tag{35}$$

If Equation (35) is true, by substituting it back to Equations (27) and (30),

$$y_r^N = 2x_r^N \sin\psi_r, \tag{36}$$

and

$$x_r^N = 2y_r^N \sin\psi_r, \tag{37}$$

are derived.

Substituting Equation (37) into Equation (36) yields

$$y_r^N(4\sin^2\psi_r - 1) = 0. \tag{38}$$

Since $\sin^2\psi_r = 1/4$ and Equation (35) conflict, the only solution is $y_r^N = 0$.

Because $x_r^N \le -d_s$, the normal range of $\psi_r$ is $[-90°, 90°]$. Substituting $y_r^N = 0$ back into Equation (27) yields $\psi_r = 0$. Therefore, Equation (23) is the sufficient condition of a perfect alignment. □

## 5 | ROW TRANSITION CONTROL

### 5.1 | Row alignment reference trajectory

In this section, a reference trajectory following the virtual motion camouflage (VMC) rule (Srinivasan & Davey, 1995; Xu & Li, 2014) is designed to guide the system state towards the desired terminal conditions.

Set $\mathcal{S}^3_{\mathbf{p}_a}$ includes all $\mathbf{p}_a$ that satisfy the row alignment conditions in Equation (23). Notice that $x_r^N$ is the only independent variable in Equation (23). Both $y_m^P$ and $z_m^C$ depend on $x_r^N$. However, because $y_m^P$ is nonlinear in $x_r^N$, the VMC rule cannot be directly applied to $\mathbf{p}_a \in \mathcal{S}^3_{\mathbf{p}_a}$. To overcome this, a new variable

$$w \triangleq \frac{f_y}{y_m^P - f_y\tan\theta_c} = -\frac{\left(x_c^R + x_r^N\right) + \left(z_m^N - z_r^N\right)\tan\theta_c}{\left(z_m^N - z_r^N\right)\sec^2\theta_c},\ x_r^N \le -d_s, \tag{39}$$

which is linear to $x_r^N$, is defined. Note that in Equation (39), $z_m^N \ne z_r^N$ is guaranteed by the fact that the robot is taller than the visual marker. Define $\mathbf{w} \triangleq \begin{bmatrix}0 & w & z_m^C\end{bmatrix}^T$, then we have

$$\mathbf{w} = \begin{bmatrix} 0 \\ -\frac{\left(x_c^R + x_r^N\right) + \left(z_m^N - z_r^N\right)\tan\theta_c}{\left(z_m^N - z_r^N\right)\sec^2\theta_c} \\ -\left(x_c^R + x_r^N\right)\cos\theta_c - \left(z_m^N - z_r^N\right)\sin\theta_c \end{bmatrix}. \tag{40}$$

*Remark* 2. The second equality of Equation (40) holds when $\mathbf{p}_a$ satisfies the alignment conditions in Equation (23). Thus, $\mathbf{w}$ also satisfies the alignment conditions.

Let $\mathbf{w}_d = \mathbf{w}|_{x_r^N=-d_s}$ and $\mathbf{w}_{ref,t_i} = \mathbf{w}|_{x_r^N = x_{r,t_i}^N}$ be the two endpoints of a $\mathbf{w}$ trajectory that satisfies the alignment conditions. We denote this $\mathbf{w}$ trajectory as $\mathcal{S}^3_{\mathbf{w}}$. Following the VMC rule (Li & Xu, 2020), any $\mathbf{w}_{t_k} \in \mathcal{S}^3_{\mathbf{w}}$ can be expressed as follows:

$$\mathbf{w}_{t_k} = \mathbf{w}_{ref,t_i} + \upsilon_{t_k}(\mathbf{w}_d - \mathbf{w}_{ref,t_i}),\ \upsilon_{t_k} \in \mathcal{V}^1, k \in [i, f\,], \tag{41}$$

where $\upsilon_{t_k} \in \mathcal{V}^1$ is the path-control-parameter (PCP) (Li & Xu, 2020; Xu & Li, 2014) in the 1D PCP space $\mathcal{V}^1$.

*Remark* 3. The projection of $\mathcal{S}^3_{\mathbf{w}}$ in the $x$–$y$ plane of the local NED coordinates is the straight line that connects $\mathbf{x}_{r,t_f}^N$ and $\mathbf{x}_{ref,t_i}^N = [x_{ref,t_i}^N \quad 0]^T$. As we have proven in Section 4.2, for alignment conditions to be established, $\psi_r = 0$. Therefore, $\mathbf{w}_{t_k}$ corresponds to a robot state $\bar{\mathbf{x}}_{r,t_k} = [x_{r,t_k}^N \quad 0 \quad 0]^T$ that satisfies the terminal condition in Equation (8).

**Definition 1.** The projection of $\mathcal{S}^3_{\mathbf{w}}$ in the $\bar{\mathbf{x}}_r$ space is a state trajectory that ensures the robot to be aligned with the visual marker. We name this state trajectory as the alignment reference trajectory, which is denoted as $\mathcal{S}^3_{l,t_i}$.

In the cross-bed alignment scenario, $\mathcal{V}^1 = [0, 1]$ can ensure there is no overshoot. Similar to Li and Xu (2020), the inverse mapping of Equation (41) is given by

$$\upsilon_{t_k} = \left\langle \mathbf{w}_{t_k} - \mathbf{w}_{ref,t_i}, \mathbf{w}_d - \mathbf{w}_{ref,t_i} \right\rangle / \|\mathbf{w}_d - \mathbf{w}_{ref,t_i}\|_2, \tag{42}$$

Since $d_s$ is fixed, $\mathbf{w}_d$ can be calculated beforehand. However, $\mathbf{w}_{ref,t_i}$ changes corresponding to $x_{r,t_i}^N$, which is calculated based on the vision feedback at every time step.

### 5.2 | Minimum-time row transition controller

From the bijection defined by Equations (41) and (42), reaching $\mathbf{w}_d$ is equivalent to reaching $\upsilon_d$. Given $\upsilon_{t_k}$, the corresponding $\mathbf{w}_{t_k}$ can be calculated using Equation (41). Then $y_{m,t_k}^P$ can be calculated by inverting Equation (39) as

$$y_{m,t_k}^P = f_y(1 + w_{t_k}\tan\theta_c)/w_{t_k}. \tag{43}$$

Since $\mathbf{p}_{a,t_k}$ and $\mathbf{w}_{t_k}$ share the same third element, $\mathbf{p}_{a,t_k}$ is obtained. From Equation (19), $\breve{\mathbf{x}}_{r,t_k}^N \in \mathcal{S}^3_{l,t_i}$ can then be calculated as

$$\breve{\mathbf{x}}_{r,t_k}^N = \mathbf{x}_m^N - \mathbf{R}^{NR}\mathbf{x}_c^R - z_{m,t_k}^C \mathbf{R}^{NC}\mathbf{M}^{-1}\mathbf{p}_{t_k}. \tag{44}$$

*Remark* 4. Since $\mathbf{p}_{a,t_k} \in \mathcal{S}^3_{\mathbf{p}_a,t_i}$ satisfies the visual alignment condition Equation (23), which is a projection of Equation (8) in the vision system, $\breve{\mathbf{x}}_{r,t_k}^N \in \mathcal{S}^3_{l,t_i}$ satisfies Equation (8). For $\forall\, \breve{\mathbf{x}}_{r,t_k}^N \in \mathcal{S}^3_{l,t_i}$, the robot's yaw angle $\psi_{r,t_k} = 0$. Therefore, from Equations (18) and (20), $\mathbf{R}^{NR}$ and $\mathbf{R}^{NC}$ are constant in Equation (44).

For brevity, we define the above procedure of calculating $\breve{\mathbf{x}}^{N}_{r,t_k} \in \mathcal{S}^3_{l,t_i}$ using $\upsilon_{t_k} \in \mathcal{V}^1$ as a mapping $\mathbf{h} : \mathcal{V}^1 \to \mathcal{S}^3_{l,t_i}$. Then, for $\forall\, \upsilon_{t_k} \in \mathcal{V}^1$, $\breve{\mathbf{x}}^{N}_{r,t_k} = \mathbf{h}(\upsilon_{t_k})$ is used to find the control input $\mathbf{u}_{t_k} = \pi(\bar{\mathbf{x}}_{r,t_k}, \bar{\mathbf{x}}_{d,t_f})$ via the ID approach discussed in Section 3.1 to reach $\mathbf{x}^N_{d,t_f}$. Applying $\mathbf{u}_{t_k}$ to the dynamic model in Equation (1), a prediction of $\mathbf{x}^N_{r,t_{k+1}}$ can be made and subsequently used to find the corresponding $\upsilon_{t_{k+1}} = \mathbf{h}^{-1}(\mathbf{x}^N_{r,t_{k+1}})$.

Customized from the ID-VMC based value iteration (ID-VMC-VI) algorithm (Li & Xu, 2020), the original problem Equation (6) can be solved over the VMC subspace using the Bellman equation (Howard, 1960) of

$$V_j(\upsilon_{t_{k+1}}) = L(\upsilon_{t_k}) + \gamma V_{j-1}(\upsilon_{t_k}), \forall\, \upsilon_{t_k} \in \mathcal{V}^1, \tag{45}$$

where $\gamma$ is the discount factor. $V_j$ denotes the value function of the $j$th iteration. $L(\upsilon)$ is the step cost and is defined as

$$L(\upsilon) = \begin{cases} T, & \text{when constraints Equations(10)-(13)} \\ & \quad \text{are satisfied.} \\ \text{penalty}, & \text{otherwise.} \end{cases} \tag{46}$$

The above value iteration will stop once the following criterion (Howard, 1960) is met

$$\max_{\upsilon \in \mathcal{V}^1} \|V_j(\upsilon) - V_{j-1}(\upsilon)\| \le \varepsilon, \tag{47}$$

at which point, $V_j(\upsilon)$ is reckoned to have converged to the optimal value function $V^*(\upsilon)$ for $\forall\, \upsilon \in \mathcal{V}^1$.

Here, a multistep lookahead, similar to Bertsekas (2013, 2020) and Sutton (1988), will be taken from the current robot state $\bar{\mathbf{x}}_{r,t_i}$ towards $\bar{\mathbf{x}}_{d,t_i}(\upsilon) \triangleq [(\mathbf{h}(\upsilon))^T \quad \Psi(\mathbf{h}(\upsilon), \mathbf{x}^N_{r,t_i})]^T$, for $\forall\, \upsilon \in \mathcal{V}^1$. The optimal PCP-to-go from $\bar{\mathbf{x}}_{r,t_i}$ is then found by searching $\mathcal{V}^1$ via

$$\upsilon^* = \arg\min_{\upsilon \in \mathcal{V}^1} \left\{ \sum_{p=0}^{l-1} \gamma^p L(\upsilon_{t_{i+p}}) + \gamma^l V^*(\upsilon_{t_{i+l}}) \right\}, \tag{48}$$

where $l$ is the number of steps it takes to reach $\bar{\mathbf{x}}_{d,t_i}(\upsilon)$ from $\bar{\mathbf{x}}_{r,t_i}$, which is not a determined value.

*Remark* 5. The ID policy, shown in Equation (5), tries to drive the system state from $\bar{\mathbf{x}}_{r,t_i}$ to $\bar{\mathbf{x}}_{d,t_i}$ in one time step. However, because of the control constraints, this may not be attainable. Therefore, the output of the ID policy is saturated according to the control constraints, and a multistep lookahead (Bertsekas, 2013, 2020; Sutton, 1988) is used in Equation (48) to ensure that the control constraints are satisfied when searching for the optimal PCP-to-go.

*Remark* 6. As we discussed in Remark 3, there $\exists\, \bar{\mathbf{x}} \in \mathcal{S}^3_{l,t_i}$ for $\forall\, \upsilon \in \mathcal{V}^1$. Therefore, Equation (48) essentially searches all possible trajectories from $\bar{\mathbf{x}}_{r,t_i}$ to $\mathcal{S}^3_{l,t_i}$ that follow the ID policy and satisfy the control constraints. $\upsilon^*$ corresponds to the optimal state-to-go on the alignment reference trajectory, which is $\bar{\mathbf{x}}_{d,t_i}(\upsilon^*)$. In comparison to Xu and Li (2014), where the search is limited to the straight line connecting $\bar{\mathbf{x}}_{r,t_i}$ and $\bar{\mathbf{x}}_{r,t_f}$, the search strategy presented in this paper covers a much larger search space and thus delivers improved solution optimality.

After the optimal PCP-to-go is found, the optimal control to take at $t_i$ is then calculated via ID Equation (5) as

$$\mathbf{u}^*_{t_i} = \pi(\bar{\mathbf{x}}_{r,t_i}, \mathbf{h}(\upsilon^*)). \tag{49}$$

The above procedure will be repeated at every time step until $\bar{\mathbf{x}}_{d,t_f}$ is reached.

*Remark* 7. Once $\bar{\mathbf{x}}_{r,t_i}$ reaches $\mathcal{S}^3_{l,t_i}$ following Equation (49), the alignment conditions are satisfied. The robot will then follow the alignment reference trajectory until it reaches $\mathbf{x}_{r,t_f}$. The row transition is finished at that point.

### 5.3 | Ground condition estimation via vision feedback

Deviations between the visual feedback and the predicted visual information are used to update the ID policy. Let $\boldsymbol{\chi} = [\eta \quad \rho]^T$. $\hat{\boldsymbol{\chi}}$ denotes the estimate of $\boldsymbol{\chi}$. $\hat{\mathbf{p}}_a$ and $\hat{\psi}_r$ represent the predictions of $\mathbf{p}_a$ and $\psi_r$ based on $\hat{\boldsymbol{\chi}}$, respectively. $\hat{\boldsymbol{\chi}}$ will be updated so that $\hat{\mathbf{p}}_a$ and $\hat{\psi}_r$ will approach $\mathbf{p}_a$ and $\psi_r$. This is equivalent to solving the following optimization problem (Curry, 1944)

$$\hat{\boldsymbol{\chi}} = \arg\min_{\hat{\boldsymbol{\chi}}} e, \tag{50}$$

where $e = \left(\tilde{\mathbf{p}}_a{}^T \tilde{\mathbf{p}}_a + \tilde{\psi}_r{}^2\right) \big/ 2$, $\tilde{\mathbf{p}}_a = \mathbf{p}_a - \hat{\mathbf{p}}_a$, and $\tilde{\psi}_r = \psi_r - \hat{\psi}_r$.

By solving Equation (50), seeing Appendix A, the respective update rules for $\hat{\eta}$ and $\hat{\rho}$ are

$$\begin{gathered} \hat{\eta}_{t_{i+1}} = \hat{\eta}_{t_i} + \alpha_1 \left( \left\| \mathbf{x}_{e,t_i} \right\|_2 \mathbf{a}^T_{3\times1} \tilde{\mathbf{p}}_{a,t_{i+1}} + \alpha_2 \psi_{e,t_i} \tilde{\psi}_{r,t_{i+1}} \right), \\ \hat{\rho}_{t_{i+1}} = \hat{\rho}_{t_i} + \alpha_3 \psi_{e,t_i} \tilde{\psi}_{r,t_{i+1}} \end{gathered} \tag{51}$$

where

$$\mathbf{a}_{3\times1} = \begin{bmatrix} \dfrac{ \begin{gathered} f_x \left\{ \left[ \left( z^N_m - z^N_r \right) \sin\theta_c + x^R_c \cos\theta_c \right] \sin\left( \psi_{r,t_i} - \psi_{r,t_{i+1}} \right) + \left( x^N_{r,t_{i+1}} \sin\psi_{r,t_i} \right.\right. \\ \left.\left. - \, y^N_{r,t_{i+1}} \cos\psi_{r,t_i} \right) \cos\theta_c \right\} \end{gathered} }{ \left[ \left( x^R_c + x^N_{r,t_{i+1}} \cos\psi_{r,t_{i+1}} + y^N_{r,t_{i+1}} \sin\psi_{r,t_{i+1}} \right) \cos\theta_c + \left( z^N_m - z^N_{r,t_{i+1}} \right) \sin\theta_c \right]^2 } \\ \dfrac{ f_y \left( z^N_m - z^N_r \right) \cos\left( \psi_{r,t_i} - \psi_{r,t_{i+1}} \right) }{ \left[ \left( x^R_c + x^N_{r,t_{i+1}} \cos\psi_{r,t_{i+1}} + y^N_{r,t_{i+1}} \sin\psi_{r,t_{i+1}} \right) \cos\theta_c + \left( z^N_m - z^N_{r,t_{i+1}} \right) \sin\theta_c \right]^2 } \\ -\cos\theta_c \cos\left( \psi_{r,t_i} - \psi_{r,t_{i+1}} \right) \end{bmatrix}. \tag{52}$$

The convergence analysis follows a similar approach in (Curry, 1944), thus omitted here. It is worth noting that when tuning Equation (51), $\alpha_2$ should be kept close to 1 to maintain the balance between the effects of $\tilde{\mathbf{p}}_a$ and $\tilde{\psi}_r$ on $\hat{\eta}$, respectively. $\alpha_1$ and $\alpha_3$ only need to be tuned once when the robot is working on a different terrain condition to achieve a balanced convergence speed and a proper estimation of uncertain parameters.

### 5.4 | Row transition algorithm

Table 2 summarizes the proposed algorithm, significantly revised based on ID-VMC-VI (Li & Xu, 2020), to solve the discrete-time constrained optimal control problem defined in Section 3. The main structure of the algorithm is the same as any ADP approaches (Bertsekas, 2013, 2020; Sutton, 1988; Wei et al., 2017). The state search space is reduced to the PCP search space in Line 7. The control policy is listed in Line 9. The projections among the augmented pixel, pixel, and robot NED coordinates are in Line 18. The unknown parameters are updated in Line 20 and Line 21. The algorithm executes at every time step $t_i$ during the alignment with the target row. The algorithm runs in real-time, which is a significant advantage over many other ADP-based optimal controls (Ni et al., 2015; Nosair et al., 2010; Zhou et al., 2018).

*Remark* 8. Since the computational cost of the algorithm is low, it is executed at each time step in the framework of model predictive control (MPC). The stability of the proposed algorithm can be easily proven following similar approaches as in many MPC-related studies (Mayne & Raković, 2003; Mayne et al., 2000).

*Remark* 9. Since the proposed algorithm is executed in a MPC framework as discussed in Remark 8, the control commands along the time horizon of $[t_i, t_{i+1}]$ are calculated using the real-time feedback information obtained at the beginning of this horizon. Therefore, although the proposed controller is not a robust controller, it can reject noise and mismatches to a certain degree.

In addition to the CPU time advantage, because the search space dimension is significantly reduced to 1D, the memory required to store the data is dramatically lowered. This feature will benefit field robots having only a limited RAM capacity.

## 6 | SIMULATION RESULTS AND DISCUSSION

### 6.1 | Simulation settings

The proposed row transition controller is first tested in a simulation environment before field experiments. The MATLAB simulation, illustrated in Figure 6, consists of the navigation software and the simulated environment model. The cross-bed navigation software includes the proposed control algorithm, the ID policy improvement algorithm, and the necessary coordinate transformations for the robot pose estimation. The environment model is comprised of the robot dynamic model, the visual marker model, and the wheel-terrain interaction parameters. Here, the uncertainty parameters in the tire/terrain interaction are adaptively adjusted, whereas the simulation robot model uses the assumed actual values. The assumed actual $\eta$ is 0.9 and $\rho$ is 0.85. The PCP space $\mathcal{V}^1$ is discretized into 70 nodes. The simulation time step size is 0.3 s. The step sizes of the ID policy improvement are chosen to be $\alpha_1 = 4 \times 10^{-4}$, $\alpha_2 = 1$, and $\alpha_3 = 0.8$. The simulation environment also uses the robot specifications from Table 1 and the actual test field dimensions as listed in Table 3.

### 6.2 | Simulation results and analyses

The following contains the results from the proposed algorithm and a benchmark dynamic programming (DP) algorithm for comparison.

**TABLE 2** Custom-Designed ID-VMC-VI Algorithm at $t_i$

| | |
|---|---|
| 1: | Given $\mathcal{V}^1$, stopping criterion $\varepsilon$, and update step sizes $\alpha_1$, $\alpha_2$, $\alpha_3$. |
| 2: | $\mathbf{w}_d \leftarrow \mathbf{w}\big\|_{x_r^N=-d_s}$, $\mathbf{w}_{ref,t_i} \leftarrow \mathbf{w}\big\|_{x_r^N=x_{r,t_i}^N}$ |
| 3: | $j = 0$ |
| 4: | $V_j(\upsilon) \leftarrow 0, \forall\, \upsilon \in \mathcal{V}^1$. |
| 5: | **while** $\max\limits_{\upsilon\in\mathcal{V}^1} \|V_j(\upsilon) - V_{j-1}(\upsilon)\| > \varepsilon$ |
| 6: | $j \leftarrow j + 1$ |
| 7: | **for** $\forall\, \upsilon_{t_k} \in \mathcal{V}^1 \backslash \{\upsilon_d\}$ |
| 8: | $\breve{\mathbf{x}}^N_{r,t_k} \leftarrow \mathbf{h}(\upsilon_{t_k})$ |
| 9: | $\mathbf{u}_{t_k} \leftarrow \pi(\bar{\mathbf{x}}_{r,t_k}, \bar{\mathbf{x}}_{d,t_f})$ |
| 10: | Propagate $\bar{\mathbf{x}}_{r,t_k}$ to $\bar{\mathbf{x}}_{r,t_{k+1}}$ by substituting $\mathbf{u}_{t_k}$ into Equation (A4). |
| 11: | $\upsilon_{t_{k+1}} \leftarrow \mathbf{h}^{-1}(\mathbf{x}^N_{r,t_{i+1}})$ |
| 12: | $V_j(\upsilon_{t_{k+1}}) \leftarrow L(\upsilon_{t_k}) + \gamma V_{j-1}(\upsilon_{t_k})$ |
| 13: | **end** |
| 14: | **end** |
| 15: | $V^*(\upsilon) \leftarrow V(\upsilon), \forall\, \upsilon \in \mathcal{V}^1$ |
| 16: | $\upsilon^* \leftarrow \arg\min\limits_{\upsilon\in\mathcal{V}^1} \left\{\sum_{p=0}^{l-1} \gamma^p L(\upsilon_{t_{i+p}}) + \gamma^l V^*(\upsilon_{t_{i+l}})\right\}$ |
| 17: | $\mathbf{u}^*_{t_i} \leftarrow \pi(\bar{\mathbf{x}}_{r,t_i}, \mathbf{h}(\upsilon^*))$ |
| 18: | Update $\mathbf{p}_{a,t_{i+1}}$ and $\bar{\mathbf{x}}_{r,t_{i+1}}$ from the vision feedback. |
| 19: | Update predictions $\hat{\mathbf{p}}_{a,t_{i+1}}$ and $\hat{\psi}_{r,t_{i+1}}$. |
| 20: | $\tilde{\mathbf{p}}_{a,t_{i+1}} \leftarrow \mathbf{p}_{a,t_{i+1}} - \hat{\mathbf{p}}_{a,t_{i+1}}$<br>$\tilde{\psi}_{r,t_{i+1}} \leftarrow \psi_{r,t_{i+1}} - \hat{\psi}_{r,t_{i+1}}$ |
| 21: | $\hat{\eta}_{t_{i+1}} \leftarrow \hat{\eta}_{t_i} + \alpha_1\left(\|\mathbf{x}_{e,t_i}\|_2 \mathbf{a}^T_{3\times 1}\tilde{\mathbf{p}}_{a,t_{i+1}} + \alpha_2 \psi_{e,t_i}\tilde{\psi}_{r,t_{i+1}}\right)$<br>$\hat{\rho}_{t_{i+1}} \leftarrow \hat{\rho}_{t_i} + \alpha_3 \psi_{e,t_i}\tilde{\psi}_{r,t_{i+1}}$ |

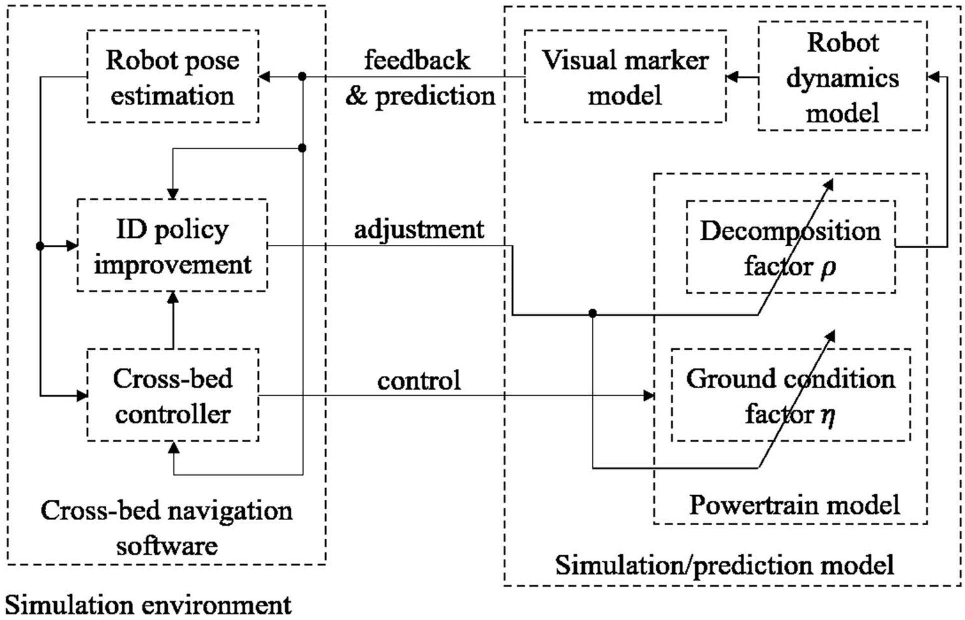


**FIGURE 6** Structure of the simulation environment

**TABLE 3** Some dimension parameters of the test field

| Names | Values (m) |
|---|---|
| Distance between the adjacent beds | 1.5 |
| Drive-off distance | 3.4 |
| Stopping safety distance | 0.9 |

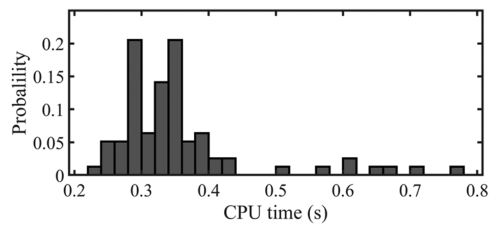


**FIGURE 7** Distribution of the CPU time spent on calculating the control command at each time step using the proposed algorithm in simulation. The average CPU time is 0.27 s

The benchmark DP generates the optimal solution in the whole state space. The PCP space is discretized into 70 nodes, while the search space of the benchmark DP algorithm is discretized into 4900 nodes. It is observed that the minimum-time row transition solution given by the benchmark DP algorithm is 44.4 s, whereas the proposed algorithm gives 44.7 s. This 0.3 s difference is due to the small oscillation in the control commands of the proposed algorithm between 28.2 s and 30.3 s. The minimum-time solution of the proposed algorithm is very close to that of the optimal one. As a significant feature, the proposed algorithm takes only 0.27 s on average to compute at each control time step. 90% of the control commands are computed within 0.52 s using the proposed algorithm (as shown in Figure 7). The corresponding computing time for the benchmark DP algorithm is 6.4 s on average (as shown in Figure 8). To conclude, the proposed algorithm is about 23 times faster than the benchmark DP algorithm on average, while delivering nearly an identical optimal solution.

Since the results obtained using the benchmark algorithm and the proposed algorithm are very similar, only the plots of the proposed algorithm are shown here. In the simulation, the robot drives off the first bed ("Bed 1") and then aligns with the second bed ("Bed 2"). The controlled robot trajectory in the simulation is shown in Figure 9. We refer to the visual markers fixed to Bed 1 and Bed 2 as Marker 1 and Marker 2, respectively. The open-loop trajectories are inserted in Figure 9 to help illustrate the complete cross-bed transition, which starts at the end of Bed 1. After the initial open-loop motion, the robot covers the drive-off distance with reference to Marker 1. Then, the proposed visual alignment controller takes over and guides the robot to achieve an alignment with Marker 2 in the minimum time. The local NED coordinate frame is fixed to Bed 2 during the cross-bed transition.

Figures 10 and 11 show the heading angle and the bearing angle of Marker 2 with respect to the robot position in the local NED coordinate frame during the alignment. Because the control commands (shown in Figure 12) are saturated in the attempt to achieve the minimum-time performance index, the alignment section appears to be separated into three stages. During Stage 1, the robot adjusts its heading angle towards the desired value. Since in this stage the left and right wheels rotate in the opposite directions at the constrained angular velocity, the robot stays at the circle maker position in Figure 9 with a constant bearing angle. During Stage 2, the robot translates towards its desired final alignment position

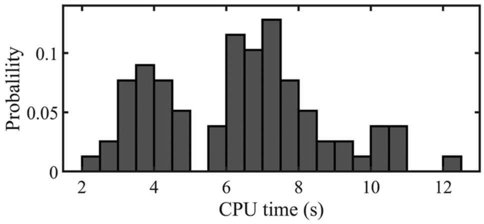


**FIGURE 8** Distribution of the CPU time spent on calculating the control command at each time step using the benchmark DP algorithm in simulation. The average CPU time is 6.4 s

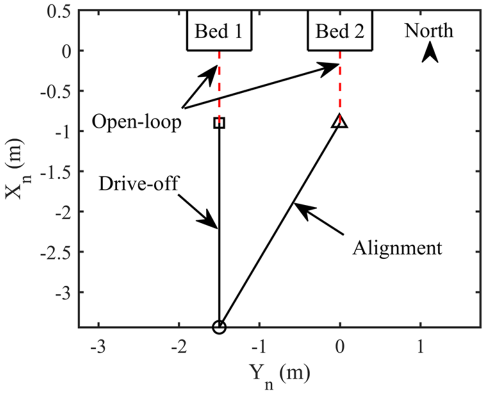


**FIGURE 9** Robot trajectory in the local north-east-down coordinates (Square: beginning of the drive-off section; Circle: beginning of the alignment section; Triangle: end of the alignment section.) [Color figure can be viewed at wileyonlinelibrary.com]

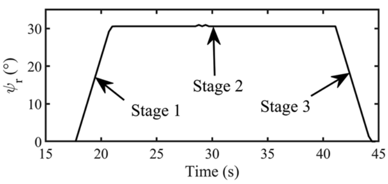


**FIGURE 10** Heading angle of the robot during the row alignment

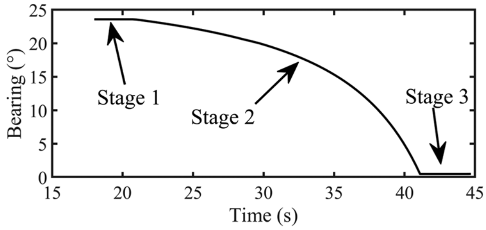


**FIGURE 11** Bearing angle of Marker 2 with respect to the robot position in the local north-east-down coordinates during the row alignment

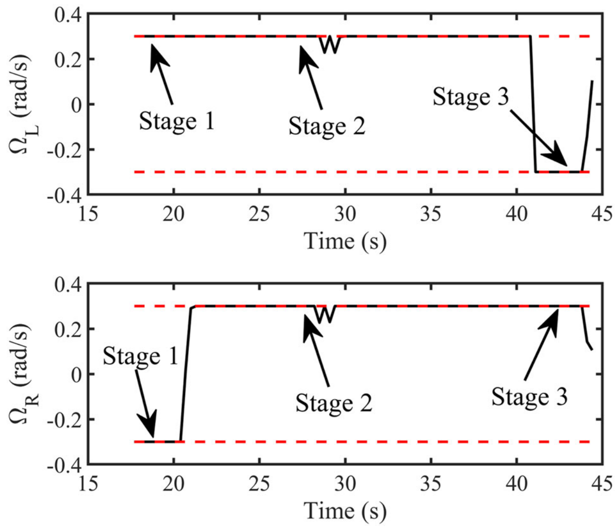


**FIGURE 12** Robot wheel angular velocity controls during the row alignment (Red-dashed lines: control constraints.) [Color figure can be viewed at wileyonlinelibrary.com]

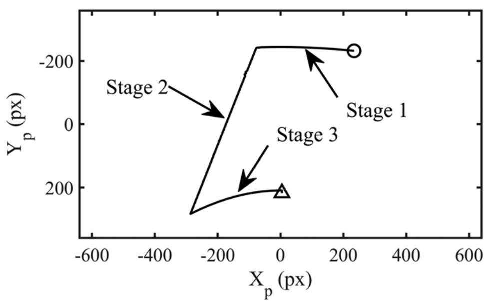


**FIGURE 13** Trajectory of the Marker 2 center point in the pixel coordinates during the row alignment (Circle: beginning of the alignment; Triangle: end of the alignment.)

with some micro adjustments in the heading angle. Therefore, its bearing angle steadily reduces towards 0° while its heading angle appears constant. At the end of Stage 2, the robot reaches the triangle marker position in Figure 9, and the first terminal constraint in Equation (8) is met. During Stage 3, the robot adjusts its heading angle to 0° to satisfy the second terminal constraint in Equation (8). Again, the left and right wheels rotate in the opposite directions at the saturated angular velocity and the robot position and bearing angle are constant.

Figures 13 and 14 present the trajectories of the Marker 2 center point during alignment in the pixel coordinates and the augmented pixel coordinate frame, respectively. Figure 13 shows that $x_m^P$ reaches 0 at the end of Stage 3, which partially reflects that the alignment condition Equation (23) is met. Due to the adjustment of the robot heading angle at Stages 1 and 3, the center point pixel location changes mostly along the $X^P$ axis, creating the steep changes in the line of sight (LOS) angle of the marker center point as shown in Figure 15. On the other hand, because of the robot transition during Stage 2, the center point trajectory moves steadily in the 3D augmented pixel coordinate frame, corresponding to the gradual LOS angle change shown in Figure 15. Figure 15 also shows that the controller can achieve alignment without violating the FOV constraint.

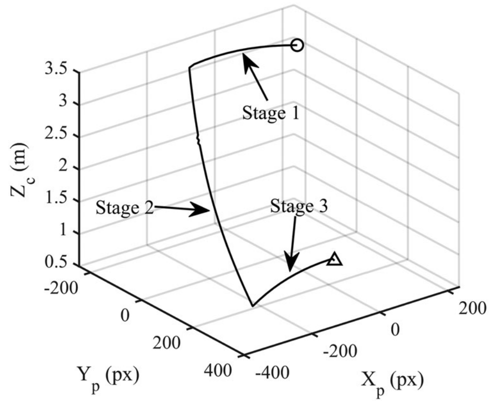


**FIGURE 14** Trajectory of the Marker 2 center point in the augmented pixel coordinates during the row alignment (Circle: beginning of the alignment; Triangle: end of the alignment.)

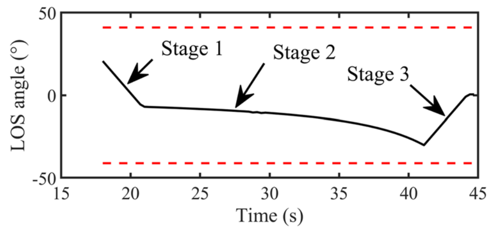


**FIGURE 15** Horizontal line of sight (LOS) angle of the Marker 2 center point in the camera FOV during the row alignment (Red-dashed lines: FOV constraints.). FOV, field of view [Color figure can be viewed at wileyonlinelibrary.com]

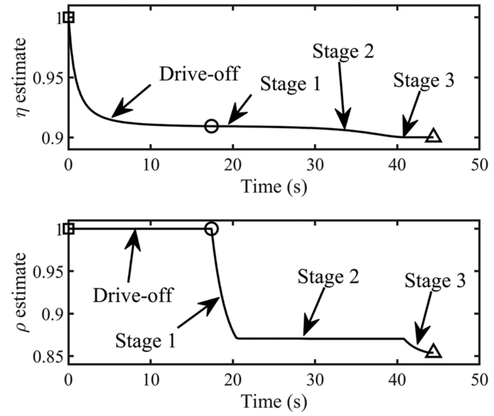


**FIGURE 16** Estimates of uncertainty parameters (Square: beginning of the drive-off section; Circle: beginning of the alignment section; Triangle: end of the alignment section.)

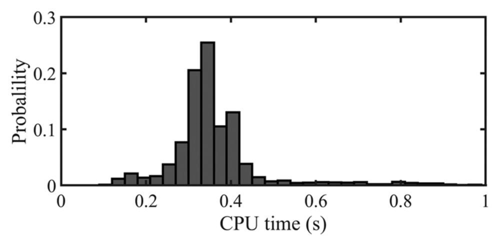


**FIGURE 17** Distribution of the CPU time spent on each control step for 20 cross-bed experiments

The ID policy improvement starts as soon as Marker 1 becomes visible to the robot, even before the ID-VMC-VI controller that relies on Marker 2 begins to work. The adaptive adjustments of the uncertainty parameters are shown in Figure 16. The adjustment to $\hat{\rho}$ is indiscernible during the drive-off and Stage 2 of alignment due to the fact that the decomposition factor $\rho$ has no effect on the marker location prediction in the straight translation. On the other hand, $\hat{\eta}$ is adjusted throughout the cross-bed transition because it affects all stages of the robot motion. Figure 16 shows that the estimates of both uncertainty parameters converge to the assumed values under the adaptive ID policy improvement rule.

## 7 | EXPERIMENT RESULTS AND DISCUSSION

### 7.1 | Experiment settings

The proposed cross-bed controller is tested in the field as shown in Figure 1 with the same task as described in the simulation section. To help illustrate the complete cross-bed trajectory, some over-bed and open-loop trajectory sections are included. However, the discussions about these non-cross-bed sections are beyond the scope of this paper.

The PCP space $\mathcal{V}^1$ is discretized into 50 nodes. The robot software (shown in Figure 3) is not real-time, therefore a fixed time step size cannot be guaranteed. Based on the statistics obtained in 20 field experiments, as shown in Figure 17, 93.31% of the control steps take less than 0.5 s to compute. Therefore, we choose the control time step size to be 0.5 s for the field experiments.

The step sizes of the ID policy improvement are chosen to be $\alpha_1 = 2 \times 10^{-6}$, $\alpha_2 = 1$, and $\alpha_3 = 6 \times 10^{-3}$. All the other experiment parameters follow Tables 1 and 3. During these experiments, the ground is dry and the surface has a loose layer of light sand. The length of the AprilTag edge is 0.172 m. Different 2D barcode images will be used for different rows, so the AprilTag visual fiducial system (Olson, 2011) can find out which row the robot is moving towards.

### 7.2 | Experiment results and analyses

The system was tuned once, involving 11 trials to better match the specific hardware and field conditions. The parameters to be tuned can be separated into three groups: the discretization of the PCP space, the constraints considering the actual dimensions of the field, and the adaptation step sizes. Once tuned, a total of 20 cross-bed experiments were conducted and all were successful. The statistics of the 20 experiments are shown in Figures 18 and 19, where the mean finishing time for the cross-bed section is 71.3 s. In comparison, a

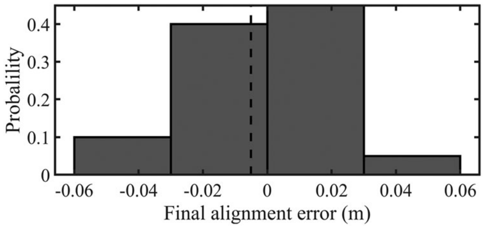


**FIGURE 18** Distribution of the final alignment errors for the 20 cross-bed experiments (Dashed line: the mean final alignment error.)

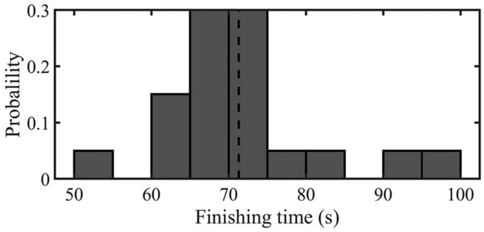


**FIGURE 19** Distribution of the finishing time of the cross-bed alignment section for the 20 experiments (Dashed line: the mean finishing time.)

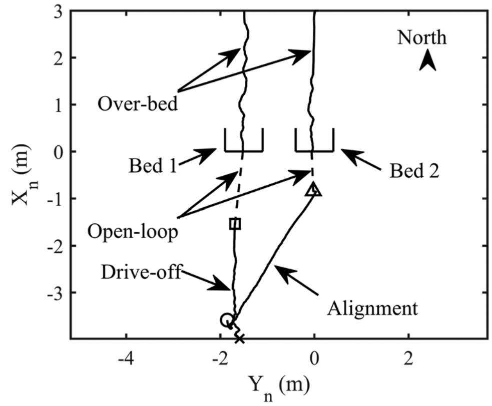


**FIGURE 20** Robot trajectory in the local north-east-down coordinates (Square: beginning of the drive-off section; Cross: end of the drive-off section; Circle: beginning of the alignment section; Triangle: end of the alignment section.)

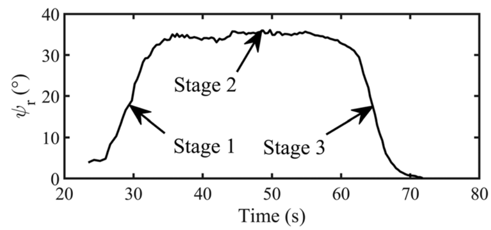


**FIGURE 21** Heading angle of the robot during the row alignment

typical cross-bed section based on Li et al. (2020) takes 138 s. The row transition time can be lower if the speed limit setting of the robot is higher. However, the robot may damage the soil in row headlines. The average row alignment error is 0.5 cm, and in most cases the alignment error is within 3 cm. This alignment accuracy is sufficient for the robot row transition. Once the robot is aligned with the row headline, the range finders will start to help the robot in its further alignment before it starts to move over the row. In the following paragraphs, we discuss the performance of the proposed controller using the results from one experiment.

The complete cross-bed trajectory from one experiment is shown in Figure 20. The robot starts traveling along Bed 1 relying on the over-bed controller (Li et al., 2020). Once it reaches the end of Bed 1, the robot drives off straight for 1 m in open-loop. Figure 20 shows that the robot actually ends at 1.54 m from Bed 1 at the end of the open-loop drive-off. This is where Marker 1 has entered the FOV of the camera and begins to provide visual feedback for the distance calculation and the ID policy improvement. The robot continues driving off for another 2.5 m, then the "scheduler" in Figure 3 initializes the cross-bed controller to align the robot with Marker 2. As shown in Figure 20, the proposed cross-bed controller is capable of aligning the robot with Marker 2 and stopping it at the predefined safety distance of 0.9 m at the end of the alignment section. The small gap between the end of the drive-off trajectory and the beginning of the alignment trajectory is partially due to the switching between two visual markers. Another reason for this gap is that the accuracy of the AprilTag visual fiducial system (Olson, 2011) falls as the distance between the tag and the camera increases. After the alignment, the robot drives in open-loop until the over-bed controller takes over, and the cross-bed translation is completed.

Corresponding to the alignment section in Figure 20, the following figures all share the three-stage pattern. During Stage 1, the control commands for the left and right motors saturate in opposite directions to steer the robot in the minimum time, resulting in the steep changes in the robot heading angle (shown in Figure 21) and the horizontal LOS angle of Marker 2 in the camera FOV (shown in Figure 22). The trajectories of the Marker 2 center point in the pixel coordinates (shown in Figure 23) and the augmented pixel coordinates (shown in Figure 24) also exhibit a distinguishable Stage 1 that changes mostly along the $X^P$ axis. During Stage 2, the robot steadily translates towards Marker 2 with gradual adjustments in the heading angle (shown in Figure 21). Stage 2 in Figure 25 shows that the proposed controller can reserve the difference between the left and right motor commands without violating the control constraint. While changes in the horizontal LOS angle of Marker 2 are slow (shown in Figure 22), its center point trajectories in the pixel and augmented pixel coordinates change dramatically along the $Y^P$ and $Z^C$ axes (shown in Figures 23 and 24). During Stage 3, the controller rotates the robot to satisfy the visual alignment condition. Figure 21 shows that at the end of Stage 3, the heading angle of the robot is 0°, indicating that the alignment condition is met. Figure 22 shows that the horizontal LOS angle of Marker 2 is close to 0 at the end of Stage 3, which meets the expectation of a visual alignment. The same is true at the end of Stage 3 as shown in Figures 23 and 24.

Figure 26 shows the ID policy improvement result in the experiment. At the end of the cross-bed transition, $\hat{\eta}$ and $\hat{\rho}$ converge to

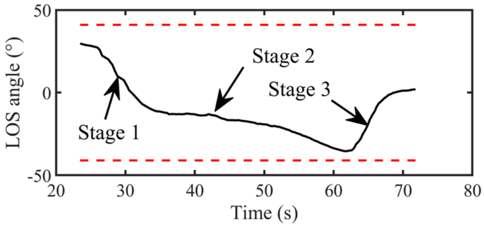


**FIGURE 22** Horizontal LOS angle of the Marker 2 center point in the camera FOV during the row alignment (Red-dashed lines: FOV constraints.) [Color figure can be viewed at wileyonlinelibrary.com]

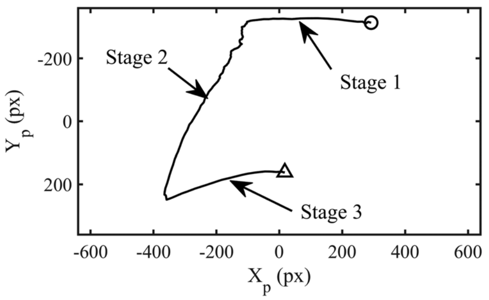


**FIGURE 23** Trajectory of the Marker 2 center point in the pixel coordinates during the row alignment (Circle: beginning of the alignment; Triangle: end of the alignment.)

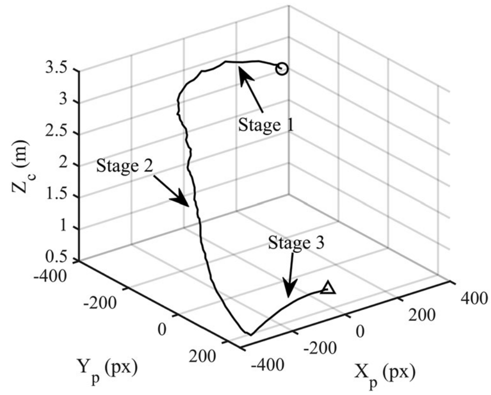


**FIGURE 24** Trajectory of the Marker 2 center point in the augmented pixel coordinates during the row alignment (Circle: beginning of the alignment; Triangle: end of the alignment.)

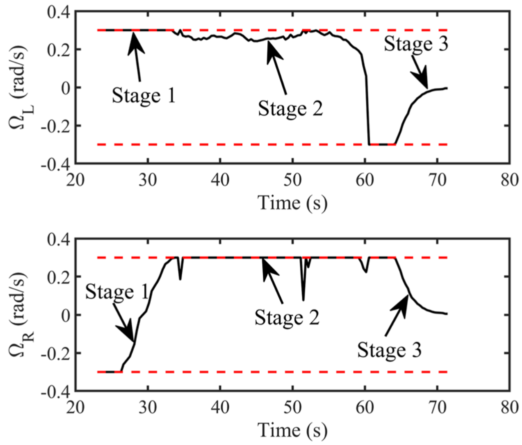


**FIGURE 25** Robot wheel angular velocity controls during the row alignment (Red-dashed lines: control constraints.) [Color figure can be viewed at wileyonlinelibrary.com]

0.66 and 0.96, respectively. The tendency shown in Figure 26 at different stages highly resembles those of the simulation result.

Overall, the experiment results demonstrate that the proposed minimum-time row transition controller can reliably accomplish the task while satisfying the control and state constraints.

## 8 | DISCUSSION

In this paper, we developed a vision-based minimum-time row transition solution for a field robot in a strawberry field. The solution has been tested with simulation and field trials. When implementing the proposed control algorithm, the constraints relating to the dimension of the field, the FOV of the camera, and the driving motors should be considered. These constraints will ensure that the trajectory satisfies the final alignment conditions while meeting the proper working conditions of the hardware. We consider the following aspects for future improvements.

(i) The proposed solution is currently implemented and demonstrated for transition operations between adjacent rows utilizing the front and rear cameras. Nevertheless, it can be easily modified for transition operations between remote rows by introducing a headland translation between the current drive-off and the alignment operations. Additional side cameras will be helpful to localize the robot during long headland translations.

(ii) The robot software can be further optimized. In the current implementation, the image processing task takes roughly 0.17 s to process 1280 × 720 image frame. In comparison, the average control loop CPU time is 0.36 s when running the software. It is expected that parallelizing the image processing task with the rest of the control loop will halve the runtime. Another observation is that, when implemented in MATLAB, the computational overhead of the current robot software is very high. Therefore, it is suggested to use a different programming language such as C/C++ to speed up the code execution.

(iii) We expect that the presented row transition solution can be customized for vision-based control applications other than those in agricultural fields. The underlying subspace ADP-based optimal control method can be adopted to solve a broad class of real-time optimal control problems subject to environment uncertainties.

(iv) The robot drive system design is sensitive to the soil type or surface condition of a field. We have conducted experiments in two extreme cases: flat solid surfaces in laboratory settings and very loose, sandy surfaces in a nearby commercial U-pick farm (a very challenging

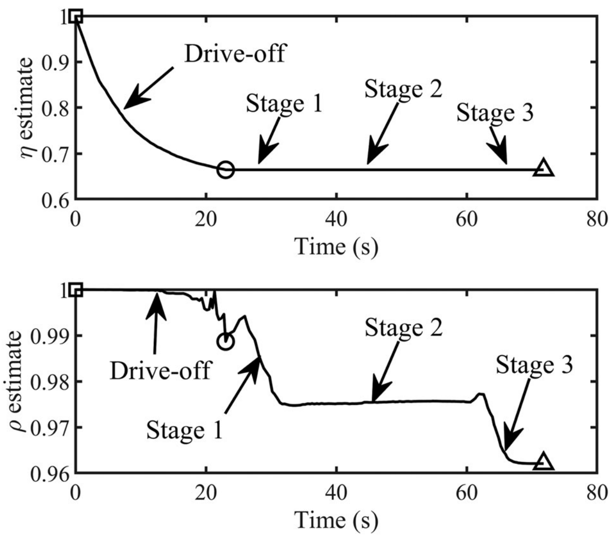


**FIGURE 26** Estimates of uncertainty parameters (Square: beginning of the drive-off section; Circle: beginning of the alignment section; Triangle: end of the alignment section.)

condition). Within the scope of this study, we demonstrate the effectiveness of the algorithm in a very challenging condition.

## 9 | CONCLUSION

Row transition motion control is a routine task for autonomous robots to traverse semistructured agricultural fields with raised beds. In this study, an ADP-based optimal control method in a pixel subspace is developed to improve the cross-bed performance of a robot designed for scouting strawberry fields using only RGB cameras. Visual markers at the end of strawberry beds are used to estimate the robot posture. Based on the derived row alignment conditions, the pixel trajectory of the visual marker center point is guided by a bioinspired motion rule, via which, the search space dimension of the algorithm is significantly reduced. The developed algorithm provides a real-time prediction of the optimal augmented pixel location-to-reach, which is converted to the robot pose-to-reach, and then used to calculate the control commands following the ID policy. Two uncertainty parameters that account for ground conditions and wear of drivetrain subsystem are adaptively adjusted based on the visual feedback to improve the control performance in field conditions.

The proposed algorithm shows great reliability and satisfactory performance in both simulation and field experiments. In simulation, the proposed algorithm generates a near-optimal solution in a fraction of the CPU time required by a benchmark DP method. Via the adaptation algorithm, uncertain parameters can rapidly converge to their actual values. Different from the widely used PID type controllers in row transition operations, the developed optimal control algorithm complies with the constraints imposed by the dimension of the field, the FOV of the cameras, and the driving motors. All 20 experiments were successful in row transition and showed a significant improvement in row transition time as compared with our previous PID-based design. A centimeter-level row alignment accuracy is achieved solely relying on a RGB camera.

In the future, implementation of the proposed row transition control algorithm can be further enhanced along the following directions. (i) Parallelizing the software and programming in C/C++ will further improve the overall computational efficiency. (ii) The current application is limited to transition operations between adjacent rows. Transition controls between remote rows can be the next interesting step. It is expected that the studied algorithm can be customized and adopted in a wide range of vision-based control applications.

## ACKNOWLEDGMENT

This study was supported by the National Science Foundation under Grant No. 1924622.

## DATA AVAILABILITY STATEMENT

Research data are not shared.

## ORCID

*Qiang Li* http://orcid.org/0000-0002-8775-5363

## REFERENCES

Angelova, A., Matthies, L., Helmick, D., & Perona, P. (2007). Learning and prediction of slip from visual information. *Journal of Field Robotics*, *24*(3), 205–231.

Arad, B., Balendonck, J., Barth, R., Ben-Shahar, O., Edan, Y., Hellström, T., Hemming, J., Kurtser, P., Ringdahl, O., Tielen, T., & Tuijl, B. (2020). Development of a sweet pepper harvesting robot. *Journal of Field Robotics*, *37*(6), 1027–1039.

Åstrand, B., & Baerveldt, A. J. (2002). An agricultural mobile robot with vision-based perception for mechanical weed control. *Autonomous Robots*, *13*(1), 21–35.

Auat Cheein, F., Torres-Torriti, M., Hopfenblatt, N. B., Prado, Á. J., & Calabi, D. (2017). Agricultural service unit motion planning under harvesting scheduling and terrain constraints. *Journal of Field Robotics*, *34*(8), 1531–1542.

Bak, T., & Jakobsen, H. (2004). Agricultural robotic platform with four wheel steering for weed detection. *Biosystems Engineering*, *87*(2), 125–136.

Bakker, T. (2009). *An autonomous robot for weed control—design, navigation and control* (Ph.D. thesis), Wageningen University. https://library.wur.nl/WebQuery/edepot/1099

Ball, D., Upcroft, B., Wyeth, G., Corke, P., English, A., Ross, P., Patten, T., Fitch, R., Sukkarieh, S., & Bate, A. (2016). Vision-based obstacle detection and navigation for an agricultural robot. *Journal of Field Robotics*, *33*(8), 1107–1130.

Bertsekas, D. P. (2013). Rollout algorithms for discrete optimization: A survey, *Handbook of Combinatorial Optimization* (pp. 2989–3013). Springer.

Bertsekas, D. P. (2020). *Rollout, policy iteration, and distributed reinforcement learning*. Athena Scientific.

Birrell, S., Hughes, J., Cai, J. Y., & Iida, F. (2020). A field-tested robotic harvesting system for iceberg lettuce. *Journal of Field Robotics*, *37*(2), 225–245.

Cariou, C., Lenain, R., Thuilot, B., & Berducat, M. (2009). Automatic guidance of a four-wheel-steering mobile robot for accurate field operations. *Journal of Field Robotics*, *26*(6–7), 504–518.

Curry, H. B. (1944). The method of steepest descent for non-linear minimization problems. *Quarterly of Applied Mathematics*, *2*(3), 258–261.

de la Puente, P., & Rodríguez-Losada, D. (2014). Feature based graph-SLAM in structured environments. *Autonomous Robots*, *37*(3), 243–260.

de Paula Veronese, L., Auat Cheein, F., Bastos-Filho, T., de Souza, A. F., & de Aguiar, E. (2016). A computational geometry approach for localization and tracking in GPS-denied environments. *Journal of Field Robotics*, *33*(7), 946–966.

Defterli, S. G., Shi, Y., Xu, Y., & Ehsani, R. (2016). Review of robotic technology for strawberry production. *Applied Engineering in Agriculture*, *32*(3), 301–318.

Dong, F., Heinemann, W., & Kasper, R. (2011). Development of a row guidance system for an autonomous robot for white asparagus harvesting. *Computers and Electronics in Agriculture*, *79*(2), 216–225.

Dusadeerungsikul, P. O., & Nof, S. Y. (2019). A collaborative control protocol for agricultural robot routing with online adaptation. *Computers & Industrial Engineering*, *135*, 456–466.

Fang, H., Fan, R., Thuilot, B., & Martinet, P. (2006). Trajectory tracking control of farm vehicles in presence of sliding. *Robotics and Autonomous Systems*, *54*(10), 828–839.

Fiala, M. (2005). ARTag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Vol. 2, pp. 590-596). IEEE.

Forrest-Barlach, M., & Babcock, S. (1987). Inverse dynamics position control of a compliant manipulator. *IEEE Journal on Robotics and Automation*, *3*(1), 75–83.

Freese, D. J., & Xu, Y. (2019). Multiphase scouting control of an agricultural field robot with reachability analyses. *Journal of Dynamic Systems, Measurement and Control*, *141*(5), 051009.

Gai, J., Xiang, L., & Tang, L. (2021). Using a depth camera for crop row detection and mapping for under-canopy navigation of agricultural robotic vehicle. *Computers and Electronics in Agriculture*, *188*, 106301.

Hague, T., & Tillett, N. D. (1996). Navigation and control of an autonomous horticultural robot. *Mechatronics*, *6*(2), 165–180.

Higuti, V. A., Velasquez, A. E., Magalhaes, D. V., Becker, M., & Chowdhary, G. (2019). Under canopy light detection and ranging-based autonomous navigation. *Journal of Field Robotics*, *36*(3), 547–567.

Hiremath, S. A., van der Heijden, G. W., van Evert, F. K., Stein, A., & ter Braak, C. J. (2014). Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter. *Computers and Electronics in Agriculture*, *100*, 41–50.

Howard, R. A. (1960). *Dynamic programming and Markov processes*. MIT Press.

Ji, Y., Yamashita, A., & Asama, H. (2015). RGB-D SLAM using vanishing point and door plate information in corridor environment. *Intelligent Service Robotics*, *8*(2), 105–114.

Kalaitzakis, M., Cain, B., Carroll, S., Ambrosi, A., Whitehead, C., & Vitzilaios, N. (2021). Fiducial markers for pose estimation. *Journal of Intelligent & Robotic Systems*, *101*(4), 1–26.

Kayacan, E., Young, S. N., Peschel, J. M., & Chowdhary, G. (2018). High-precision control of tracked field robots in the presence of unknown traction coefficients. *Journal of Field Robotics*, *35*(7), 1050–1062.

Kneip, J., Fleischmann, P., & Berns, K. (2020). Crop edge detection based on stereo vision. *Robotics and Autonomous Systems*, *123*, 103323.

Kragh, M., & Underwood, J. (2020). Multimodal obstacle detection in unstructured environments with conditional random fields. *Journal of Field Robotics*, *37*(1), 53–72.

Laumond, J. P., Sekhavat, S., & Lamiraux, F. (1998). Guidelines in nonholonomic motion planning for mobile robots, *Robot Motion Planning and Control* (pp. 1–53). Springer.

Le, T. D., Ponnambalam, V. R., Gjevestad, J. G., & From, P. J. (2020). A low-cost and efficient autonomous row-following robot for food production in polytunnels. *Journal of Field Robotics*, *37*(2), 309–321.

Li, Q., Kong, X., & Xu, Y. (2020). Marker based row alignment control for an agricultural scouting robot. In *Dynamic Systems and Control Conference* (Vol. 84270, p. V001T05A005). ASME.

Li, Q., & Xu, Y. (2020). Unmanned aerial vehicle angular velocity control via reinforcement learning in dimension reduced search spaces. In *2020 American Control Conference (ACC)* (pp. 4926-4931). IEEE.

Liang, C. Y., Allen, R. W., Rosenthal, T. J., Chrstos, J. P., & Nunez, P. (2004). Tire modeling for off-road vehicle simulation. *SAE Transactions*, *113*, 1063–1073.

Liang, Z., Gao, X., Chen, Y., & Zhu, S. (2013). A novel loop closure detection method in monocular SLAM. *Intelligent Service Robotics*, *6*(2), 79–87.

Mahmud, M. S. A., Abidin, M. S. Z., Mohamed, Z., Rahman, M. K. I. A., & Iida, M. (2019). Multi-objective path planner for an agricultural mobile robot in a virtual greenhouse environment. *Computers and Electronics in Agriculture*, *157*, 488–499.

Marchant, J. A., Hague, T., & Tillet, N. D. (1997). Row-following accuracy of an autonomous vision-guided agricultural vehicle. *Computers and Electronics in Agriculture*, *16*(2), 165–175.

Mayne, D. Q., & Raković, S. (2003). Model predictive control of constrained piecewise affine discrete-time systems. *International Journal of Robust and Nonlinear Control*, *13*(3-4), 261–279.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, *36*(6), 789–814.

Menendez-Aponte, P., Garcia, C., Freese, D., Defterli, S., & Xu, Y. (2016). Software and hardware architectures in cooperative aerial and ground robots for agricultural disease detection. In *2016 International Conference on Collaboration Technologies and Systems (CTS)* (pp. 354-358). IEEE.

Nagasaka, Y., Saito, H., Tamaki, K., Seki, M., Kobayashi, K., & Taniwaki, K. (2009). An autonomous rice transplanter guided by global positioning system and inertial measurement unit. *Journal of Field Robotics*, *26*(6-7), 537–548.

Narvaez, F. Y., Reina, G., Torres-Torriti, M., Kantor, G., & Cheein, F. A. (2017). A survey of ranging and imaging techniques for precision agriculture phenotyping. *IEEE/ASME Transactions on Mechatronics*, *22*(6), 2428–2439.

Nguyen, T. H., Nguyen, T. M., & Xie, L. (2020). Tightly-coupled ultra-wideband-aided monocular visual SLAM with degenerate anchor configurations. *Autonomous Robots*, *44*(8), 1519–1534.

Ni, Z., He, H., Zhong, X., & Prokhorov, D. V. (2015). Model-free dual heuristic dynamic programming. *IEEE Transactions on Neural Networks and Learning Systems*, *26*, 1834–1839.

Noguchi, N., & Terao, H. (1997). Path planning of an agricultural mobile robot by neural network and genetic algorithm. *Computers and Electronics in Agriculture*, *18*(2-3), 187–204.

Nosair, H., Yang, Y., & Lee, J. M. (2010). Min-max control using parametric approximate dynamic programming. *Control Engineering Practice*, *18*, 190–197.

Olson, E. (2011). AprilTag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation* (pp. 3400–3407). IEEE.

Rankin, A. L., & Matthies, L. H. (2010). Passive sensor evaluation for unmanned ground vehicle mud detection. *Journal of Field Robotics*, *27*(4), 473–490.

Schor, N., Bechar, A., Ignat, T., Dombrovsky, A., Elad, Y., & Berman, S. (2016). Robotic disease detection in greenhouses: Combined detection of powdery mildew and tomato spotted wilt virus. *IEEE Robotics and Automation Letters*, *1*(1), 354–360.

Schwendner, J., Joyeux, S., & Kirchner, F. (2014). Using embodied data for localization and mapping. *Journal of Field Robotics*, *31*(2), 263–295.

Shangguan, L., Thomasson, J. A., & Gopalswamy, S. (2021). Motion planning for autonomous grain carts. *IEEE Transactions on Vehicular Technology*, *70*(3), 2112–2123.

Shojaei, K. (2021). Intelligent coordinated control of an autonomous tractor-trailer and a combine harvester. *European Journal of Control*, *59*, 82–98.

Srinivasan, M. V., & Davey, M. (1995). Strategies for active camouflage of motion. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, *259*(1354), 19–25.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*, 9–44.

Thuilot, B., Cariou, C., Martinet, P., & Berducat, M. (2002). Automatic guidance of a farm tractor relying on a single CP-DGPS. *Autonomous Robots*, *13*(1), 53–71.

Tu, X., Gai, J., & Tang, L. (2019). Robust navigation control of a 4WD/4WS agricultural robotic vehicle. *Computers and Electronics in Agriculture*, *164*, 104892.

Underwood, J. P., Jagbrant, G., Nieto, J. I., & Sukkarieh, S. (2015). Lidar-based tree recognition and platform localization in orchards. *Journal of Field Robotics*, *32*(8), 1056–1074.

Underwood, J. P., Wendel, A., Schofield, B., McMurray, L., & Kimber, R. (2017). Efficient in-field plant phenomics for row-crops with an autonomous ground vehicle. *Journal of Field Robotics*, *34*(6), 1061–1083.

United Nations. (2019). *World population prospects 2019*. United Nations.

Valin, H., Sands, R. D., Van der Mensbrugghe, D., Nelson, G. C., Ahammad, H., Blanc, E., Bodirsky, B., Fujimori, S., Hasegawa, T., Havlik, P., Heyhoe, E., Kyle, P., Mason-D'Croz, D., Paltsev, S., Rolinski, S., Tabeau, A., van Meijl, H., von Lampe, M., & Willenbockel, D. (2014). The future of food demand: Understanding differences in global economic models. *Agricultural Economics*, *45*(1), 51–67.

van Henten, E. J., Hemming, J., van Tuijl, B. A. J., Kornet, J. G., Meuleman, J., Bontsema, J., & van Os, E. A. (2002). An autonomous robot for harvesting cucumbers in greenhouses. *Autonomous Robots*, *13*(3), 241–258.

Wei, Q., Liu, D., Lin, Q., & Song, R. (2017). Discrete-time optimal control via local policy iteration adaptive dynamic programming. *IEEE Transactions on Cybernetics*, *47*(10), 3367–3379.

Winterhalter, W., Fleckenstein, F. V., Dornhege, C., & Burgard, W. (2018). Crop row detection on tiny plants with the pattern hough transform. *IEEE Robotics and Automation Letters*, *3*(4), 3394–3401.

Winterhalter, W., Fleckenstein, F. V., Dornhege, C., & Burgard, W. (2021). Localization for precision navigation in agricultural fields—beyond crop row following. *Journal of Field Robotics*, *38*, 429–451.

Xiong, Y., Ge, Y., Grimstad, L., & From, P. J. (2020). An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation. *Journal of Field Robotics*, *37*(2), 202–224.

Xu, Y., & Li, N. (2014). Bio-inspired varying subspace based computational framework for a class of nonlinear constrained optimal trajectory planning problems. *Bioinspiration & Biomimetics*, *9*(3), 036010.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(11), 1330–1334.

Zhou, Y., Kampen, E., & Chu, Q. P. (2018). Incremental model based online dual heuristic programming for nonlinear adaptive control. *Control Engineering Practice*, *73*, 13–25.



**How to cite this article:** Li, Q., & Xu, Y. (2021). Minimum-time row transition control of a vision guided agricultural robot. *Journal of Field Robotics*, 1–20. https://doi.org/10.1002/rob.22053


## APPENDIX A

To solve Equation (50), the basic gradient descent procedure in Curry (1944) is followed, that is,

$$\hat{\boldsymbol{\chi}}_{t_{i+1}} = \hat{\boldsymbol{\chi}}_{t_i} - \alpha\left(\frac{\partial e_{t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}}\right)^T, \tag{A1}$$

where $\alpha > 0$ is an update step size, and

$$\begin{aligned}
\frac{\partial e_{t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}} &= \tilde{\mathbf{p}}^T_{a,t_{i+1}} \frac{\partial \tilde{\mathbf{p}}_{a,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}} + \tilde{\psi}_{r,t_{i+1}} \frac{\partial \tilde{\psi}_{r,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}} \\
&= \tilde{\mathbf{p}}^T_{a,t_{i+1}} \left(\frac{\partial \mathbf{p}_{a,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}} - \frac{\partial \hat{\mathbf{p}}_{a,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}}\right) + \tilde{\psi}_{r,t_{i+1}} \left(\frac{\partial \psi_{r,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}} - \frac{\partial \hat{\psi}_{r,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}}\right) \\
&= \tilde{\mathbf{p}}_{a,t_{i+1}} \left(\frac{\partial \mathbf{p}_{a,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}} - \frac{\partial \hat{\mathbf{p}}_{a,t_{i+1}}}{\partial \hat{\mathbf{x}}^N_{r,t_{i+1}}} \frac{\partial \hat{\mathbf{x}}^N_{r,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}}\right) + \tilde{\psi}_{r,t_{i+1}} \left(\frac{\partial \psi_{r,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}} - \frac{\partial \hat{\psi}_{r,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}}\right).
\end{aligned} \tag{A2}$$

At $t_i$, the control inputs are calculated via the ID approach in Equation (5) based on $\hat{\boldsymbol{\chi}}_{t_i}$, that is,

$$\mathbf{u}_{t_i} = \frac{1}{2Tr_w \hat{\eta}_{t_i}} \begin{bmatrix} \|\mathbf{x}_{e,t_i}\|_2 + \frac{1}{2\hat{\rho}_{t_i}} \psi_{e,t_i} W_r \\ \|\mathbf{x}_{e,t_i}\|_2 - \frac{1}{2\hat{\rho}_{t_i}} \psi_{e,t_i} W_r \end{bmatrix}. \tag{A3}$$

Meanwhile, by substituting $\hat{\boldsymbol{\chi}}_{t_i}$ into Equation (1), we have the state prediction following:

$$\begin{bmatrix} \hat{\mathbf{x}}^N_{r,t_{i+1}} \\ \hat{\psi}_{r,t_{i+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^N_{r,t_i} \\ \psi_{r,t_i} \end{bmatrix} + Tr_w \hat{\eta}_{t_i} \begin{bmatrix} \cos\psi_{r,t_i} & \cos\psi_{r,t_i} \\ \sin\psi_{r,t_i} & \sin\psi_{r,t_i} \\ 2\hat{\rho}_{t_i}/W_r & -2\hat{\rho}_{t_i}/W_r \end{bmatrix} \mathbf{u}_{t_i}. \tag{A4}$$

Substituting Equation (A3) into Equation (A4) yields

$$\begin{bmatrix} \hat{\mathbf{x}}^N_{r,t_{i+1}} \\ \hat{\psi}_{r,t_{i+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^N_{r,t_i} \\ \psi_{r,t_i} \end{bmatrix} + \begin{bmatrix} \|\mathbf{x}_{e,t_i}\|_2 \cos\psi_{r,t_i} \\ \|\mathbf{x}_{e,t_i}\|_2 \sin\psi_{r,t_i} \\ \psi_{e,t_i} \end{bmatrix}, \tag{A5}$$

from which it is apparent that $\hat{\mathbf{x}}^N_{r,t_{i+1}}$ and $\hat{\psi}_{r,t_{i+1}}$ are not functions of $\hat{\boldsymbol{\chi}}_{t_i}$. Therefore, $\partial \hat{\mathbf{x}}^N_{r,t_{i+1}}/\partial \hat{\boldsymbol{\chi}}_{t_i} = \mathbf{0}$ and $\partial \hat{\psi}_{r,t_{i+1}}/\partial \hat{\boldsymbol{\chi}}_{t_i} = \mathbf{0}$. Thus, Equation (A2) can be simplified as follows:

$$\frac{\partial e_{t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}} = \tilde{\mathbf{p}}^T_{a,t_{i+1}} \frac{\partial \mathbf{p}_{a,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}} + \tilde{\psi}_{r,t_{i+1}} \frac{\partial \psi_{r,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}}, \tag{A6}$$

where, following the chain rule, we have

$$\frac{\partial \mathbf{p}_{a,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}} = \frac{\partial \mathbf{p}_{a,t_{i+1}}}{\partial \mathbf{x}^N_{r,t_{i+1}}} \frac{\partial \mathbf{x}^N_{r,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}}. \tag{A7}$$

To calculate the derivatives on the right-hand side of Equation (A7), let us substitute Equation (A3) into the actual dynamics Equation (1), leading to

$$\begin{bmatrix} \mathbf{x}^N_{r,t_{i+1}} \\ \psi_{r,t_{i+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^N_{r,t_i} \\ \psi_{r,t_i} \end{bmatrix} + \frac{\eta}{\hat{\eta}_{t_i}} \begin{bmatrix} \|\mathbf{x}_{e,t_i}\|_2 \cos\psi_{r,t_i} \\ \|\mathbf{x}_{e,t_i}\|_2 \sin\psi_{r,t_i} \\ \frac{\rho}{\hat{\rho}_{t_i}} \psi_{e,t_i} \end{bmatrix}. \tag{A8}$$

Extracting the first two equations from Equation (A8) gives

$$\mathbf{x}^N_{r,t_{i+1}} = \mathbf{x}^N_{r,t_i} + \frac{\eta}{\hat{\eta}_{t_i}}\|\mathbf{x}_{e,t_i}\|_2 \begin{bmatrix}\cos\psi_{r,t_i}\\ \sin\psi_{r,t_i}\end{bmatrix} \tag{A9}$$

Note that $\|\mathbf{x}_{e,t_i}\|_2$ is not a function of $\hat{\boldsymbol{\chi}}_{t_i}$. Take the partial derivative of Equation (A9) with respect to $\hat{\boldsymbol{\chi}}_{t_i}$, the following equation is achieved:

$$\frac{\partial \mathbf{x}^N_{r,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}} = \begin{bmatrix}\cos\psi_{r,t_i}\\ \sin\psi_{r,t_i}\end{bmatrix}\begin{bmatrix}-\frac{\eta}{\hat{\eta}^2_{t_i}}\|\mathbf{x}_{e,t_i}\|_2 & 0\end{bmatrix} \tag{A10}$$

Next, we are going to calculate $\partial \mathbf{p}_{a,t_{i+1}}/\partial \mathbf{x}^N_{r,t_{i+1}}$ in Equation (A7). Since the third element of $\mathbf{p}$ is 1, from the third equality in Equation (22), we have

$$z^C_{m,t_{i+1}} = -x^R_c\cos\theta_c - \left(z^N_m - z^N_r\right)\sin\theta_c - x^N_{r,t_{i+1}}\cos\theta_c\cos\psi_{r,t_{i+1}} - y^N_{r,t_{i+1}}\cos\theta_c\sin\psi_{r,t_{i+1}}, \tag{A11}$$

which when substituted back into Equation (22) yields

$$\mathbf{p}_{a,t_{i+1}} = \begin{bmatrix} f_x\dfrac{y^N_{r,t_{i+1}}\cos\psi_{r,t_{i+1}} - x^N_{r,t_{i+1}}\sin\psi_{r,t_{i+1}}}{\left(x^R_c + x^N_{r,t_{i+1}}\cos\psi_{r,t_{i+1}} + y^N_{r,t_{i+1}}\sin\psi_{r,t_{i+1}}\right)\cos\theta_c + \left(z^N_m - z^N_{r,t_{i+1}}\right)\sin\theta_c} \\ f_y\dfrac{\left(x^R_c + x^N_{r,t_{i+1}}\cos\psi_{r,t_{i+1}} + y^N_{r,t_{i+1}}\sin\psi_{r,t_{i+1}}\right)\sin\theta_c - \left(z^N_m - z^N_{r,t_{i+1}}\right)\cos\theta_c}{\left(x^R_c + x^N_{r,t_{i+1}}\cos\psi_{r,t_{i+1}} + y^N_{r,t_{i+1}}\sin\psi_{r,t_{i+1}}\right)\cos\theta_c + \left(z^N_m - z^N_{r,t_{i+1}}\right)\sin\theta_c} \\ -\left(x^R_c + x^N_{r,t_{i+1}}\cos\psi_{r,t_{i+1}} + y^N_{r,t_{i+1}}\sin\psi_{r,t_{i+1}}\right)\cos\theta_c - \left(z^N_m - z^N_{r,t_{i+1}}\right)\sin\theta_c \end{bmatrix}. \tag{A12}$$

Taking the derivative of the above expression with respect to $\mathbf{x}^N_{r,t_{i+1}}$ leads to

$$\frac{\partial \mathbf{p}_{a,t_{i+1}}}{\partial \mathbf{x}^{(n)}_{r,t_{i+1}}} = \begin{bmatrix} -\dfrac{f_x\left[\left(z^N_m - z^N_r\right)\sin\psi_{r,t_{i+1}}\sin\theta_c + x^R_c\sin\psi_{r,t_{i+1}}\cos\theta_c + y^N_{r,t_{i+1}}\cos\theta_c\right]}{\left[\left(x^R_c + x^N_{r,t_{i+1}}\cos\psi_{r,t_{i+1}} + y^N_{r,t_{i+1}}\sin\psi_{r,t_{i+1}}\right)\cos\theta_c + \left(z^N_m - z^N_{r,t_{i+1}}\right)\sin\theta_c\right]^2} & \dfrac{f_x\left[\left(z^N_m - z^N_r\right)\cos\psi_{r,t_{i+1}}\sin\theta_c + x^R_c\cos\psi_{r,t_{i+1}}\cos\theta_c + x^N_{r,t_{i+1}}\cos\theta_c\right]}{\left[\left(x^R_c + x^N_{r,t_{i+1}}\cos\psi_{r,t_{i+1}} + y^N_{r,t_{i+1}}\sin\psi_{r,t_{i+1}}\right)\cos\theta_c + \left(z^N_m - z^N_{r,t_{i+1}}\right)\sin\theta_c\right]} \\ \dfrac{f_y\left(z^N_m - z^N_r\right)\cos\psi_{r,t_{i+1}}}{\left[\left(x^R_c + x^N_{r,t_{i+1}}\cos\psi_{r,t_{i+1}} + y^N_{r,t_{i+1}}\sin\psi_{r,t_{i+1}}\right)\cos\theta_c + \left(z^N_m - z^N_{r,t_{i+1}}\right)\sin\theta_c\right]} & \dfrac{f_y\left(z^N_m - z^N_r\right)\sin\psi_{r,t_{i+1}}}{\left[\left(x^R_c + x^N_{r,t_{i+1}}\cos\psi_{r,t_{i+1}} + y^N_{r,t_{i+1}}\sin\psi_{r,t_{i+1}}\right)\cos\theta_c + \left(z^N_m - z^N_{r,t_{i+1}}\right)\sin\theta_c\right]} \\ -\cos\theta_c\cos\psi_{r,t_{i+1}} & -\cos\theta_c\sin\psi_{r,t_{i+1}} \end{bmatrix}. \tag{A13}$$

Substituting Equation (A10) and Equation (A13) into Equation (A7), we have

$$\frac{\partial \mathbf{p}_{a,t_{i+1}}}{\partial \hat{\boldsymbol{\chi}}_{t_i}} = -\frac{\eta}{\hat{\eta}^2_{t_i}}\|\mathbf{x}_{e,t_i}\|_2\begin{bmatrix}\mathbf{a}_{3\times1} & \mathbf{0}_{3\times1}\end{bmatrix}, \tag{A14}$$

where $\mathbf{0}_{3\times1}$ is a zero vector and

$$\mathbf{a}_{3\times1} = \begin{bmatrix} \dfrac{f_x\left\{\left[\left(z^N_m - z^N_r\right)\sin\theta_c + x^R_c\cos\theta_c\right]\sin\left(\psi_{r,t_i} - \psi_{r,t_{i+1}}\right) + \left(x^N_{r,t_{i+1}}\sin\psi_{r,t_i} - y^N_{r,t_{i+1}}\cos\psi_{r,t_i}\right)\cos\theta_c\right\}}{\left[\left(x^R_c + x^N_{r,t_{i+1}}\cos\psi_{r,t_{i+1}} + y^N_{r,t_{i+1}}\sin\psi_{r,t_{i+1}}\right)\cos\theta_c + \left(z^N_m - z^N_{r,t_{i+1}}\right)\sin\theta_c\right]^2} \\ \dfrac{f_y\left(z^N_m - z^N_r\right)\cos\left(\psi_{r,t_i} - \psi_{r,t_{i+1}}\right)}{\left[\left(x^R_c + x^N_{r,t_{i+1}}\cos\psi_{r,t_{i+1}} + y^N_{r,t_{i+1}}\sin\psi_{r,t_{i+1}}\right)\cos\theta_c + \left(z^N_m - z^N_{r,t_{i+1}}\right)\sin\theta_c\right]^2} \\ -\cos\theta_c\cos\left(\psi_{r,t_i} - \psi_{r,t_{i+1}}\right) \end{bmatrix} \tag{A15}$$

To calculate $\partial(\psi_{r,t_{i+1}})^T/\partial\hat{\boldsymbol{\chi}}_{t_i}$, the third equation from Equation (A8) is rewritten as

$$\psi_{r,t_{i+1}} = \psi_{r,t_i} + \frac{\eta\rho}{\hat{\eta}_{t_i}\hat{\rho}_{t_i}}\psi_{e,t_i}. \tag{A16}$$

The partial derivative of Equation (A16) with respect to $\hat{\boldsymbol{\chi}}_{t_i}$ is

$$\frac{\partial\psi_{r,t_{i+1}}}{\partial\hat{\boldsymbol{\chi}}_{t_i}} = -\frac{\eta\rho}{\hat{\eta}^2_{t_i}\hat{\rho}^2_{t_i}}\psi_{e,t_i}\begin{bmatrix}\hat{\rho}_{t_i} & \hat{\eta}_{t_i}\end{bmatrix}. \tag{A17}$$

Substituting Equation (A14) and Equation (A17) into Equation (A6), the following equation:

$$\begin{aligned}\frac{\partial e_{t_{i+1}}}{\partial\hat{\boldsymbol{\chi}}_{t_i}} = &-\frac{\eta}{\hat{\eta}^2_{t_i}}\|\mathbf{x}_{e,t_i}\|_2\tilde{\mathbf{p}}^T_{a,t_{i+1}}\begin{bmatrix}\mathbf{a}_{3\times1} & \mathbf{0}_{3\times1}\end{bmatrix}\\ &-\frac{\eta\rho}{\hat{\eta}^2_{t_i}\hat{\rho}^2_{t_i}}\psi_{e,t_i}\tilde{\psi}_{r,t_{i+1}}\begin{bmatrix}\hat{\rho}_{t_i} & \hat{\eta}_{t_i}\end{bmatrix}\end{aligned} \tag{A18}$$

is derived, which is then substituted into Equation (A1) to obtain

$$\begin{aligned}\hat{\boldsymbol{\chi}}_{t_{i+1}} = \hat{\boldsymbol{\chi}}_{t_i} + \alpha\Bigg(&\frac{\eta}{\hat{\eta}^2_{t_i}}\|\mathbf{x}_{e,t_i}\|_2\begin{bmatrix}\mathbf{a}_{3\times1} & \mathbf{0}_{3\times1}\end{bmatrix}^T\tilde{\mathbf{p}}_{a,t_{i+1}}\\ &+\frac{\eta\rho}{\hat{\eta}^2_{t_i}\hat{\rho}^2_{t_i}}\psi_{e,t_i}\tilde{\psi}_{r,t_{i+1}}\begin{bmatrix}\hat{\rho}_{t_i} & \hat{\eta}_{t_i}\end{bmatrix}^T\Bigg).\end{aligned} \tag{A19}$$

In Equation (A19), $\eta$ and $\rho$ are the real values and unknown. Yet, since $0 < \eta < 1$, the direction of the gradient is not affected by the uncertainty in $\eta$. The same is true for $\rho$. By defining $\alpha_1 = \alpha\eta/\hat{\eta}^2_{t_i}$, $\alpha_2 = \rho/\hat{\rho}_{t_i}$ and $\alpha_3 = \alpha\eta\rho/(\hat{\eta}_{t_i}\hat{\rho}^2_{t_i})$, we have the respective update rules for $\hat{\eta}$ and $\hat{\rho}$ as

$$\begin{aligned}\hat{\eta}_{t_{i+1}} &= \hat{\eta}_{t_i} + \alpha_1\left(\left\|\mathbf{x}_{e,t_i}\right\|_2\mathbf{a}^T_{3\times1}\tilde{\mathbf{p}}_{a,t_{i+1}} + \alpha_2\psi_{e,t_i}\tilde{\psi}_{r,t_{i+1}}\right)\\ \hat{\rho}_{t_{i+1}} &= \hat{\rho}_{t_i} + \alpha_3\psi_{e,t_i}\tilde{\psi}_{r,t_{i+1}}.\end{aligned} \tag{A20}$$